MACHINE LEARNING

Cao Văn Chung
cvanchung@hus.edu.vn
Informatics Dept., MIM, HUS, VNU Hanoi

Naive Bayes Classifier

Naive Bayes Classifier

- Xét bài toán phân loại thống kê với C lớp $1, 2, \ldots, C$:
- Cho dữ liệu $\mathbf{x} \in \mathbb{R}^d$, tính xác suất để $\mathbf{x}$ thuộc về lớp $c \in \{1, 2, \ldots, C\}$

$$p(y = c|\mathbf{x}) \qquad \text{hoặc viết gọn} \quad p(c|\mathbf{x}). \tag{1}$$

  Tức tính xác suất để đầu ra là class c với điều kiện đầu vào là $\mathbf{x}$.
- Xác suất $p(c|\mathbf{x})$, nếu tính được, cho phép xác định khả năng để $\mathbf{x}$ rơi vào mỗi lớp c.
- Từ đó, trong các bài toán phân loại, ta sẽ dự đoán đầu ra c là phân lớp có xác suất lớn nhất

$$c = \arg \max_{c \in \{1,\ldots,C\}} p(c|\mathbf{x}). \tag{2}$$

Naive Bayes Classifier

- Thực tế nhiều trường hợp biểu thức (2) không tính được trực tiếp.
- Khi đó, quy tắc Bayes thường được sử dụng:

$$p(c|\mathbf{x}) = \frac{p(\mathbf{x}|c)p(c)}{p(\mathbf{x})} = p(\mathbf{x}|c)p(c).$$

Ở đây đẳng thức thứ nhất là quy tắc Bayes; đẳng thức thứ hai là do dữ liệu quan sát $\mathbf{x}$ không phụ thuộc vào lớp c.

- Do đó (2) trở thành

$$\text{Tìm} \qquad c = \arg\max_{c \in \{1,\dots,C\}} p(\mathbf{x}|c)p(c). \tag{3}$$

Naive Bayes Classifier

- Trong biểu thức (3), $p(c)$ có thể được hiểu là xác suất để một điểm bất kỳ rơi vào phân lớp c.
- Từ đó, $p(c)$ có thể được tính theo ước lượng hợp lý cực đại (Maximum Likelihood Estimation - MLE), tức là tỷ lệ số phần tử thuộc lớp c trên tổng số phần tử tập dữ liệu

$$p(c) = \frac{\left|\{x \in \mathbf{X} : \text{label}(x) = c\}\right|}{|\mathbf{X}|}.$$

- Đại lượng còn lại $p(\mathbf{x}|c)$ là phân phối xác suất của dữ liệu bên trong lớp c. Thực tế, với $\mathbf{X}$ là đại lượng ngẫu nhiên quan sát được, trong không gian nhiều chiều, thì việc xác định chính xác phân bố $p(\mathbf{x}|c)$ là không khả thi.

Naive Bayes Classifier

- Để giải quyết được, người ta giả thiết các thành phần của $\mathbf{x}$ là độc lập với nhau, với c đã cho. Tức là

$$p(\mathbf{x}|c) = p(x_1, x_2, \ldots, x_d|c) = \prod_{i=1}^{d} p(x_i|c). \tag{4}$$

- Giả thiết về sự độc lập của các chiều dữ liệu này được gọi là *Naive Bayes*.
- Như tên của nó, thực tế giả thiết này quá chặt và ít khi có được với dữ liệu thực. Mặc dù vậy cứ áp dụng nó lại cho những kết quả ngoài mong đợi.
- Cách xác định class của dữ liệu dựa trên giả thiết này có tên là *Naive Bayes Classifier* (*NBC*).
- Do giả thiết Naive Bayes dẫn đến tính toán đơn giản hơn rất nhiều, nên phương pháp này rất phù hợp với dữ liệu lớn - large scale.

Naive Bayes Classifier

- **Phần Training**: Các phân phối $p(c)$ và $p(x_i|c), i = 1, \ldots, d$ được xác định dựa vào training data (bằng MLE hoặc MAP).
- **Phần Test**: với một điểm dữ liệu mới $\mathbf{x}$, nó sẽ được phân vào lớp c mà

$$c = \arg \max_{c \in \{1,\ldots,C\}} p(c) \prod_{i=1}^{d} p(x_i|c). \quad (5)$$

- Chú ý các $p(x_i|c) < 1$ và có thể rất nhỏ, nên khi số chiều d rất lớn, tích $\prod_{i=1}^{d} p(x_i|c)$ sẽ rất bé và do đó nhạy cảm với sai số.

Naive Bayes Classifier

- Do đó thay cho (5), ta dùng log Nepe của nó, và bài toán cực trị trở thành:

$$c = \arg \max_{c \in \{1,\dots,C\}} = \log(p(c)) + \sum_{i=1}^{d} \log(p(x_i|c)). \quad (6)$$

- Do hàm log đồng biến nên kết quả không đổi. Lợi thế của NBC là tính toán rất nhanh do giả thiết độc lập của các thành phần.
- Như trên đã nói, ta tính được các $p(c), c = 1, 2, \dots, C$. Ta cần tính các $p(\mathbf{x_i}|c)$.
- Việc tính các $p(\mathbf{x_i}|c)$ phụ thuộc vào loại dữ liệu. Có ba loại được sử dụng phổ biến là: *Gaussian Naive Bayes*, *Multinomial Naive Bayes*, và *Bernoulli Naive*.

Gaussian Naive Bayes

- Mô hình này được sử dụng chủ yếu trong loại dữ liệu mà các thành phần là các biến liên tục.
- Giả thiết, tại chiều dữ liệu thứ i ($i = 1, \ldots, d$) và phân lớp c, dữ liệu x_i tuân theo phân bố chuẩn có kỳ vọng μ_{ci} và phương sai σ_{ci}^2, tức là

$$p(x_i|c) = p(x_i|\mu_{ci}, \sigma_{ci}^2) = \frac{1}{\sqrt{2\pi\sigma_{ci}^2}} \exp\left(-\frac{(x_i - \mu_{ci})^2}{2\sigma_{ci}^2}\right). \quad (7)$$

- Trong đó bộ tham số $\theta = \{\mu_{ci}, \sigma_{ci}^2\}$ được xác định bằng Maximum Likelihood

$$(\mu_{ci}, \sigma_{ci}^2) = \arg\max_{\mu_{ci}, \sigma_{ci}^2} \prod_{n=1}^{N} p(x_i^{(n)}|\mu_{ci}, \sigma_{ci}^2). \quad (8)$$

- Đây là cách tính được tham khảo từ thư viện sklearn.

Multinomial Naive Bayes

- Mô hình này chủ yếu được sử dụng trong phân loại văn bản mà feature vectors được tính bằng *Bags of Words* (tự tìm hiểu các kỹ thuật trích xuất dữ liệu - *Feature Engineering*).
- Trong kỹ thuật này, mỗi văn bản được biểu diễn bởi một vector có độ dài d - là số từ trong từ điển. Dễ thấy số chiều sẽ rất lớn.
- Giá trị của thành phần (tọa độ) thứ i của mỗi vector chính là số lần từ thứ i (trong từ điển) xuất hiện trong văn bản.

Multinomial Naive Bayes

- Theo cách đặt trên $p(x_i|c)$ sẽ là tần suất xuất hiện từ thứ i trong toàn bộ các văn bản của lớp c. Giá trị này thường được tính theo công thức

$$\lambda_{ci} = p(x_i|c) = \frac{N_{ci}}{N_c}. \tag{9}$$

Ở đây

- N_{ci} là tổng số lần từ thứ i xuất hiện trong các văn bản thuộc lớp c. Nó được tính là tổng của tất cả các thành phần (tọa độ) thứ i của các điểm dữ liệu (feature vectors) trong phân lớp c.
- N_c là tổng số từ (kể cả lặp) xuất hiện trong phân lớp c. Tức là N_c bằng tổng độ dài tính theo từ của toàn bộ các văn bản thuộc vào lớp c.
- Có thể suy ra $N_c = \sum_{i=1}^{d} N_{ci}$, và do đó $\sum_{i=1}^{d} \lambda_{ci} = 1$.

Multinomial Naive Bayes

- Nhược điểm của các công thức trên: Nếu một từ không xuất hiện trong phân lớp c thì $\lambda_{ci} = p(x_i|c) = 0$, dẫn tới $p(\mathbf{x}|c)$ trong (5) luôn bằng 0, cho dù các từ khác có tần suất rất lớn.
- Đặc điểm này sẽ dẫn đến kết quả không chính xác.
- Để tránh nhược điểm này, một kỹ thuật được gọi là *Laplace smoothing* được áp dụng:

$$\hat{\lambda}_{ci} = \frac{N_{ci} + \alpha}{N_c + d\alpha}. \tag{10}$$

  - α là một số dương, thường bằng 1, để tránh trường hợp tử số bằng 0.
  - Mẫu số được cộng với $d\alpha$ nên có thể đảm bảo tổng xác suất $\sum_{i=1}^{d} \hat{\lambda}_{ci} = 1$.
- Bây giờ mỗi lớp c sẽ được mô tả bằng một bộ d số dương có tổng bằng 1: $\hat{\lambda}_c = \{\hat{\lambda}_{c1}, \ldots, \hat{\lambda}_{cd}\}$.

Bernoulli Naive Bayes

- Mô hình này được áp dụng cho các loại dữ liệu mà mỗi thành phần là một giá trị binary - bẳng 0 hoặc 1.
- **Ví dụ**: cũng với loại văn bản nhưng thay vì đếm tổng số lần xuất hiện của 1 từ trong văn bản, ta chỉ cần quan tâm từ đó có xuất hiện hay không.
- Trong trường hợp này, công thức tính các $p(x_i|c)$ như sau

$$p(x_i|c) = p(i|c)^{x_i}(1 - p(i|c)^{1-x_i}. \tag{11}$$

- $p(i|c)$ có thể được hiểu là xác suất từ thứ i xuất hiện trong các văn bản của lớp c.

Eg.1. Spam email - Classification

- Xét bài toán phân loại mail Spam (S) và Not Spam (N).
- Ta có bộ training data gồm E_1, E_2, E_3. Cần phân loại E_4.
- Bảng từ vựng: $[w_1, w_2, w_3, w_4, w_5, w_6, w_7]$.
- Ví dụ này có thể xử lý bằng Multinomial Naive Bayes hoặc Bernoulli Naive Bayes. Tuy nhiên ta sẽ sử dụng Multinomial Naive Bayes.
- Bảng thống kê số lần xuất hiện của từng từ trong từng email tương ứng có trong trang sau

Eg.1. Spam email - Classification

	Email	w_1	w_2	w_3	w_4	w_5	w_6	w_7	Label
Training data	E1	1	2	1	0	1	0	0	N
	E2	0	2	0	0	1	1	1	N
	E3	1	0	1	1	0	2	0	S
Test data	E4	1	0	0	0	0	0	1	?

- Tính các $p(c)$: Ta có $P(S) = \frac{1}{3}$, $P(N) = \frac{2}{3}$.
- Sử dụng Laplace Smoothing với $\alpha = 1$ ta tính được xác suất xuất hiện của từng từ trong văn bản như sau

Eg.1. Spam email - Classification

	Email	w_1	w_2	w_3	w_4	w_5	w_6	w_7
	E3	1	0	1	1	0	2	0
$P(w_i\|S)$	(trước Smoothing)	1/5	0/5	1/5	1/5	0/5	2/5	0/5
$P(w_i\|S)$	(sau Smoothing)	2/12	1/12	2/12	2/12	1/12	3/12	1/12

class = Not Spam (N)

	Email	w_1	w_2	w_3	w_4	w_5	w_6	w_7
	E1	1	2	1	0	1	0	0
	E2	0	2	0	0	1	1	1
Tổng		1	4	1	0	2	1	1
$P(w_i\|N)$	(trước Smoothing)	1/10	4/10	1/10	0/10	2/10	1/10	1/10
$P(w_i\|N)$	(sau Smoothing)	2/17	5/17	2/17	1/17	3/17	2/17	2/17

Eg.1. Spam email - Classification

Từ đó tính được

$$
\begin{aligned}
P(S|E_4) &\propto P(S)\prod_{i=1}^{7} P(w_i|S) \\
&\propto \frac{1}{3}\times(\frac{2}{12}\times\frac{1}{12}) \\
&\propto 0.0046
\end{aligned}
$$

$$
\begin{aligned}
P(N|E_4) &\propto P(N)\prod_{i=1}^{7} P(w_i|N) \\
&\propto \frac{2}{3}\times(\frac{2}{17}\times\frac{2}{17}) \\
&\propto 0.0092
\end{aligned}
$$

Eg.1. Spam email - Classification

Xác suất phân lớp tương ứng

$$P(S|E_4) = \frac{0.0046}{0.0046 + 0.0092} \approx 0.334$$
$$P(N|E_4) = \frac{0.0092}{0.0046 + 0.0092} \approx 0.666$$

Do đó ta phân loại E_4 là Not Spam (N).