Warming Up

Text Classification, Data Basics, and Perceptrons

Cornell CS 5740: Natural Language Processing
Yoav Artzi, Spring 2023

Table of Contents

Text Classification

- One of the most basic NLP tasks
- Input: a text
- Output: a label from a predefined set
- Learning problem: estimate the parameters of a function that maps a text to its label

Text Classification

Spam vs. Ham

- Input: email
- Output: spam/ham
- Setup:
  - Get a large collection of example emails, each labeled “spam” or “ham”
  - Note: someone has to hand label all this data!
  - Goal: learn to predict labels of new, future emails
- Features: the attributes used to make the ham / spam decision

Text Classification

Spam vs. Ham

- Input: email
- Output: spam/ham
- Setup:
  - Get a large collection of example emails, each labeled “spam” or “ham”
  - Note: someone has to hand label all this data!
  - Goal: learn to predict labels of new, future emails
- Features: the attributes used to make the ham / spam decision

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.

Text Classification

Spam vs. Ham

- Input: email
- Output: spam/ham
- Setup:
    - Get a large collection of example emails, each labeled “spam” or “ham”
    - Note: someone has to hand label all this data!
    - Goal: learn to predict labels of new, future emails
- Features: the attributes used to make the ham / spam decision

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.

Text Classification

Spam vs. Ham

- Input: email
- Output: spam/ham
- Setup:
    - Get a large collection of example emails, each labeled “spam” or “ham”
    - Note: someone has to hand label all this data!
    - Goal: learn to predict labels of new, future emails
- Features: the attributes used to make the ham / spam decision
    - Words: FREE!
    - Text Patterns: $dd, CAPS
    - Non-text: SenderInContacts — text is not alone 😍

✗

> 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.

Text Classification

- One of the most basic NLP tasks
- Let $\mathscr{V}$ be a vocabulary, $\mathscr{Y}$ be a set of classes, and $\mathscr{X}$ be the set of all texts
- A text $\bar{x} \in \mathscr{X}$ is a sequence of tokens $\bar{x} = \langle x_1, \ldots, x_n \rangle, x_i \in \mathscr{V}$
  - How do we go from text to tokens? 🤔
- A classifier is a function $f_\theta : \mathscr{X} \rightarrow \mathscr{Y}$ (i.e., $f_\theta(\bar{x}) = y$)
- Learning problem: estimate classifier parameters θ

Text Classification

Example: Text Categorization

Goal: classify documents into broad semantic topics

> Obama is hoping to rally support for his $825 billion stimulus package on the eve of a crucial House vote. Republicans have expressed reservations about the proposal, calling for more tax cuts and less spending. GOP representatives seemed doubtful that any deals would be made.

> California will open the 2009 season at home against Maryland Sept. 5 and will play a total of six games in Memorial Stadium in the final football schedule announced by the Pacific-10 Conference Friday. The original schedule called for 12 games over 12 weekends.

- Which one is the POLITICS document? Did this require a deep analysis?
- Usually start with a labeled corpus containing examples of each class
- Is this a good way to think about the topic of a text?

Text Classification

Example: Sentiment Analysis

Goal: detect the overall sentiment of the text

This movie was great! Will watch again

Not bad at all! but not a masterpiece

Could never enjoy, even with closed eyes

Text Classification

Example: Sentiment Analysis

Goal: detect the overall sentiment of the text

This movie was great! Will watch again 🤗

Not bad at all! but not a masterpiece 🙂

Could never enjoy, even with closed eyes 🤮

Text Classification

Example: Sentiment Analysis

Goal: detect the overall sentiment of the text

This movie was great! Will watch again 🤗

Not bad at all! but not a masterpiece 🙂

Could never enjoy, even with closed eyes 🤮

Text Classification

Example: Sentiment Analysis

Goal: detect the overall sentiment of the text

This movie was great! Will watch again 🤗

Not bad at all! but not a masterpiece 🙂

Could never enjoy, even with closed eyes 🤮

- Did this require more reasoning compared to categorization?

Text Classification

Example: Sentiment Analysis

Goal: detect the overall sentiment of the text

This movie was great! Will watch again 🤗

Not bad at all! but not a masterpiece 🙂

Could never enjoy, even with closed eyes 🤮

- Did this require more reasoning compared to categorization?
- Just spotting individual words is not enough
- Is this a reasonable way to model sentiment?

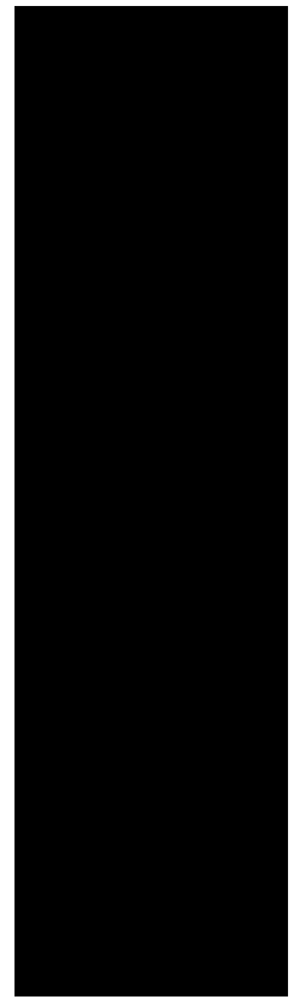
Text Classification

Learning Setup

- The most common approach is using supervised learning
- Assume an annotated dataset $\{(\bar{x}^{(j)}, y^{(j)}\}_{j=1}^{N}$ of N text-label pairs
- Use this data to train your model, and life is great 🏝️
- Simple enough, right?

Training with Data

Data



- What is our goal when we train a model?
- We want a model that will preform as good as possible (i.e., 🚀) when it is given data in the wild
- So, need to test our model on this data
- But: this is not possible — why?
- The question is: how can we get as close as possible to this with the data we have

Data Splits

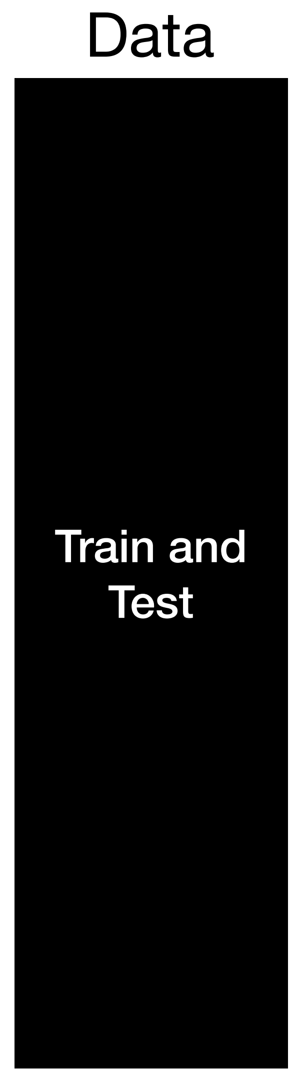
Proposal 1

- The more data we have for training the better
- The more data we have for testing the better
- So: train on all the data, and test on all the data 🤓
- The biggest possible number of examples for both training and testing 👍

Data Splits

Proposal 1

- The more data we have for training the better
- The more data we have for testing the better
- So: train on all the data, and test on all the data
- The biggest possible number of examples for both training and testing 👍
- Any issues?



Data Splits

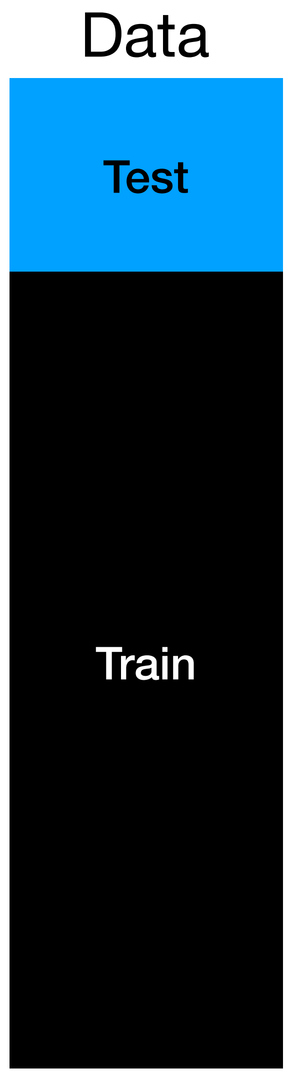
Proposal 1

- The more data we have for training the better
- The more data we have for testing the better
- So: train on all the data, and test on all the data
- The biggest possible number of examples for both training and testing 👍
- Any issues?
  - We optimize our parameters on the test data
  - Can just do great by memorizing it, performance means little 🤦‍♂️

Data Splits

Proposal 2: Separate Train and Test

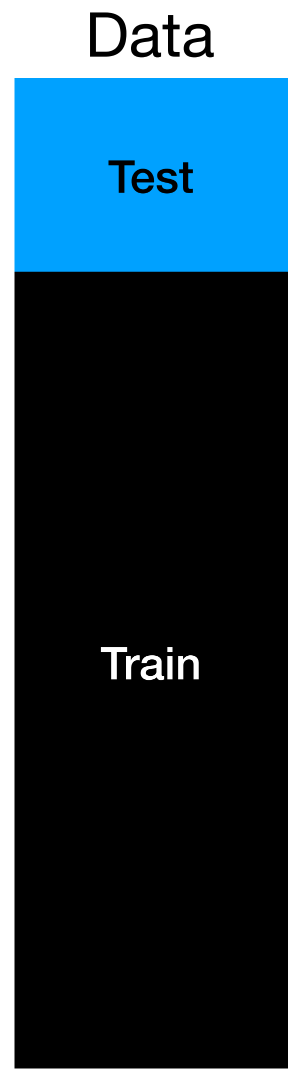
- Let's split train and test
- How to split? Need to balance 😵‍💫
  - More training data → better parameter estimates
  - More test data → evaluation is more reliable
- If we have very little data, consider cross validation → why?
- Any issues?



Data Splits

Proposal 2: Separate Train and Test

- Let's split train and test
- How to split? Need to balance
  - More training data → better parameter estimates
  - More test data → evaluation is more reliable
- If we have very little data, consider cross validation → why?
- Any issues?
  - During development we train and test many times to evaluate design decisions and select hyper parameters
  - As we use the test data more and more, we overfit to it, and it reflects reality less and less



Data Splits

Proposal 3: train/test/dev

- Let's create another split to distinguish development testing from real testing
- How to split? Same considerations
- How to choose between using test and dev?
  - Ideally: use test only once, and never look at the data 🙈
  - This way you make no decisions based on it, and it reflects real-life performance as well as possible
- No free lunch: slicing the same dataset to smaller sets
- Are we happy? Any issues?

Data Splits

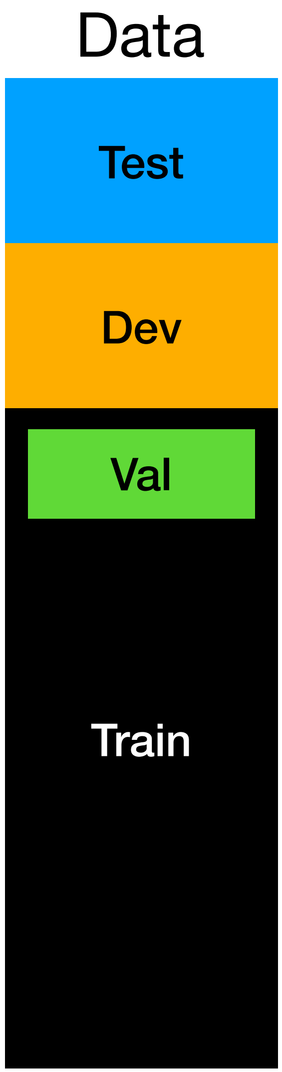
Proposal 3: train/test/dev

- Let's create another split to distinguish development testing from real testing
- How to split? Same considerations
- How to choose between using test and dev?
  - Ideally: use test only once, and never look at the data
  - This way you make no decisions based on it, and it reflects real-life performance as well as possible
- No free lunch: slicing the same dataset to smaller sets
- Are we happy? Any issues?
  - Contemporary ML methods require model selection
  - Can we use the development data?

Data Splits

Proposal 4: train/test/dev/validation

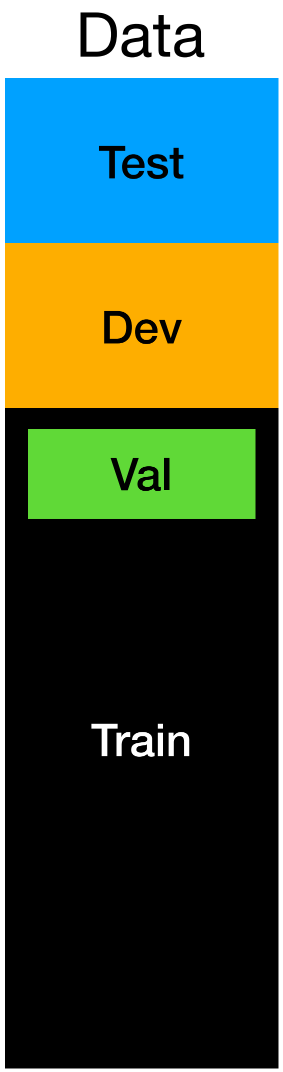
- Let's create a special set for model selection
- After training, we test on the development data
- Why can't we use validation for testing?
  - Because the model selection decision will overfit to it
- All is well?



Data Splits

Proposal 4: train/test/dev/validation

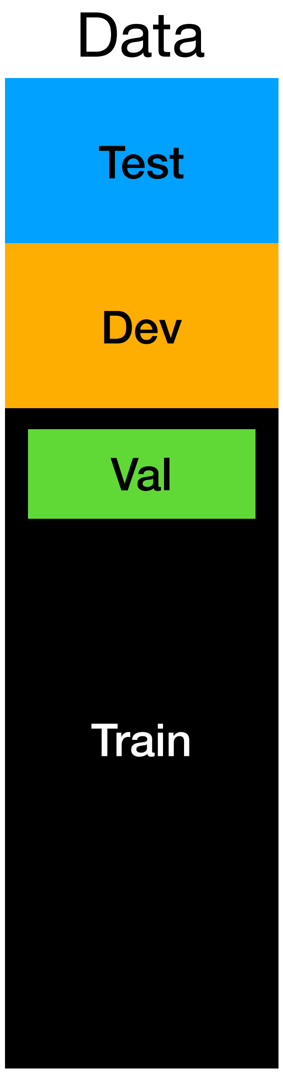
- Let's create a special set for model selection
- After training, we test on the development data
- Why can't we use validation for testing?
  - Because the model selection decision will overfit to it
- All is well?
  - Almost!



Data Splits

A Few Final Considerations

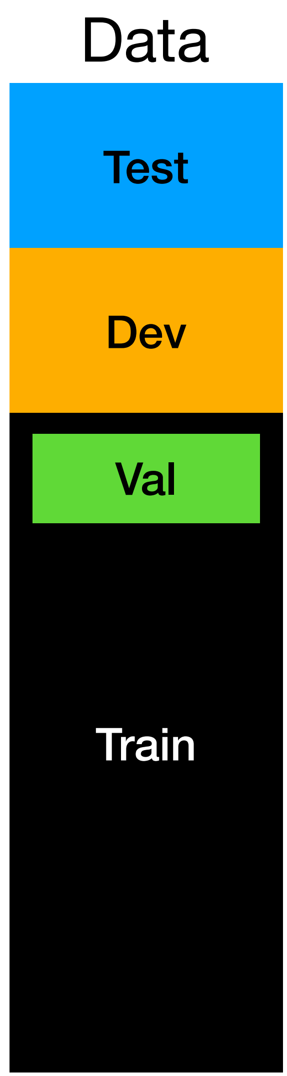
- Test and development are often standard in NLP tasks — why is this not ideal? So why do it?
- Validation is usually not standard, and just sampled from training data
  - Because the need for it is method dependent
  - Good to keep it stable for debugging
- Want the data to give you the most accurate picture of deployment?
  - Shuffle dev and train once in a while
  - Touch test as little as possible
    - Do I want to use test? Yes… am I deploying next? So, no!



Data Splits

A Few Final Considerations

- In most case, you won't get the data nicely packaged and organized
- Handling it well is on you, or you will get hit when you deploy
- It gets more complicated in non-stationary scenarios — almost any deployed system



Data

Other Issues

- Annotation
- Evaluation
- Data source issues
  - Ownership, privacy
- Bias
- Validity for deployment

Linear Models

- A class of models that scores outputs using a linear function
- We will discuss the simple Perceptron linear model and learning algorithm
- Good to know about, but won't discuss:
  - Computing distributions with the addition of normalization
  - Such discriminative model (i.e., that compute a conditional distribution) vs. generative model (i.e., that compute a joint distribution)

Linear Models

The Binary Case

- Let w be a weight vector, $\mathcal{Y} = \{-1, 1\}$ is the set of output labels, and ϕ be a feature computation function

- Given an example text $\bar{x}$, the linear binary prediction rule (i.e., to find the best output):

$$y = \text{sign}(w^\top \phi(\bar{x}))$$

- Example problems?

The Perceptron

Frank Rosenblatt ’50, Ph.D. ’56, works on the “perceptron” – what he described as the first machine “capable of having an original idea.”

NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo

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

July 8,1958

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.

Without Human Controls

The Navy said the perceptron would be the first non-living mechanism “capable of receiving, recognizing and identifying its surroundings without any human training or control.”

The “brain” is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

Mr. Rosenblatt said in principle it would be possible to build brains that could reproduce themselves on an assembly line and which would be conscious of their existence.

In today's demonstration, the “704” was fed two cards, one with squares marked on the left side and the other with squares on the right side.

Learns by Doing

In the first fifty trials, the machine made no distinction between them. It then started registering a “Q” for the left squares and “O” for the right squares.

Dr. Rosenblatt said he could explain why the machine learned only in highly technical terms. But he said the computer had undergone a “self-induced change in the wiring diagram.”

The first Perceptron will have about 1,000 electronic “association cells” receiving electrical impulses from an eye-like scanning device with 400 photo-cells. The human brain has 10,000,000,000 responsive cells, including 100,000,000 connections with the eyes.

The New York Times
Published: July 8, 1958

The Perceptron

- A very simple algorithm to train a linear model
- An error-driven algorithm
- Additive update rule
- Will cover the binary and multi-class case
  - Structured case is a simple generalization of multi-class (but won't cover it)
- Nice theoretical properties (will not discuss)
- Can be described in a slide, and implemented easily

The Perceptron

Binary Case

- Given:
  - A feature function ϕ
  - An annotated training set $\mathcal{D} = \{(\bar{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$
- Output:
  - A weight vector w

The Perceptron

Binary Case

1. Initialize weight vector with zeros: $w = \bar{0}$

2. Iterate over examples $(\bar{x}^{(i)}, y^{(i)}) \in \mathscr{D}$ until there are no errors:

    2.1. Make a prediction: $y^* = \text{sign}(w^\top \phi(\bar{x}^{(i)}))$

    2.2. If $y^* = y^{(i)}$ (i.e., the prediction is correct): goto next example

    2.3. Else: adjust weights

$$w = w - y^* \phi(\bar{x}^{(i)})$$

The Perceptron

Binary Case: What is it doing?

- If we made an error on $\bar{x}^{(i)}$, and the label is positive $y^{(i)} = 1$:
  - The new weight vector is $w'' = w' + \phi(\bar{x}^{(i)})$
  - The prediction rule is:
    $y* = \text{sign}(w^{''\top}\phi(\bar{x}^{(i)})) = \text{sign}((w' + \phi(\bar{x}^{(i)}))^\top \phi(\bar{x}^{(i)}))$:
  - Inside the sign:
    $(w' + \phi(\bar{x}^{(i)}))^\top \phi(\bar{x}^{(i)}) = w^{'\top}\phi(\bar{x}^{(i)}) + |\phi(\bar{x}^{(i)})|^2 > w^{'\top}\phi(\bar{x}^{(i)})$
  - So more likely (but not guaranteed) that $\text{sign}(w^{''\top}\phi(\bar{x}^{(i)})) > 0$
- Do the same at home for $y^{(i)} = -1$

The Perceptron

Two Binary Examples

Dataset I:

$$\phi(\bar{x}^{(1)}) = [1,1] \qquad y^{(1)} = 1$$

$$\phi(\bar{x}^{(2)}) = [1,-1] \qquad y^{(2)} = 1$$

$$\phi(\bar{x}^{(3)}) = [-1,-1] \qquad y^{(3)} = -1$$

Dataset II:

$$\phi(\bar{x}^{(1)}) = [1,1] \qquad y^{(1)} = 1$$

$$\phi(\bar{x}^{(2)}) = [1,-1] \qquad y^{(2)} = 1$$

$$\phi(\bar{x}^{(3)}) = [-1,-1] \qquad y^{(3)} = -1$$

$$\phi(\bar{x}^{(4)}) = [0.25,0.25] \qquad y^{(3)} = -1$$

1. Initialize weight vector with zeros: $w = \bar{0}$
2. Iterate over examples $(\bar{x}^{(i)}, y^{(i)}) \in \mathscr{D}$ until there are no errors:
   2.1. Make a prediction: $y^* = \text{sign}(w^\top \phi(\bar{x}^{(i)}))$
   2.2. If $y^* = y^{(i)}$ (i.e., the prediction is correct): goto next example
   2.3. Else: adjust weights

$$w = w - y^* \phi(\bar{x}^{(i)})$$

The Perceptron

Separating Hyperplane

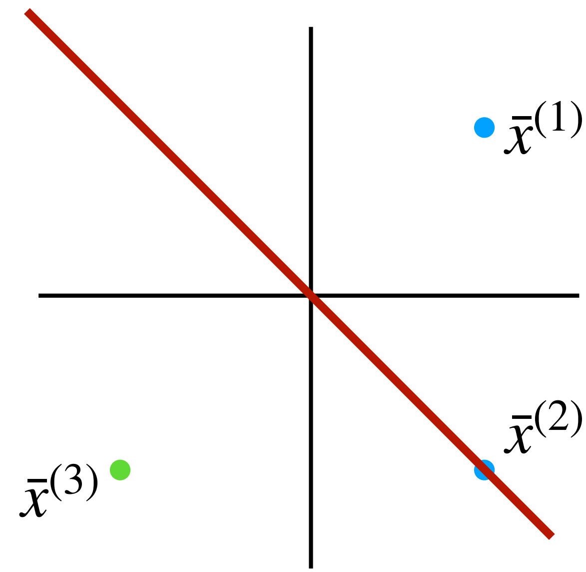
- The perceptron finds a separating hyperplane

$$\phi(\bar{x}^{(1)}) = [1,1] \qquad y^{(1)} = 1$$

$$\phi(\bar{x}^{(2)}) = [1,-1] \qquad y^{(2)} = 1$$

$$\phi(\bar{x}^{(3)}) = [-1,-1] \qquad y^{(3)} = -1$$

$$w = [1,1]$$

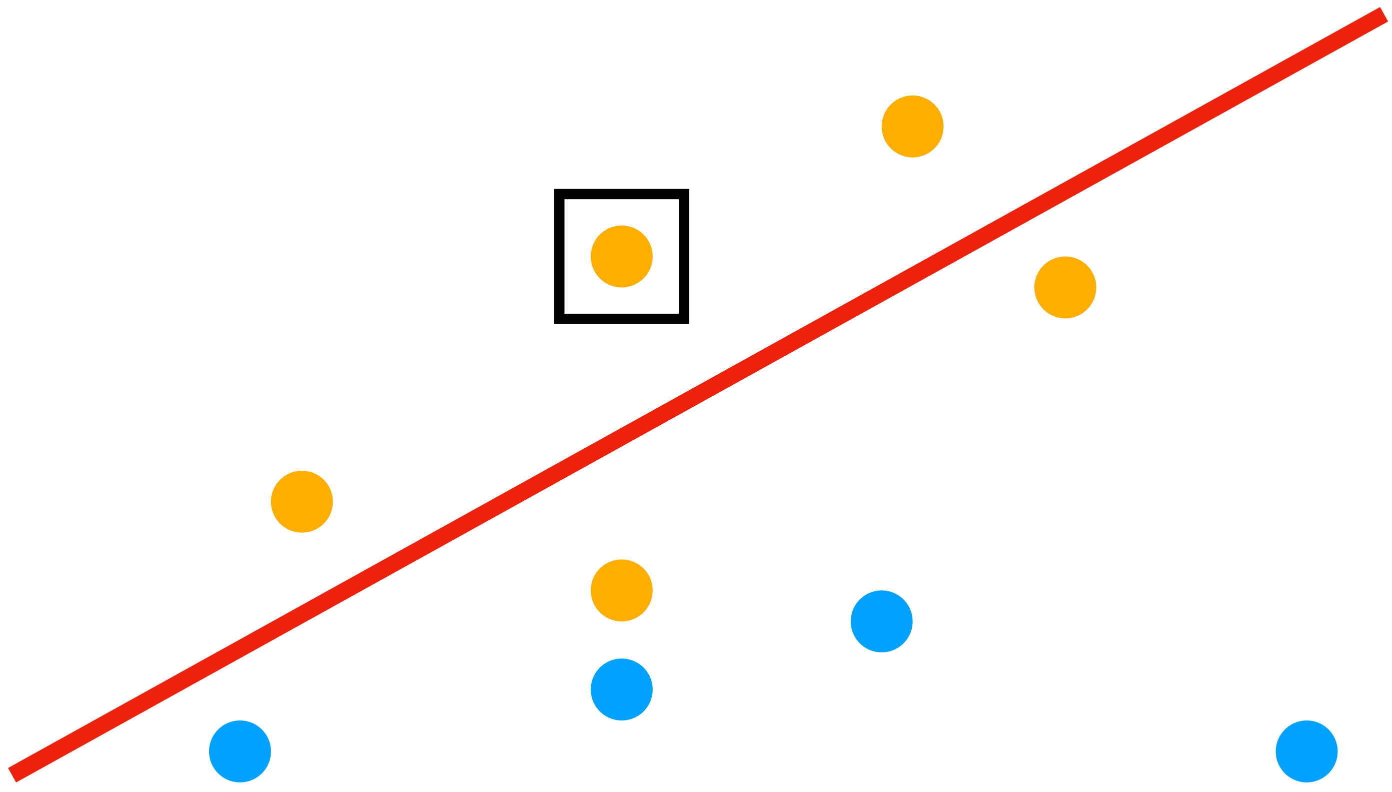


- Finding the hyperplane:

$$w^{\top}[x,y] = 1 \times x + 1 \times y = 0$$

The Perceptron

Separable Case



The Perceptron

Separating Hyperplane

- So what's going on with the second dataset?

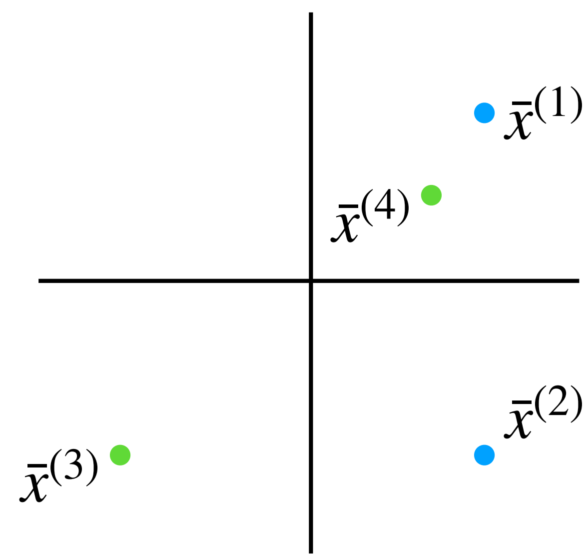
$$\phi(\bar{x}^{(1)}) = [1,1] \qquad y^{(1)} = 1$$

$$\phi(\bar{x}^{(2)}) = [1,-1] \qquad y^{(2)} = 1$$

$$\phi(\bar{x}^{(3)}) = [-1,-1] \qquad y^{(3)} = -1$$

$$\phi(\bar{x}^{(4)}) = [0.25,0.25] \qquad y^{(3)} = -1$$

$$w = [0,0]$$
$$w = [1,1]$$
$$w = [0.75,0.75]$$
$$w = [0.5,0.5]$$
$$w = [0.25,0.25]$$
$$w = [0,0]$$



Is there a separating hyperplane here?

The Perceptron

Adding Bias

- Decision rule: $y^* = \text{sign}(w^\top \phi(\bar{x}^{(i)}))$
- Algorithm stays the same!
- Only difference: add a dummy always-on feature

$$\phi(\bar{x}^{(1)}) = [1,1] \qquad y^{(1)} = 1$$
$$\phi(\bar{x}^{(2)}) = [1,-1] \qquad y^{(2)} = 1$$
$$\phi(\bar{x}^{(3)}) = [-1,-1] \qquad y^{(3)} = -1$$
$$w = [0,0] \in \mathbb{R}^2$$

→

$$\phi(\bar{x}^{(1)}) = [1,1,1] \qquad y^{(1)} = 1$$
$$\phi(\bar{x}^{(2)}) = [1,1,-1] \qquad y^{(2)} = 1$$
$$\phi(\bar{x}^{(3)}) = [1,-1,-1] \qquad y^{(3)} = -1$$
$$w = [0,0,0] \in \mathbb{R}^3$$

Linear Models

Multi-class Formulation

- Let w be a weight vector, $\mathcal{Y}$ is the set of all output labels, and ϕ be a feature computation function

- Given an example text $\bar{x}$ and a potential label $y \in \mathcal{Y}$, the score of assigning the label y to $\bar{x}$ is:

$$w^\top \phi(\bar{x}, y)$$

- The linear prediction rule (i.e., to find the best output):

$$y = \arg\max_{y \in \mathcal{Y}} w^\top \phi(\bar{x}, y)$$

- This requires a slightly different representation of w

Linear Models

Block Feature Representation

- Each feature-label combination has weight assigned to it in the weight vector w

- Both w and the features computed by ϕ follow a block structure, so $w \in \mathbb{R}^d$ and $\phi(\bar{x}, y) \in \mathbb{R}^d$, such that:

$$d = |\mathcal{Y}| \times \text{number of features}$$

Linear Models

Block Feature Representation Example

- Consider a simple text categorization problem
- $\mathscr{Y} = \{\text{SPORTS}, \text{POLITICS}, \text{OTHER}\}$
- Assume we have four binary features for the words *win*, *game*, *election*, or *car* appear

… win the election …

$$\downarrow$$

$$[1,0,1,0]$$

$$\downarrow$$

$$\phi(\bar{x}, \text{SPORTS}) = [1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$$

$$\phi(\bar{x}, \text{POLITICS}) = [0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0]$$

$$\phi(\bar{x}, \text{OTHER}) = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0]$$

Linear Models

Block Feature Representation Example

- Each feature computed by ϕ gets a weight in w

$$\phi(\bar{x}, \text{SPORTS}) = [1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$$

$$\phi(\bar{x}, \text{POLITICS}) = [0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0]$$

$$\phi(\bar{x}, \text{OTHER}) = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0]$$

$$w = [1\ 1\ -1\ -2\ 1\ -1\ 1\ -2\ -2\ -1\ -1\ 1]$$

- Compare labels based on their linear scores:

$$w^\top \phi(\text{... win the election ..., SPORTS}) = 1 \times 1 + (-1) \times 1 = 0$$

$$w^\top \phi(\text{... win the election ..., POLITICS}) = 1 \times 1 + 1 \times 1 = 2$$

$$w^\top \phi(\text{... win the election ..., OTHER}) = (-2) \times 1 + (-1) \times 1 = -3$$

- The highest scoring label is POLITICS

The Perceptron

Multi-class Case

- Just like before …
- Given:
  - A feature function ϕ
  - An annotated training set $\mathscr{D} = \{(\bar{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$
- Output:
  - A weight vector w

The Perceptron

Multi-class Case

1. Initialize weight vector with zeros: $w = \bar{0}$
2. Iterate over examples $(\bar{x}^{(i)}, y^{(i)}) \in \mathscr{D}$ until there are no errors:

   2.1. Make a prediction: $y^* = \arg\max_{y \in \mathscr{Y}} w^\top \phi(\bar{x}^{(i)}, y)$

   2.2. If $y^* = y^{(i)}$ (i.e., the prediction is correct): goto next example

   2.3. Else: adjust weights

$$w = w + \phi(\bar{x}^{(i)}, y^{(i)}) - \phi(\bar{x}^{(i)}, y^*)$$

The Perceptron

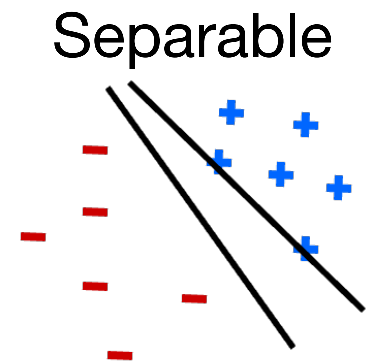
Multi-class Case: What is it doing?

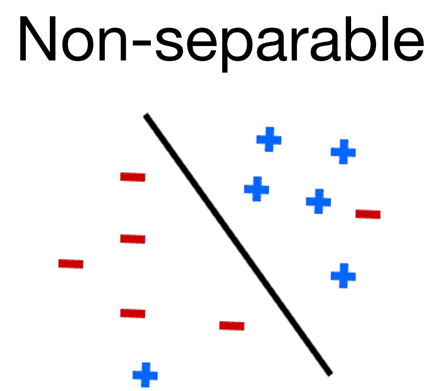
- Error on $\bar{x}^{(i)}$, predicted $y^* \neq y^{(i)}$:
  - New weight vector is: $w'' = w' + \phi(x^{(i)}, y^{(i)}) - \phi(x^{(i)}, y^*)$
  - Scoring breaks along block structure:
    - Let the block for label y is $w[y]$
    - Because all features $\phi(x^{(i)}, y)$ are 0's outside $w[y]$, the score for label y is: $w^\top \phi(x^{(i)}, y) = w[y]^\top \phi(x^{(i)}, y)[y]$
    - Now you can proceed exactly like the binary case!
    - Except that the update modifies both the score for label $y^{(i)}$ (increase) and the score for label y^* (decrease)
    - So, two birds for the price of one bird!

Perceptron Learning

Theoretical Properties

- **Separability:** some parameters get the training set perfectly correct
- **Convergence:** if the training is separable, perceptron will eventually converge
- **Mistake Bound:** the maximum number of mistakes (binary case) related to the margin or degree of separability

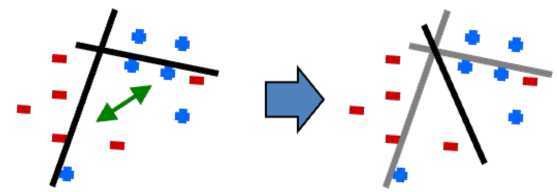




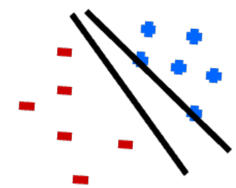
Perceptron Learning

Issues

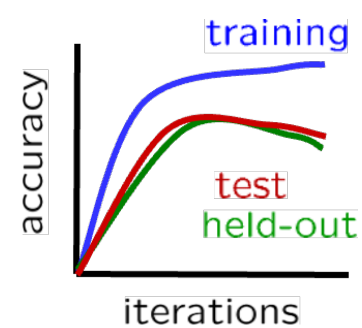
- Noise: if the data isn't separable, weights might thrash
  - Averaging weight vectors over time can help (averaged perceptron)



- Mediocre generalization: finds a "barely" separating solution



- Overtraining: test / held-out accuracy usually rises, then falls
  - Overtraining is a kind of overfitting



Perceptron

Example

Drugs

- Apo-Loperamide
- Minims Tropicamide
- Mexate
- Maxair

Names

- Alexander
- Anders
- Frederick
- Donald

Task: develop a feature function, and train a classifier using the Perceptron. Use at least two features.

Test data

1. Initialize weight vector with zeros: $w = \bar{0}$
2. Iterate over examples $(\bar{x}^{(i)}, y^{(i)}) \in \mathscr{D}$ until there are no errors:
   2.1. Make a prediction:
   $$y^* = \arg\max_{y \in \mathscr{Y}} w^\top \phi(\bar{x}^{(i)}, y)$$
   2.2. If $y^* = y^{(i)}$ (i.e., the prediction is correct): goto next example
   2.3. Else: adjust weights
   $$w = w + \phi(\bar{x}^{(i)}, y^{(i)}) - \phi(\bar{x}^{(i)}, y^*)$$

Perceptron

Example

Drugs

- Apo-Loperamide
- Minims Tropicamide
- Mexate
- Maxair

Names

- Alexander
- Anders
- Frederick
- Donald

Task: develop a feature function, and train a classifier using the Perceptron. Use at least two features.

Test data

- Tebamide
- Dexedrine

- Roderick
- Malcolm

1. Initialize weight vector with zeros: $w = \bar{0}$
2. Iterate over examples $(\bar{x}^{(i)}, y^{(i)}) \in \mathscr{D}$ until there are no errors:
   2.1. Make a prediction: $y^* = \arg\max_{y \in \mathscr{Y}} w^\top \phi(\bar{x}^{(i)}, y)$
   2.2. If $y^* = y^{(i)}$ (i.e., the prediction is correct): goto next example
   2.3. Else: adjust weights

$$w = w + \phi(\bar{x}^{(i)}, y^{(i)}) - \phi(\bar{x}^{(i)}, y^*)$$