ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

MATH FOR AI

Spam Filtering with Naive Bayes

NHÓM 5

Giáo viên	Cấn Trần Thành Trung
	Nguyễn Ngọc Toàn
23122003	Nguyễn Văn Linh
23122022	Trần Hoàng Gia Bảo
23122026	Trần Chấn Hiệp
23122040	Nguyễn Thị Mỹ Kim

TP. HỒ CHÍ MINH, THÁNG 6/2025

Mục lục

1 Chứng minh toán học cho mô hình thống kê naive bayes classifier

1.1 Phân phối Multinomial Naive Bayes sử dụng trong mô hình

Tổng quan:

Naive Bayes Classification là mô hình học máy có giám sát phổ biến trong bài toán Classification. Đây là một mô hình đơn giản nhưng hiệu quả trong các bài toán phân loại, cụ thể là nhận biết email có là spam hay không, (bên cạnh đó còn các phân loại như cảm xúc thông qua văn bản (sentiment classification), phân loại tài liệu (document classification)). Trong đồ án này, ta sẽ xây dựng mô hình Naive Bayes Classification cho việc nhận biết spam emails.

Tóm tắt thuật toán:

Xử lí dữ liệu đầu vào. Lọc ra 2 loại spam và ham, thống kê số lượng từ và tập hợp các từ xuất hiện ứng với từng loại. Từ đó thống kê số lần số lần xuất hiện của từng từ.
Dùng MLE hoặc MAP để ước lượng phân phối cho từng nhãn dán.
Từ phân phối tìm được, tính điểm dựa trên phân phối cho từng nhãn và so sánh để phân loại.

Cơ sở toán học của mô hình:

Việc tính điểm được dựa trên công thức Bayes về xác suất có điều kiện:

$$P(D|\theta) = \frac{P(\theta|D)P(D)}{P(\theta)}$$

Từ đây tập dữ liệu, ta sẽ dùng MLE hoặc MAP để ước lượng phân phối cho số lần xuất hiện của từ (Multinomial Distribution). Giả sử từ dữ liệu θ ta có $S = \{w_1, ..., w_v\}$ là các từ xuất hiện trong tất cả các email có nhãn là spam, ta tin rằng một email spam thì sẽ có thể chứa các từ phổ biến, nên ta sẽ giả sử tồn tại một phân phối (Multinomial Distribution) cho sự xuất hiện của các từ trong email là spam, và ta sẽ dựa vào dự liệu để ước lượng hợp lí phân phối này (Δ). Từ phân phối ta sẽ áp dụng vào việc tính toán, xác suất có điều kiện để một email là spam:

$$P(spam \mid email) = \frac{P(email \mid spam).P(spam)}{P(email)}$$

$P(email) = \sum_{all\ class\ C} P(email \mid C)$ ở đây được hiểu là xác suất xuất hiện của email với nội dung cụ thể (ta chỉ quan tâm tới các từ có xuất hiện trong email và tần số của chúng, không quan tâm thứ tự). Ta định nghĩa tương tự cho $P(ham \mid email)$. Thuật toán chính là khi ta tính $P(spam \mid email)$ và $P(ham \mid email)$ và phân loại dựa vào nhãn có kết quả lớn hơn nên việc tính $P(email)$ là không cần thiết (và ta cũng khó tính trực tiếp).
Chính vì vậy ta sẽ tập trung tính $P(email \mid spam).P(spam)$ (tương tự cho ham)
Ở đây, ta sẽ có thêm một giả sử nữa, dựa trên mô hình Naive Bayes Classification, ta đã biết là ta chỉ quan tâm đến từ và số lần xuất hiện của chúng chứ không quan tâm thứ tự, thêm vào đó, ta được biết thêm là nếu ta đã biết loại của email thì xác suất xuất hiện của từng từ là độc lập nhau. Chính vì vậy nên ta có:

$$P(email \mid spam).P(spam) = \prod_{w\ \in\ email} P(w \mid spam).P(spam)$$

Ta đã tìm hiểu xong về cách mô hình phân loại thư. Tiếp đến ta sẽ tìm hiểu 2 hướng ước lượng phân phối (Multinomial Distribution) cho tập hợp từ.

1.2 Maximum Likelihood estimation (MLE)

Đây là một trong hai cách ước lượng phân phối cho tập hợp từ trong email. Ta sẽ giả sử các từ tuân theo phân phối Multinomial Distribution (phù hợp cho bài toán có dạng văn bản), gọi $\theta = \{\theta_1, ..., \theta_v\}$ là xác suất xuất hiện của $S = \{w_1, ..., w_v\}$ trong tập tất cả các từ có trong email nhãn spam như đã đề cập ở phần Tóm tắt thuật toán (tương tự cho ham). MLE sẽ cho ta biết: $\theta_i = \frac{k_i}{N}$ với k_i là số lần từ w_i xuất hiện và $N = \sum k_i$ là tổng tất cả các từ có được từ tất cả email có nhãn là spam.

Chứng minh:

Ta có hàm hợp lí: $\mathcal{L}(\theta) = \prod_{i=1}^{v} \theta_i^{k_i}$. Ta đã biết hàm $\log(x)$(hay $\ln(x)$) đồng biến trên $\mathbb{R}^+$ nên ta sẽ tối ưu hàm $F(\theta) = \log(\mathcal{L}(\theta)) = \sum_{i=1}^{v} k_i.\log(\theta_i)$.
Ta có hàm $f(x) = k\log(x),\ k > 0$ là hàm lõm (concave function) trên $\mathbb{R}^+$ do $f^{''}(x) = -\frac{k}{x^2} < 0,\ x > 0$. Tương đương:

$$\forall x, y \in \mathbb{R}^+ \text{ thì } f(\lambda x + (1-\lambda)y) \geq \lambda f(x) + (1-\lambda)f(y), \forall \lambda \in [0,1]$$

Từ đây dễ dàng có được, $\forall X = (x_1, ..., x_v), Y = (y_1, ..., y_v)$ với $x_i, y_i > 0$:
Gọi $D' = \{X = (x_1, ..., x_v) \in \mathbb{R}^v \mid x_i > 0, i = \overline{1,v}\}$. Khi này, 2 điểm X và Y thuộc D', gọi $Z = \lambda X + (1-\lambda)Y,\ \lambda \in [0,1]$, hiển nhiên $z_i = \lambda x_i + (1-\lambda)y_i > 0$. Suy ra $Z \in D'$, hay ta có ngay D' lồi (convex).
Và:

$$F(\lambda X + (1-\lambda)Y) = \sum_{i=1}^{v} k_i \log(\lambda x_i + (1-\lambda)y_i) \geq \sum_{i=1}^{v} \lambda k_i \log(x_i) + (1-\lambda)k_i \log(y_i)$$

$$\Leftrightarrow F(\lambda X + (1-\lambda)Y) \geq \lambda F(X) + (1-\lambda)F(Y), \forall \lambda \in [0,1]$$

Từ đây ta có $F(\theta)$ là hàm lõm trên D'. Tiếp theo ta sẽ dùng phương pháp Nhân tử Lagrange để tìm cực trị cho hàm $F(\theta)$ với điều kiện $g(\theta) = \sum_{i=1}^{v} \theta_i - 1 = 0$. Ta cần có, nếu $\overline{\theta}$ là điểm dừng thì:

$$\nabla F(\overline{\theta}) = \lambda \nabla g(\overline{\theta}) \Leftrightarrow \frac{k_i}{\overline{\theta_i}} = \lambda \Leftrightarrow \overline{\theta_i} = \frac{k_i}{\lambda},\ i = \overline{1,v}$$

$$\Rightarrow 1 = \frac{\sum k_i}{\lambda} \Leftrightarrow \lambda = N$$

Thay vào ta có ngay: $\overline{\theta} = \{\frac{k_1}{N}, ..., \frac{k_v}{N}\}$ là một nghiệm thỏa điều kiện ràng buộc và $\overline{\theta} \in D'$.
Bây giờ ta gọi miền D chính là miền bị giới hạn từ các hàm ràng buộc $g(X)$ và $X \in D'$. Ta xét 2 điểm X và Y bất kì thuộc D, gọi $Z = \lambda X + (1-\lambda)Y,\ \lambda \in [0,1]$, hiển nhiên $Z \in D'$. Xét $\sum z_i = \sum(\lambda x_i + (1-\lambda)y_i) = \lambda + 1 - \lambda = 1$. Suy ra $Z \in D$, hay ta có ngay D lồi (convex). Do $\sum \overline{\theta_i} = 1$ nên $\overline{\theta} \in D$
Ta đã có $F(X)$ là hàm lõm trên miền D lồi:

$$F(\lambda Y + (1-\lambda)X) \geq \lambda F(Y) + (1-\lambda)F(X), \forall \lambda \in [0,1]$$

$$\Leftrightarrow F(Y) - F(X) \leq \frac{F(\lambda(Y-X)+X) - F(X)}{\lambda}$$

Cho $\lambda \to 0$ thu được: $F(Y) - F(X) \leq D_{Y-X}F(X) = \nabla F^T(X).(Y-X)$.
Ta xét:

$$\nabla F^T(\overline{\theta}).(Y - \overline{\theta}) = \sum_{i=1}^{v} \frac{k_i}{\overline{\theta_i}}(y_i - \overline{\theta_i}) = N\sum_{i=1}^{v} y_i - N = 0$$

Suy ra: $F(Y) - F(\overline{\theta}) \leq \nabla F^T(\overline{\theta}).(Y - \overline{\theta}) = 0,\ \forall Y \in D$ hay $F(Y) \leq F(\overline{\theta}),\ \forall Y \in D$. Suy ra ta có ngay $\overline{\theta}$ chính là nghiệm tối ưu cho hàm F hay là làm Log Hợp lí ta cần tìm ($\square$).
Ta đã hoàn thành phần ước lượng phân phối từ dữ liệu.

1.3 Maximum A Posteriori estimation (MAP) với Laplace Smoothing

Đây là cách còn lại để ước lượng phân phối cho tập hợp từ trong email. Ta vẫn sẽ giả sử các từ tuân theo phân phối Multinomial Distribution (phù hợp cho bài toán có dạng văn bản), gọi $\theta = \{\theta_1, ..., \theta_v\}$ là xác suất xuất hiện của $S = \{w_1, ..., w_v\}$. Tuy nhiên ta sẽ tìm θ không chỉ từ data như MLE, MAP sẽ có thêm một yếu tố gọi là "prior" đóng góp vào việc xác định phân phối. Ở mô hình này, ta sẽ thường giả sử θ sẽ tuân theo phân phối Dirichlet (Dirichlet Distribution) (hay $\theta \sim Dirichlet(\alpha)$, ta chọn $\alpha = 1$). MAP sẽ cho ta biết: $\theta_i = \frac{k_i+1}{N+V}$ với k_i là số lần từ w_i xuất hiện, $N = \sum k_i$ là tổng tất cả các từ có được từ tất cả email có nhãn là spam và $V = |S| = v$

Chứng minh:

Ta sẽ nói về phân phối Dirichlet (Dirichlet Distribution): Ta coi $\theta = \{\theta_1, ..., \theta_v\}$ là vector xác suất (Probabilities Vector), trong đó θ_i đại diện cho xác suất label thứ i nào đó ta đang quan tâm. Phân phối Dirichlet có tham số $\alpha = \{\alpha_1, ..., \alpha_v\}$. Khi đó, hàm mật độ xác suất của phân phối Dirichlet cho vector xác suất θ là:

$$Dir(\theta; \alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{v} \theta_i^{\alpha_i - 1}$$

trong đó $B(\alpha)$ là hàm Beta (Beta function), nếu α cố định thì $B(\alpha)$ là hằng số (constant). Bây giờ, ta sẽ tính hàm hợp lí hậu nghiệm:
Ta gọi Θ là tập gồm các từ trong tất cả emails nhãn spam, ta giả sử các từ xuất hiện trong tất cả emails nhãn spam tuân theo phân phối Multinomial θ. Ta ước lượng θ:
Ta tìm hợp lí tiền nghiệm (như MLE): $P(\Theta|\theta) = \prod_{i=1}^{v} \theta_i^{k_i}$. Ta có phân phối tiền nghiệm (ta có hiểu đơn giản xác suất xảy ra θ với giả thiết θ tuân theo Dirichlet Distribution với tham số α) $P(\theta) = \frac{1}{B(\alpha)} \prod_{i=1}^{v} \theta_i^{\alpha_i - 1}$. Tới đây hàm hợp lí hậu nghiệm là:

$$\mathcal{L}(\theta|\theta_0; \Theta) = P(\theta) \prod_{i=1}^{|\Theta|} \theta_i = \frac{1}{B(\alpha)} \prod_{i=1}^{v} \theta_i^{\alpha_i - 1} \prod_{i=1}^{v} \theta_i^{k_i} = \frac{1}{B(\alpha)} \prod_{i=1}^{v} \theta_i^{k_i + \alpha_i - 1}$$

. Ta cần tối ưu hàm hợp lí hậu nghiệm nên ta bỏ qua hằng số $\frac{1}{B(\alpha)}$, hơn nữa, $\log(x)$ là hàm đồng biến trên $\mathbb{R}^+$ nên ta sẽ tối ưu hàm log hàm hợp lí hậu nghiệm. Đặt:

$$F(\theta) = \log(B(\alpha)\mathcal{L}(\theta|\theta_0; \Theta)) = \sum_{i=1}^{v} (k_i + a_i - 1).\log(\theta_i)$$

Ta sẽ tối ưu $F(\theta)$ mà ta biết $\alpha_i > 0,\ i = \overline{1, v}$ (do điều kiện phân phối Dirichlet) nên kéo theo $k_i + a_i - 1 > 0,\ i = \overline{1, v}$. Chính vì vậy, như đã chứng minh ở phần MLE, ta tương tự cũng có $F(\theta)$ là hàm lõm trên D với $D = \{X = (x_1, ..., x_v) \in \mathbb{R}^v \ | x_i > 0,\ i = \overline{1, v}$ và $g(X) = \sum x_i - 1 = 0\}$. Ta dùng phương pháp Nhân tử Lagrange để tìm cực trị $\overline{\theta}$ cho $F(\theta)$:

$$\nabla F(\overline{\theta}) = \lambda g(\overline{\theta}) \Leftrightarrow \frac{k_i + \alpha_i - 1}{\overline{\theta_i}} = \lambda,\ \forall i = \overline{1, v} \Leftrightarrow \frac{k_i + \alpha_i - 1}{\lambda} = \overline{\theta_i},\ \forall i = \overline{1, v}$$

$$\Rightarrow \lambda = N + (\sum_{i=1}^{v} \alpha_i) - v$$

Vậy $\overline{\theta_i} = \frac{k_i+\alpha_i-1}{N+(\sum_{i=1}^{v}\alpha_i)-v}$. Ta lại có: $F(Y) - F(\overline{\theta}) \leq \nabla F^T(\overline{\theta}).(Y - \overline{\theta}) = \sum_{i=1}^{v} \frac{k_i+\alpha_i-1}{\overline{\theta_i}}(y_i - \overline{\theta_i}) = (N - v + \sum a_i)(\sum y_i) - (N - v + \sum a_i) = 0$. Thêm vào đó, hiển nhiên $\overline{\theta} \in D$. Suy ra $\overline{\theta}$ là nghiệm tối ưu của $F(\theta)$. Ta cho $a_i = 2$ (Laplace Smoothing) (giá trị này có thể thay đổi tùy ý) thì có ngay $\overline{\theta} = \{\frac{k_1+1}{N+v}, ..., \frac{k_v+1}{N+v}\}$. Tới đây ta đã hoàn tất chứng minh (□).

1.4 Kết luận

Ta đã hoàn thành 2 giải thích cũng như chứng minh 2 thuật toán MLE và MAP cho việc ước lượng phân phối Multinomial cho mô hình Naive Bayes. Hai thuật toán đều có thể tìm phân phối từ dữ liệu, nhưng MAP (đã kết hợp Laplace Smoothing) giúp cho xác suất không bao giờ bằng 0 dù cho từ đấy có tồn tại hay không trong tập train; tuy nhiên về mặt toán học thì MLE không thể làm được.
Trong code của thuật toán, ta cũng tận dụng tính đồng biến của $\log(x)$ và thay vì tính trực tiếp $score_{spam}$ và $score_{ham}$ như công thức, ta có thể tính khác đi để thuận tiện cho xử lí số vì mục đích của ta là so sánh 2 xác suất. Với $P(email \mid label) = \prod P(w_i \mid label)$ thì ta sẻ tính thông qua hàm $\log(P(email \mid label))$. Để khắc phục nhược điểm của MLE thì nếu như từ không xuất hiện trong dữ liệu train thì ta sẽ mặc định nó là một hàng số dương xấp xỉ 0 để tránh $\log(0)$. Từ đó việc tính toán được trở nên dễ dàng hơn.

2 Thực nghiệm trên tập dữ liệu Enron-Spam

2.1 Tiền xử lý dữ liệu

Nhóm sử du mô hình Naive Bayes, khi dự đoán label của văn bản như email (Spam/Ham), mô hình dựa vào xác suất: $\boldsymbol{P(\text{label}|\text{text}) \propto P(\text{label}) \times P(\text{text}|\text{label})}$ Tuy nhiên do text(email) là một chuỗi dài, ta không thể tính xác suất trực tiếp cho toàn bộ câu. Mô hình thống kê Bayes có giả thiết "Naive", nghĩa là các token (từ) trong câu là độc lập với nhau.

$$P(\text{text}|\text{label}) = \prod_i P(\text{token}_i\,|\text{label})$$

Do đó, ta cần tách câu thành các token để mô hình học và tính xác suất trên từng từ. Như vậy, trong tiền xử lí dữ liệu, nhiệm vụ chính của phần này là tách các nội dung trong email (một email có nội dung được kết hợp từ "Subject" và "Message") thành các token một cách hiệu quả. Nhóm cũng cần loại bỏ các yếu gây nhiễu như nội dung mail trùng lặp, các kí từ non-words, v.v

1. Loại bỏ dữ liệu trùng lặp

Dữ liệu có 51 dòng bị trùng lặp. Dữ liệu này có thể khiến mô hình học overfitting do các mẫu bị lặp, làm giảm khả năng tổng quát (generalization). Ví dụ: Nếu cùng một nội dung email lặp lại nhiều lần, mô hình phân loại email có thể học sai trọng số cho các từ đó.

2. Làm sạch dữ liệu, loại bỏ các non-words

Trước tiên, nhóm đưa các nội dung email trở về dạng **lower**, tránh trường hợp các từ giống nhau nhưng bị phân biệt thành các từ khác nhau. Ví dụ như: ["hello", "Hello", "hELLo"]. Sau đó, nhóm làm sạch dữ liệu bằng cách loại bỏ các "non-words". Ví dụ một đoạn text trong email

có cấu trúc như sau: "music can be started by clicking on the sound icon . to stop music , right click and end show :) heather". Non-word trong email này các kí tự như dấu phẩy ",", kí hiệu mặt cười ":)", dấu chấm ".", ... Ta cần loại bỏ các kí tự này để tránh nhiễu. Ngoài ra, nhóm còn loại bỏ các URLs (đường link), HTML tags, v.v

3. Tokenize

Để tách từ thành các từ riêng biệt, nhóm sử dụng **word_tokenize** của thư viện **nltk**. Hàm này tách token tốt hơn sử dụng **split** thông thường. Giả sử ta có một đoạn text như sau "hello world! how are you?", hàm **split** sẽ tách từ thành một list như sau ["hello", "world!", "how", "are", "you?"]. Tuy nhiên với hàm **word_tokenize**, do thư liện **nltk** hỗ trợ xử lý một phần ngôn ngữ, do đó kết quả sau khi tách từ là ["hello", "world", "!", "how", "are", "you", "?"]. Sau khi tách token, nhóm cũng loại bỏ các từ quá ngắn, hay quá dài. Ví dụ trong nội dung mail gốc có chứa kí tự mặt cười như ": D", việc tiền xử lí trong các phần trước sẽ đưa kí tự này về dạng "d" (lower + loại bỏ "non-word"). Các kí tự này có thể gây nhiễu trong quá trình đánh giá mô hình thống kê do không mang một ngữ nghĩa nào cả. Các kí tự quá dài (độ dài kí tự lớn hơn 20) mà không mang ngữ nghĩa như "csdcgbvuokejisroaqtckf" cũng là yếu tố gây nhiễu cần loại bỏ.

4.Loại bỏ stop-words và lemmatize

Stop words là những từ rất phổ biến, nhưng ít mang ý nghĩa phân biệt trong văn bản. Ví dụ trong tiếng Anh: the, is, at, on, in, and, but, you, I,.. Hơn nữa, stop words xuất hiện nhiều và đều trong nội dung của mail và đây cũng có thể là nhiễu do không giúp ích gì trong quá trình phân loại lớp "spam" và "ham".

Việc tách từ cần được hoàn thiện nhờ kĩ thuật lemmatization. Đây là kĩ thuật đưa các token tương tự về nghĩa như ["eat", "eaten", "eating", "ate"] về định dạng gốc ("eat") để tránh việc mô hình phân loại các từ trên là khác biệt nhau. Nhóm sử dụng **WordNetLemmatizer** của thư viện **nltk** hỗ trợ kĩ thuật này.

Bảng 1: Kết quả đánh giá các mô hình thống kê với các bước tiền xử lí

Dữ liệu	**MLE**		**MAP**	
	Accuracy	**F1-score**	**Accuracy**	**F1-score**
`Tokenize + Xóa duplicate`	0.9772	0.9772	0.9769	0.9769
`+ Loại bỏ non-words`	0.9789	0.9788	0.9776	0.9776
`+ Loại bỏ stop words + lemmatize`	0.9822	0.9822	0.9812	0.9812

Ở bước đầu tiên chỉ áp dụng tokenize và loại bỏ dữ liệu trùng lặp, độ chính xác của mô hình đã đạt mức khá cao: khoảng 0.9772 đối với MLE và 0.9769 với MAP.

Khi tiếp tục loại bỏ các ký tự non-words, độ chính xác của cả hai mô hình đều có cải thiện nhẹ. MLE tăng lên 0.9789, trong khi MAP đạt 0.9776. Mức tăng tuy không lớn nhưng phản ánh tác dụng của việc loại bỏ các yếu tố nhiễu không mang ngữ nghĩa.

Bước xử lý cuối cùng, gồm loại bỏ stop words và áp dụng lemmatization, mang lại cải thiện rõ rệt nhất. MLE đạt độ chính xác 0.9822 và MAP đạt 0.9812, cao hơn đáng kể so với các bước trước đó. Điều này cho thấy rằng việc chuẩn hóa ngữ nghĩa và loại bỏ các từ không có giá trị phân loại đã giúp mô hình tập trung tốt hơn vào các đặc trưng quan trọng của văn bản.

2.2 Áp dụng mô hình thống kê cho tập dữ liệu

Sau bước tiền xử lý, nhóm tiến hành xây dựng và so sánh hiệu suất của hai mô hình phân loại email dựa trên thuật toán Naive Bayes. Hai phương pháp ước lượng tham số chính được khảo sát là: Ước lượng Hợp lý Cực đại (MLE) và Ước lượng Xác suất Hậu nghiệm Tối đa (MAP).

Mô hình MLE với Laplace Smoothing

Ước lượng Hợp lý Cực đại (MLE) ước tính xác suất có điều kiện của một từ dựa trên tần suất tương đối của nó trong tập huấn luyện:

$$P_{MLE}(w|c) = \frac{\text{count}(w, c)}{N_c}$$

Trong đó $\text{count}(w, c)$ là số lần từ w xuất hiện trong lớp c, và N_c là tổng số từ của lớp c.

Một nhược điểm lớn của MLE là **vấn đề tần suất bằng không (zero-frequency problem)**. Trong cài đặt của nhóm, vấn đề này được xử lý bằng cách cộng một giá trị rất nhỏ (10^{-10}) vào log của xác suất để tránh lỗi tính toán `log(0)` khi một từ trong tập kiểm thử không xuất hiện ở một lớp nào đó trong tập huấn luyện.

Thực nghiệm và Kết quả chi tiết

Mô hình MLE được huấn luyện trên tập dữ liệu gồm **25.023** email và đánh giá trên tập kiểm thử gồm **3.084** email. Kết quả đánh giá chi tiết được trình bày trong Bảng 2.

Bảng 2: Kết quả đánh giá mô hình MLE

Lớp	**Precision**	**Recall**	**F1-score**
ham	0.9785	0.9855	0.9820
spam	0.9858	0.9789	0.9823
Tổng thể (Accuracy)	**0.9822**		
Tổng thể (F1-Average)	**0.9822**		

Mô hình MAP

Ước lượng Xác suất Hậu nghiệm Tối đa (MAP) cải tiến MLE bằng cách tích hợp thêm thông tin tiên nghiệm. Trong trường hợp này, việc sử dụng tiên nghiệm Dirichlet tương đương với kỹ thuật **Làm mịn Laplace (Laplace Smoothing)** với tham số $\alpha = 1$:

$$P_{MAP}(w|c) = \frac{\text{count}(w, c) + \alpha}{N_c + \alpha \cdot |V|}$$

Phương pháp này đảm bảo mọi từ, kể cả những từ chưa từng thấy, đều có một xác suất nhỏ lớn hơn 0, giúp mô hình tổng quát hóa tốt hơn.

Thực nghiệm và Kết quả

Mô hình MAP được huấn luyện với $\alpha = 1$ trên cùng điều kiện. Kết quả đánh giá được thể hiện trong Bảng 3.

Bảng 3: Kết quả đánh giá mô hình MAP với Làm mịn Laplace ($\alpha = 1$)

Lớp	**Precision**	**Recall**	**F1-score**
ham	0.9747	0.9875	0.9811
spam	0.9877	0.9750	0.9813
Tổng thể (Accuracy)		**0.9812**	
Tổng thể (F1-Average)		**0.9812**	

2.3 Kết luận

Bảng 4: So sánh hiệu suất giữa mô hình MLE và MAP

Metric	**Mô hình MLE**	**Mô hình MAP** ($\alpha = 1$)
Accuracy	**0.9822**	0.9812
F1-score	**0.9822**	0.9812
F1-score (ham)	**0.9820**	0.9811
F1-score (spam)	**0.9823**	0.9813

Kết quả trong Bảng 4 cho thấy rằng cả hai mô hình thống kê MLE và MAP với hệ số làm mịn Laplace $\alpha = 1$ đều đạt accuracy cao trên tập dữ liệu đã qua tiền xử lý đầy đủ. Tuy nhiên, mô hình MLE vẫn có kết quả tốt hơn ở tất cả các chỉ số đánh giá. Accuracy và F1-score tổng thể của MLE đều đạt 0.9822, cao hơn so với 0.9812 của MAP. Khi xét chi tiết hơn theo từng label, mô hình MLE vẫn tốt hơn MAP với F1-score đạt 0.9820 đối với label ham và 0.9823 đối với label spam, so với MAP là 0.9811 và 0.9813 tương ứng.

Sự khác biệt về kết quả không quá lớn (dưới 0.2%), nhưng với dữ liệu đã được lớn và tiền xử lí tốt như dữ liệu Enron-Spam, mô hình MLE dù không áp dụng thông tin prior vẫn có kết quả tốt hơn. Trong khi đó, MAP phù hợp hơn trong các trường hợp dữ liệu ít, phân bố không đều hoặc có từ hiếm.

Tài liệu

[1] K. P. Murphy, "Machine Learning: A Probabilistic Perspective," MIT Press, 2012. [Chương 2 và 3: MLE và MAP estimation].

[2] T. Mitchell, "Machine Learning," McGraw-Hill, 1997. [Chương 6: Bayesian Learning and the Naive Bayes Classifier].

[3] Scikit-learn, "Naive Bayes," [Online]. Available: `https://scikit-learn.org/stable/modules/naive_bayes.html`.

[4] S. Raschka, "Text Preprocessing for Spam Detection," [Online]. Available: `https://sebastianraschka.com/Articles/2014_spamfilter.html`.

[5] UCI Machine Learning Repository, "SMS Spam Collection Dataset," [Online]. Available: `https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection`.

[6] J. Brownlee, "How to Choose Feature Selection Methods for Machine Learning," [Online]. Available: `https://machinelearningmastery.com/feature-selection-machine-learning-python/`.