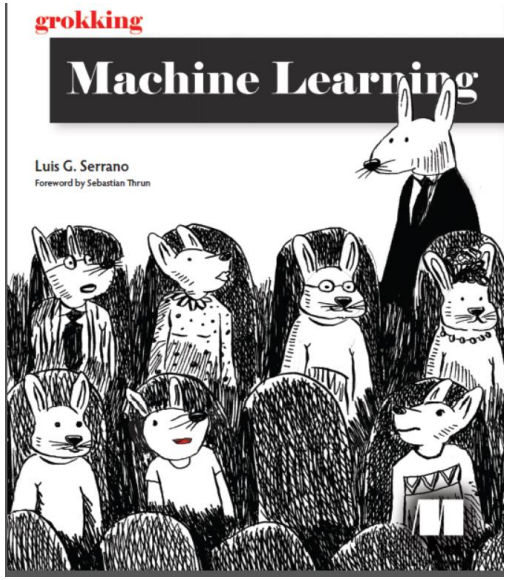

Machine Learning

| Perceptron Applications - Spam filters

Prof. Dr. Mostafa Elhosseini

https://youtube.com/@ElhosseiniAcademy

Agenda

- Applications
- Spam email filters

Applications of the perceptron algorithm

- The perceptron algorithm has many applications in real life.
- Virtually every time we need to answer a question with **yes or no**.

  - **Spam email filters**
  - Recommendation Systems
  - Health care
  - Computer vision

Spam email filters

- Identify and **block unwanted** or unsolicited emails, commonly known as spam.
- Spam emails often include **advertisements**, **phishing** attempts, malware, or other potentially harmful content.
- Spam filters help manage and **reduce** the number of these emails **reaching** users' inboxes, thereby improving the email experience and protecting against security threats.

Spam email filters – How it works

Spam filters typically use a combination of techniques to detect spam, including:

- **Keyword Filtering**: Analyzing the content of emails for specific words or phrases commonly associated with spam, such as "free," "win," "urgent," etc.

Spam email filters – How it works

Spam filters typically use a combination of techniques to detect spam, including:

- **Blacklists**: Maintaining lists of known spammer IP addresses or domains. Emails from these sources are automatically flagged or blocked.

Spam email filters – How it works

Spam filters typically use a combination of techniques to detect spam, including:

- **Heuristic Analysis**: Using rules based on common characteristics of spam emails, such as the presence of many links, suspicious attachments, or unusual sender names.

Spam email filters – How it works

Spam filters typically use a combination of techniques to detect spam, including:

- **Bayesian Filtering**: A statistical method that uses **probabilities** based on the **frequency of certain words or features** in known spam and legitimate emails. Over time, this filter learns to identify spam more accurately.

Spam email filters – How it works

Spam filters typically use a combination of techniques to detect spam, including:

- **Machine Learning**: Advanced filters use machine learning algorithms to **analyze large datasets** of emails and **identify patterns** indicative of spam. These systems can continuously learn and improve **based on user feedback and new data**.

Spam email filters – How it works

Spam filters typically use a combination of techniques to detect spam, including:

- **Header Analysis**: Examining the metadata in **email headers**, such as the sender's **IP** address, **routing information**, and inconsistencies that might indicate spoofing.

Spam email filters – How it works

Spam filters typically use a combination of techniques to detect spam, including:

- **Content Analysis**: Scanning the **email body** for scripts, executable files, or other potentially harmful content that might be embedded in **attachments** or **links**.

Spam email filters – Benefits

- **Protection** from Malicious Content: Helps prevent phishing attacks, malware, and other security threats.
- Improved **Productivity**: Reduces the time users spend **sorting** through **unwanted** emails.
- Better Email Management: Keeps the inbox clean and organized, ensuring important **emails** are **not lost** among spam.

Spam email filters – Description of Features

- **Spam Keywords**: Number of keywords in the email that are commonly associated with spam.

- **Number of Links**: Total number of hyperlinks present in the email.

Spam email filters – Description of Features

- **Size of Attachment** (KB): The size of any attachments included in the email, measured in kilobytes.

- **Number of Recipients**: Number of recipients the email is addressed to. Multiple recipients can sometimes indicate spam.

Spam email filters – Description of Features

- **Suspicious Domain**: Binary value where 1 indicates the sender's domain is known for spam activity, and 0 indicates it is not.
- **Known Sender**: Binary value where 1 means the sender is in the recipient's contact list, and 0 means they are not.
- **Spam Label**: The label indicating whether the email is classified as spam (1) or not (0).

Spam email filters – Description of Features

Email ID	Spam Keywords	Number of Links	Size of Attachment (KB)	Number of Recipients	Suspicious Domain	Known Sender	Spam Label
1	5	10	500	1	1	0	1
2	0	1	0	1	0	1	0
3	8	15	1200	3	1	0	1
4	2	2	50	2	0	1	0
5	6	5	800	1	1	0	1
6	1	0	0	1	0	1	0
7	4	3	300	1	0	1	0
8	7	7	450	1	1	0	1
9	0	1	0	1	0	1	0
10	5	12	600	1	1	0	1

Spam email filters – Description of Features

- $\hat{y} = step(w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6)$

- x_i: Feature
- w_i: Weight
- w_0: Bias
- x_1: Spam Keywords
- x_2 : Number of Links
- x_3 : Size of Attachment
- x_4 : Number of Recipients
- x_5 : Suspicious Domain
- x_6 : Known Sender

Feature correlations

- $\hat{y} = step(-1 + 2x_1 + 1.5x_2 + 0.002x_3 + 1.2x_4 + 2.5x_5 + -3x_6)$

- x_1 : Spam Keywords – A higher weight for this feature (2.0) seems reasonable, as the presence of certain keywords is often a strong indicator of spam.

Feature correlations

- $\hat{y} = step(-1 + 2x_1 + 1.5x_2 + 0.002x_3 + 1.2x_4 + 2.5x_5 + -3x_6)$

- x_2: Number of Links – A weight of 1.5 suggests a moderate correlation between the number of links and the likelihood of an email being spam, which is plausible.

- x_3:Size of Attachment – The weight of 0.002 indicates a very small contribution to the decision. This makes sense as the size of an attachment alone may not strongly indicate spam.

Feature correlations

- $\hat{y} = step(-1 + 2x_1 + 1.5x_2 + 0.002x_3 + 1.2x_4 + 2.5x_5 + -3x_6)$

- x_4: Number of Recipients – A weight of 1.2 suggests that emails sent to multiple recipients may be more likely to be spam, which is a reasonable assumption.

- x_5: Suspicious Domain – A weight of 2.5 indicates a strong association with spam, which is realistic given that many spam emails come from questionable domains.

Feature correlations

- $\hat{y} = step(-1 + 2x_1 + 1.5x_2 + 0.002x_3 + 1.2x_4 + 2.5x_5 + -3x_6)$

- x_6: Known Sender – A negative weight of -3.0 indicates that known senders significantly decrease the likelihood of an email being classified as spam. This is reasonable because emails from known contacts are typically not spam.

Feature correlations

- $\hat{y} = step(-1 + 2x_1 + 1.5x_2 + 0.002x_3 + 1.2x_4 + 2.5x_5 + -3x_6)$

- **Bias Term**: The bias term (-1) helps shift the decision boundary. Its reasonableness depends on the overall distribution of spam and non-spam emails. A negative bias suggests that the model is slightly biased towards predicting non-spam unless there's strong evidence to classify an email as spam.

More Advanced

- Currently, the perceptron algorithm (and its more advanced counterparts, logistic regression and neural networks) and other classification models are used as a part of spam classification pipelines by most of the biggest email providers, with great results.

- We can also categorize emails using classification algorithms like the perceptron algorithm.

Feature correlations

- Classifying email into personal, subscriptions, and promotions is the exact same problem.

- Even coming up with **potential responses** to an email is also a classification problem, except now the labels that we use are responses to emails.