TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO
PHƯƠNG PHÁP TOÁN CHO TRÍ TUỆ NHÂN TẠO

LAB 3:
PHÂN LOẠI THƯ RÁC

Nguyễn Tấn Hùng - 23122007

Đoàn Hải Nam - 23122011

Võ Ngọc Hiếu - 23122027

Lý Nguyên Thương - 23122055

Giảng Viên:
Cấn Trần Thành Trung
Nguyễn Ngọc Toàn

TP. Hồ Chí Minh, tháng 6/2025

Mục lục

Giới thiệu

Báo cáo này trình bày quá trình thực hiện bài Lab 3, với chủ đề **Phân loại thư rác**. Mục tiêu của bài là áp dụng các kỹ thuật học thống kê (probabilistic models), cụ thể là mô hình Naive Bayes để phân loại email thành hai nhóm: `Spam` và `Ham` dựa trên nội dung thư.

Dữ liệu sử dụng trong bài là bộ **Enron-Spam**, một bộ dữ liệu kinh điển trong các nghiên cứu về lọc thư rác. Mỗi email bao gồm hai thông tin chính: `Subject` và `Message`, cùng với nhãn phân loại `Spam/Ham` tương ứng. Tập dữ liệu đã được chia thành hai phần: `train.csv` để huấn luyện mô hình và `val.csv` để đánh giá độ chính xác.

Quá trình thực hiện bài bao gồm các bước chính:

- **Tiền xử lý dữ liệu**: Làm sạch dữ liệu thô, xử lý các ký tự đặc biệt, từ dừng, chuẩn hóa chữ viết hoa/thường và gộp nội dung email.
- **Trích xuất đặc trưng**: Áp dụng các phương pháp xử lý ngôn ngữ tự nhiên để biến văn bản thành vector đặc trưng dùng cho mô hình học máy.
- **Huấn luyện mô hình Naive Bayes**: Sử dụng các công thức Maximum Likelihood Estimation (MLE) hoặc Maximum A Posteriori Estimation (MAP) để xây dựng mô hình phân loại.
- **Đánh giá mô hình**: Sử dụng tập dữ liệu kiểm tra (validation set) để đánh giá độ chính xác, độ nhạy, độ đặc hiệu của mô hình.
- **Thử nghiệm thực tế**: Viết thêm chức năng cho phép người dùng nhập nội dung email mới để mô hình tự động phân loại.

Báo cáo này sẽ mô tả chi tiết từng bước thực hiện, thuật toán sử dụng và đánh giá kết quả phân loại trên tập dữ liệu thực tế.

Chương 1

Tập dữ liệu Enron-Spam và Tiền xử lý dữ liệu

1.1 Đọc dữ liệu

Dữ liệu được đọc từ tập `train.csv` và `val.csv`. Sau đó loại bỏ cột dữ liệu không cần thiết và hiển thị thông tin tổng quan của bộ dữ liệu.

Các cột chính của dữ liệu gồm:

- **Subject**: Tiêu đề của thư điện tử.
- **Message**: Nội dung của email.
- **Spam/Ham**: Nhãn phân loại để xác định liệu email có là spam hay ham (không phải spam).

1.2 Xử lý dữ liệu

1.2.1 Kiểm tra sự phân bố nhãn trong cột `Spam/Ham`

Trước tiên là kiểm tra cột `Spam/Ham` để đảm bảo dữ liệu chỉ chứa hai nhãn hợp lệ: "spam" và "ham". Kết quả kiểm tra cho thấy dữ liệu đều vào là hoàn toàn hợp lệ.

Tần suất phân bố:

	spam	**ham**
train	13,858	13,426
val	1,563	1,521

Tỉ lệ các nhãn gần như đồng đều (xấp xỉ 1:1) đảm bảo mô hình không bị mất cân bằng khi huấn luyện.

1.2.2 Thay thế các giá trị `NaN` trong dữ liệu

Một số email thiếu dữ liệu ở cột `Subject` hoặc `Message`. Các giá trị thiếu được thay thế bằng chuỗi rỗng `""` để đảm bảo không gây lỗi khi xử lý tiếp.

1.2.3 Quy đổi nhãn trong cột `Spam/Ham` thành các giá trị nhị phân

Các nhãn văn bản được ánh xạ như sau:

- `"spam"` $\rightarrow 1$
- `"ham"` $\rightarrow 0$

1.2.4 Xử lý các dòng trùng lặp

Các dòng bị trùng lặp hoàn toàn sẽ được loại bỏ để tránh mô hình học lặp lại thông tin, làm sai lệch kết quả đánh giá.

1.2.5 Kết hợp nội dung cột `Subject` và `Message` tạo thành một đặc trưng duy nhất

Cả tiêu đề (Subject) và nội dung (Message) của email đều chứa thông tin quan trọng để phân loại spam hay ham. Ta sẽ thực hiện kết hợp hai trường này thành một trường mới "Content" đặc trưng toàn diện hơn.

1.2.6 Loại bỏ dấu câu và chuyển đổi sang chữ thường trong cột `Content`

Dấu câu hay chữ hoa thường không mang nhiều ý nghĩa trong việc phân loại email là spam hay ham. Chúng còn làm tăng kích thước từ vựng, làm cho mô hình phức tạp hơn và có thể làm giảm hiệu suất của mô hình.

1.2.7 Thay thế URL/Email bằng token trên cột `Content`

Các chuỗi URL hoặc địa chỉ email trong nội dung được thay thế bằng token tương ứng như: `"URLTOKEN"`, `"EMAILTOKEN"` để giúp mô hình tập trung vào cấu trúc ngôn ngữ hơn là thông tin cụ thể.

1.2.8 Thực hiện chuẩn hóa khoảng trắng

Các khoảng trắng thừa ở đầu/cuối câu hoặc nhiều khoảng trắng liên tiếp được thay thế thành một khoảng trắng duy nhất.

Ngoài ra còn có thể thực hiện các bước nâng cao như:

- Loại bỏ stopwords
- Lemmatization
- Token hóa từ

Tuy nhiên sau khi thử nghiệm, các bước nâng cao này khiến hiệu suất mô hình giảm nhẹ, nên không được sử dụng trong phiên bản nộp.

Tổng kết: Quá trình tiền xử lý dữ liệu đảm bảo bộ dữ liệu sạch, đồng nhất và cân bằng. Dữ liệu chuẩn hóa tốt giúp mô hình Naive Bayes học hiệu quả hơn.

Chương 2

Mô hình Naive Bayes Classifier

Naive Bayes là thuật toán phân loại dựa trên định lý Bayes với giả định "ngây thơ" về sự độc lập có điều kiện giữa các đặc trưng. Đây là một lựa chọn phù hợp cho bài toán phân loại văn bản nhờ tính đơn giản, hiệu quả và độ chính xác cao trong thực tế.

2.1 Cơ sở lý thuyết và Thiết kế mô hình

2.1.1 Định lý Bayes

Gọi D là một email và $C \in \{\text{spam}, \text{ham}\}$ là nhãn lớp. Khi đó:

$$P(C|D) = \frac{P(D|C) \cdot P(C)}{P(D)} \tag{2.1}$$

Trong đó:

- $P(C|D)$: Xác suất hậu nghiệm – xác suất email D thuộc lớp C.
- $P(D|C)$: Khả năng – xác suất tạo ra D khi biết C.
- $P(C)$: Xác suất tiên nghiệm – xác suất xảy ra của lớp C.
- $P(D)$: Bằng chứng – xác suất quan sát email D.

Vì $P(D)$ là hằng số đối với mọi lớp, ta có thể rút gọn:

$$C^* = \arg \max_{C \in \{\text{spam,ham}\}} P(D|C) \cdot P(C) \tag{2.2}$$

2.1.2 Giả định độc lập có điều kiện (Naive Assumption)

Một email D được biểu diễn bằng các từ $\{w_1, w_2, \ldots, w_n\}$. Khi đó:

$$P(D|C) = P(w_1, w_2, \ldots, w_n|C)$$

Giả định Naive cho rằng các từ là độc lập có điều kiện khi biết lớp C, nên:

$$P(D|C) \approx \prod_{i=1}^{n} P(w_i|C) \tag{2.3}$$

Kết hợp lại:

$$C^* = \arg\max_C P(C) \prod_{i=1}^{n} P(w_i|C) \tag{2.4}$$

2.1.3 Log-transform để tránh tràn số

Nhân nhiều xác suất nhỏ dễ gây tràn số dưới (underflow). Ta chuyển sang không gian log:

$$C^* = \arg\max_C \left(\log P(C) + \sum_{i=1}^{n} \log P(w_i|C) \right) \tag{2.5}$$

2.1.4 Ước lượng tham số: MLE và MAP

2.1.4.1 Xác suất tiên nghiệm $P(C)$ (MLE):

$$\hat{P}(C) = \frac{N_C}{N} \tag{2.6}$$

Trong đó:

- N_C: Số email thuộc lớp C.
- N: Tổng số email huấn luyện.

2.1.4.2 Xác suất có điều kiện $P(w|C)$ (MAP + Laplace smoothing):

Để tránh xác suất bằng 0 (nếu từ w chưa xuất hiện trong lớp C), ta áp dụng làm mịn Laplace:

$$\hat{P}(w|C) = \frac{\text{count}(w, C) + \alpha}{\text{count}(C) + \alpha \cdot |V|} \tag{2.7}$$

Với:

- $\text{count}(w, C)$: Số lần từ w xuất hiện trong lớp C.
- $\text{count}(C)$: Tổng số từ trong lớp C.
- $|V|$: Kích thước từ vựng.
- α: Tham số làm mịn.

2.1.4.3 Ý nghĩa của `alpha`:

- Khi $\alpha = 1.0$, ta có làm mịn Laplace – tương đương giả định rằng mỗi từ trong từ vựng đã xuất hiện ít nhất một lần trong mỗi lớp. Đây là giá trị phổ biến giúp mô hình ổn định và hoạt động tốt trong hầu hết các trường hợp.

- **Nếu chọn $\alpha = 0$ (không làm mịn):** Ta quay lại phương pháp ước lượng Maximum Likelihood (MLE) thuần túy. Điều này rất nguy hiểm trong bài toán phân loại văn bản: nếu một từ trong email kiểm thử chưa từng xuất hiện trong lớp C ở tập huấn luyện, thì $P(w|C) = 0$, dẫn đến toàn bộ tích xác suất bị triệt tiêu:

$$\prod_{i=1}^{n} P(w_i|C) = 0 \Rightarrow \log P(w_i|C) = -\infty$$

  Điều này khiến mô hình bị thiên lệch nghiêm trọng hoặc dự đoán sai.

- **Kết luận: Không nên** chọn $\alpha = 0$ trừ khi chắc chắn rằng tất cả từ trong dữ liệu kiểm thử đã từng xuất hiện trong huấn luyện – điều hầu như không xảy ra trong thực tế. Làm mịn với $\alpha > 0$ là cách tiếp cận an toàn và hiệu quả hơn.

- **Tinh chỉnh:** α là một siêu tham số, có thể được lựa chọn thông qua validation để tối ưu hiệu năng. Giá trị nhỏ như 0.1 hoặc 1.0 thường cho kết quả tốt.

2.1.4.4 Tóm tắt

Công thức cuối cùng của mô hình Naive Bayes áp dụng trong code là:

$$C^* = \arg\max_C \left(\log \hat{P}(C) + \sum_{i=1}^{n} \log \hat{P}(w_i|C) \right)$$

Mô hình học theo MLE với làm mịn Laplace (MAP), đảm bảo ổn định và hiệu quả trong phân loại văn bản.

Chương 3

Đánh giá mô hình

3.1 Mục tiêu đánh giá

Sau khi huấn luyện mô hình, việc đánh giá hiệu năng là rất quan trọng để biết mô hình hoạt động tốt đến đâu. Việc chỉ báo cáo độ chính xác (accuracy) trên tập huấn luyện là không đủ vì nó có thể che giấu hiện tượng quá khớp (*overfitting*). Do đó, chúng ta cần đánh giá trên cả tập huấn luyện (train set) và tập kiểm định (validation/test set) để hiểu rõ hơn khả năng tổng quát hóa của mô hình.

3.2 Các chỉ số đánh giá hiệu năng

3.2.1 Accuracy (Độ chính xác)

Tỷ lệ các email được phân loại đúng trên tổng số email:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

3.2.2 Precision (Độ chuẩn xác)

Tỷ lệ email thực sự là *spam* trong số các email được dự đoán là *spam*:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Ý nghĩa: Precision cao rất quan trọng trong bài toán lọc thư rác, vì nếu precision thấp, nhiều email hợp lệ sẽ bị gửi nhầm vào hộp thư rác (False Positive cao).

3.2.3 Recall (Độ nhạy hay Độ phủ)

Tỷ lệ email *spam* mà mô hình phát hiện được trong tổng số email spam thực sự:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Ý nghĩa: Recall cao giúp mô hình phát hiện được nhiều thư rác, tránh bỏ sót (False Negative thấp).

3.3 Kết quả thực nghiệm với các giá trị `alpha`

Bảng 3.1 thể hiện hiệu suất của mô hình với các giá trị khác nhau của siêu tham số `alpha`. Kết quả được thống kê trên cả hai tập dữ liệu: tập huấn luyện và tập kiểm định, với các chỉ số: Accuracy, Precision và Recall.

Bảng 3.1: Kết quả huấn luyện với các giá trị `alpha` khác nhau

Alpha	**Train**			**Validation**		
	Accuracy	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**
0.00	0.9965	0.9962	0.9970	0.9903	0.9910	0.9898
0.05	0.9952	0.9945	0.9962	0.9899	0.9891	0.9910
0.10	0.9948	0.9939	0.9958	0.9899	0.9891	0.9910
0.15	0.9945	0.9936	0.9955	0.9899	0.9891	0.9910
0.20	0.9943	0.9934	0.9953	0.9899	0.9891	0.9910
0.25	0.9942	0.9934	0.9953	0.9899	0.9891	0.9910
0.30	0.9942	0.9934	0.9952	0.9899	0.9891	0.9910
0.35	0.9941	0.9934	0.9951	0.9899	0.9891	0.9910
0.40	0.9940	0.9932	0.9949	0.9899	0.9891	0.9910
0.45	0.9937	0.9929	0.9947	0.9899	0.9891	0.9910
0.50	0.9934	0.9928	0.9942	0.9899	0.9891	0.9910
0.60	0.9932	0.9926	0.9940	0.9896	0.9891	0.9904
0.70	0.9931	0.9925	0.9940	0.9896	0.9891	0.9904
0.80	0.9930	0.9924	0.9938	0.9896	0.9891	0.9904
0.90	0.9928	0.9924	0.9935	0.9893	0.9885	0.9904
1.00	0.9927	0.9923	0.9934	0.9893	0.9885	0.9904

3.4 Phân tích kết quả

Từ bảng trên, ta có thể thấy:

- Khi `alpha` = 0 (không regularization), mô hình đạt độ chính xác rất cao trên tập huấn luyện nhưng chênh lệch so với tập kiểm định, cho thấy có khả năng bị overfitting nhẹ.
- Khi `alpha` càng gần 0, hiệu suất trên tập huấn luyện càng tăng nhưng tập kiểm định vẫn duy trì ổn định.
- Khi `alpha` càng gần 1, hiệu suất ở cả tập huấn luyện và kiểm định đều giảm nhẹ.

Như vậy ta chọn giá trị alpha = 0.05.

Chương 4

Tự đánh giá

4.1 Đánh giá công việc

Họ và tên	**MSSV**	**Đánh giá**
Nguyễn Tấn Hùng	23122007	100%
Đoàn Hải Nam	23122011	100%
Võ Ngọc Hiếu	23122027	100%
Lý Nguyên Thương	23122055	100%

Bảng 4.1: Bảng đánh giá thành viên

4.2 Ưu điểm

- Các thành viên đều nắm được lý thuyết cơ bản về phân loại Naive Bayes, bao gồm MLE, MAP và vai trò của regularization.
- Mô hình được cài đặt và huấn luyện thành công, cho kết quả khá cao và ổn định trên cả tập huấn luyện và kiểm định.
- Làm việc nhóm hiệu quả, phân chia công việc hợp lý và hoàn thành đúng tiến độ.

4.3 Khuyết điểm

1. Kiến thức về xác suất và thống kê của một số thành viên còn hạn chế, gây khó khăn ban đầu trong việc hiểu MAP/MLE.
2. Một vài biểu thức và giả định xác suất còn trình bày sơ sài trong báo cáo do giới hạn về số trang.

Tài liệu tham khảo

1. Tài liệu của GVHD: Cấn Trần Thành Trung

2. Mathematics for Machine Learning