Dự đoán spam email bằng thuật toán Naive Bayes

Môn: Trí Tuệ Nhân Tạo

==========

"Set peace of mind as your highest goal, and organize your life around it." –

Brian Tracy

Trần Trung Kiên
ttkien@fit.hcmus.edu.vn

Ngày 30 tháng 4 năm 2014

Tóm tắt nội dung

Trong bài tập này, các bạn sẽ cài đặt thuật toán Naive Bayes (+ làm trơn Laplace) để dự đoán một email có phải là spam hay không. Các bạn sẽ thấy với Octave/Matlab, ta có thể cài đặt thuật toán này chỉ với vài dòng code :-)

1 Mô tả dữ liệu

Từ tập dữ liệu gồm các email đã được đánh nhãn là spam ($y = 1$) hoặc non-spam ($y = 0$), ta sẽ xây dựng danh sách các từ xuất hiện trong tập các email. Danh sách các từ này được gọi **bộ từ vựng** (vocabulary). Để dễ hình dung, các bạn xem file `voca.txt` (nên mở bằng `Notepad++`). Đây là file bộ từ vựng gồm có 1899 từ được lấy ra từ tập các email.

Sau khi đã có được từ điển (gồm n từ), bước kế tiếp là biểu diễn lại mỗi email dưới dạng một véc-tơ nhị phân (nghĩa là mỗi thành phần chỉ nhận một trong hai giá trị 0 hoặc 1). Véc-tơ này gồm có n thành phần (ứng với n từ

trong bộ từ vựng), trong đó thành phần thứ i sẽ có giá trị bằng 1 nếu từ thứ i trong bộ từ vựng xuất hiện trong email và có giá trị bằng 0 nếu ngược lại. Ví dụ, ta có một véc-tơ ứng một email như sau:

$$x = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \\ 1 \\ 0 \end{bmatrix} \in \mathbb{R}^n$$

Sau khi đã biểu diễn lại toàn bộ tập email dưới dạng các véc-tơ nhị phân như trên, ta sẽ chia tập dữ liệu làm hai tập: tập train (tập huấn luyện) và tập test (tập kiểm tra). Tập train dùng để huấn luyện bộ phân lớp spam/non-spam. Sau khi huấn luyện, bộ phân lớp của ta sẽ có khả năng dự đoán một email có phải là spam hay không. Tập test dùng để đánh giá chất lượng bộ phân lớp, nghĩa là: sau khi đã huấn luyện xong bộ phân lớp, ta sẽ dùng nó để dự đoán trên tập test; sau đó, ta so sánh giá trị dự đoán với giá trị đúng và thống kê số lần dự đoán đúng.

Trong bài tập này, các bạn đã được cung cấp sẵn tập train (file `spamTrain.mat`) và tập test (file `spamTest.mat`).

1.1 Tập train

Để load tập train, ở cửa sổ command line của Octave/Matlab các bạn gõ câu lệnh: `load('spamTrain.mat')`. Sau câu lệnh này, trong bộ nhớ của Octave/Matlab sẽ có hai biến `X` và `y`. Bây giờ, ta hãy xem hai biến này có ý nghĩa gì?

Đầu tiên, ta hãy xem biến `X`. Các bạn gõ câu lệnh `size(X)`, ta sẽ thấy `X` là một ma trận có kích thước 4000×1899. Nghĩa là, tập huấn luyện của ta gồm có 4000 email và mỗi dòng của ma trận ứng với một véc-tơ biểu diễn của một email. Mỗi véc-tơ gồm có 1899 thành phần ứng với 1899 từ trong bộ từ vựng; trong đó, thành phần thứ i sẽ có giá trị bằng 1 nếu từ thứ i trong bộ từ vựng xuất hiện trong email và có giá trị bằng 0 nếu ngược lại.

Kế đến, ta hãy xem biến `y`. Các bạn gõ câu lệnh `size(y)`, ta sẽ thấy `y` là một véc-tơ cột gồm có 4000 thành phần (ứng với 4000 email). Thành phần thứ i của véc-tơ này sẽ có giá trị bằng 1 nếu email tương ứng là spam và

bằng 0 nếu ngược lại. Các bạn cũng có thể thử gõ câu lệnh `y` để xem giá trị của biến y.

1.2 Tập test

Tương tự như vậy, để load tập test, ở cửa sổ command line của Octave/Matlab các bạn gõ câu lệnh: `load('spamTest.mat')`. Sau câu lệnh này, trong bộ nhớ của Octave/Matlab sẽ có hai biến `Xtest` và `ytest`. Bằng cách gõ các câu lệnh tương tự như ở tập train, ta sẽ thấy tập test gồm có 1000 email.

2 Huấn luyện bộ phân lớp spam/non-spam với thuật toán Naive Bayes

Các ký hiệu:

Ký hiệu	**Ý nghĩa**
x	Véc-tơ biểu diễn của một email nói chung
y	Nhãn của một email nói chung ($y = 1$: spam, $y = 0$: non-spam)
$x^{(i)}$	Véc-tơ biểu diễn của email thứ i trong tập huấn luyện
$y^{(i)}$	Nhãn của email thứ i trong tập huấn luyện
x_j	Thành phần thứ j của véc-tơ biểu diễn của một email nói chung
m	Số lượng email trong tập huấn luyện
n	Số lượng thành phần của véc-tơ biểu diễn x (số từ trong bộ từ vựng)

Trong phần này, ta dùng thuật toán Naive Bayes (+ làm trơn Laplace) để huấn luyện bộ phân lớp spam/non-spam.

Với một email x, để dự đoán nó là spam hay non-spam, ta cần phải tính $p(y = 1|x)$ (xác suất x là spam) và $p(y = 0|x)$ (xác suất x không phải là spam). Sau đó, ta sẽ quyết định: x là spam ($y = 1$) nếu $p(y = 1|x) > p(y = 0|x)$, x là non-spam ($y = 0$) nếu ngược lại.

Theo định lý Bayes, ta có:

$$p(y = 1|x) = \frac{p(x|y = 1)p(y = 1)}{p(x)}$$

$$p(y = 0|x) = \frac{p(x|y = 0)p(y = 0)}{p(x)}$$

Nhận thấy, để so sánh $p(y = 1|x)$ và $p(y = 0|x)$ thì ta sẽ so sánh $p(x|y = 1)p(y = 1)$ và $p(x|y = 0)p(y = 0)$ (vì phần mẫu giống nhau).

Như vậy, để dự đoán được một email có phải là spam hay không, trong quá trình học (huấn luyện), ta cần học:

1. $\phi = p(y = 1)$: xác suất email là spam. Ta không cần tính $p(y = 0)$ vì $p(y = 0) = 1 - p(y = 1)$.

2. $p(x|y = 1)$. Theo qui tắc mắt xích (chain rule), ta có:

   $$p(x|y = 1) = p(x_1|y = 1)p(x_2|y = 1, x_1), p(x_3|y, x_1, x_2) \dots p(x_n|y = 1, x_1, x_2, \dots, x_{n-1})$$

   Để đơn giản hóa vấn đề, thuật toán Naive Bayes giả định rằng: x_i **và** x_j **độc lập với điều kiện** y. Nghĩa là:

   $$p(x_2|y = 1, x_1) = p(x_2|y = 1)$$

   $$p(x_3|y = 1, x_1, x_2) = p(x_3|y = 1)$$

   $$\dots$$

   $$p(x_n|y = 1, x_1, x_2, \dots, x_{n-1}) = p(x_n|y = 1)$$

   Với giả định này, ta có:

   $$p(x|y = 1) = \prod_{j=1}^{n} p(x_i|y = 1)$$

   Như vậy, cái thứ hai ta cần tính trong quá trình huấn luyện là $\phi_{j|y=1} = p(x_j = 1|y = 1)$ với $j = 1, 2, \dots, n$. Ta không cần tính $p(x_j = 0|y = 1)$ vì $p(x_j = 0|y = 1) = 1 - p(x_j = 1|y = 1)$

3. $p(x|y = 0)$. Lập luận tương tự như trên, ta có:

   $$p(x|y = 0) = \prod_{j=0}^{n} p(x_i|y = 0)$$

   Như vậy, cái thứ ba ta cần tính trong quá trình huấn luyện là $\phi_{j|y=0} = p(x_j = 1|y = 0)$ với $j = 1, 2, \dots, n$. Ta không cần tính $p(x_j = 0|y = 0)$ vì $p(x_j = 0|y = 0) = 1 - p(x_j = 1|y = 0)$

Với một tập huấn luyện gồm m email, các tham số trên được tính như sau:

$$\phi = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}{m}$$

$$\phi_{j|y=0} = \frac{1+\sum_{i=1}^{m} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\}}{2+\sum_{i=1}^{m} 1\{y^{(i)} = 0\}}$$

$$\phi_{j|y=1} = \frac{1+\sum_{i=1}^{m} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\}}{2+\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}$$

Ở đây, $1\{\cdot\}$ được gọi là **indicator function**. Hàm này sẽ trả về giá trị 1 nếu tham số đầu vào của nó có giá trị `true` và sẽ trả về 0 nếu tham số đầu vào của nó có giá trị `false`. Ví dụ, $1\{3 = 2\} = 0$, $1\{3 > 2\} = 1$.

Trong công thức tính $\phi_{j|y=0}$ và $\phi_{j|y=0}$, các số màu đỏ được cộng thêm ở tử và mẫu ứng với công thức làm trơn Laplace. Các bạn nghĩ xem có đúng không?

Công việc của các bạn trong phần này là viết file `trainNB.m` ứng với hàm huấn luyện Naive Bayes. Hàm này có dạng:

```
function [phi, phi0, phi1] = trainNB(X, y)
```

Trong đó:

- `X, y`: tập huấn luyện (xem lại mục 1.1)
- `phi`, `phi0`, `phi1` lần lượt ứng với ϕ, $\phi_{j|y=0}$, $\phi_{j|y=1}$. Lưu ý: `phi` $\in \mathbb{R}$, `phi0` $\in \mathbb{R}^n$, `phi1` $\in \mathbb{R}^n$.

3 Dự đoán với bộ phân lớp đã được huấn luyện

Sau quá trình học (huấn luyện), ta đã tính được các tham số ϕ, $\phi_{j|y=0}$, $\phi_{j|y=1}$. Bây giờ, với một email mới x, ta có thể dự đoán nó có phải là spam hay không. Theo công thức Bayes, ta có:

$$p(y = 1|x) = \frac{p(x|y = 1)p(y = 1)}{p(x)}$$

$$p(y=0|x) = \frac{p(x|y=0)p(y=0)}{p(x)}$$

Ta sẽ quyết định email là spam ($y=1$) nếu $p(y=1|x) > p(y=0|x)$, nghĩa là nếu $p(x|y=1)p(y=1) > p(x|y=0)p(y=0)$ (vì phần mẫu số giống nhau).

Theo thuật toán Naive Bayes ta có:

$$p(x|y=1) = \prod_{j=1}^{n} p(x_j|y=1) = \prod_{j=1}^{n} x_j\phi_{j|y=1} + (1-x_j)(1-\phi_{j|y=1})$$

$$p(y=1) = \phi$$

$$p(x|y=0) = \prod_{j=1}^{n} p(x_j|y=0) = \prod_{j=1}^{n} x_j\phi_{j|y=0} + (1-x_j)(1-\phi_{j|y=0})$$

$$p(y=0) = 1-\phi$$

Tuy nhiên, nếu ta tính trực tiếp $p(x|y=1)p(y=1)$ và $p(x|y=0)p(y=0)$ thì có một vấn đề là: có thể xảy ra **tràn số nhỏ** vì $p(x|y=1)p(y=1)$ và $p(x|y=0)p(y=0)$ bao gồm tích của rất nhiều xác xuất (nhớ là $p(x|y=1) = \prod_{j=1}^{n} p(x_j|y=1)$ và $p(x|y=0) = \prod_{j=1}^{n} p(x_j|y=0)$). Để giải quyết vấn đề này, ta sẽ tính $\log(p(x|y=1)p(y=1))$ và $\log(p(x|y=0)p(y=0))$ rồi so sánh hai giá trị log này với nhau.

Ta có:

$$\log(p(x|y=1)p(y=1)) = \log p(x|y=1) + \log p(y=1) = \left[\sum_{j=1}^{n} \log p(x_j|y=1)\right] + \log p(y=1)$$

$$\log(p(x|y=0)p(y=0)) = \log p(x|y=0) + \log p(y=0) = \left[\sum_{j=0}^{n} \log p(x_j|y=0)\right] + \log p(y=0)$$

Công việc của các bạn trong phần này là viết file `predictSpam.m` ứng với hàm dự đoán với một tập các email mới. Hàm này có dạng:

```
function predictions = predictSpam(phi, phi0, phi1, Xtest)
```

Trong đó:

- `phi, phi0, phi1`: các tham số huấn luyện.

- `Xtest`: tập các email cần dự đoán (xem lại mục 1.2).
- `predictions`: véc-tơ dự đoán. Chẳng hạn, tập email cần dự đoán gồm có 1000 email thì `predictions` sẽ là một véc tơ cột gồm có 1000 thành phần, trong đó thành phần thứ i sẽ có giá trị là 1 nếu ta dự đoán email thứ i là spam và sẽ có giá trị là 0 nếu ngược lại.

4 Đánh giá chất lượng bộ phân lớp

Sau khi học (huấn luyện), để đánh giá chất lượng của bộ phân lớp spam/non-spam, ta sẽ cho nó dự đoán với các email của tập test. Sau đó, ta so sánh kết quả dự đoán với kết quả đúng (đáp án) và tính ra độ chính xác dự đoán. Ví dụ, tập test gồm có 1000 email và bộ phân lớp của ta dự đoán đúng 900 email thì độ chính xác là $\frac{900}{1000} = 0.9$.

Công việc của các bạn trong phần này là viết file `testNB.m` ứng với hàm tính độ chính xác dự đoán (trong hàm này, các bạn sẽ dùng lại hàm `predictSpam` đã viết). Hàm này có dạng:

```
function acc = testNB(phi, phi0, phi1, Xtest, ytest)
```

Trong đó:

- `acc`: độ chính xác dự đoán.
- `phi, phi0, phi1`: các tham số huấn luyện.
- `Xtest`: các véc-tơ biểu diễn ứng với các email trong tập test (xem lại mục 1.2).
- `ytest`: nhãn của các email trong tập test (đáp án) (xem lại mục 1.2).

5 Qui định

- Những trường hợp giống bài nhau sẽ bị 0 điểm.
- Cấu trúc thư mục bài nộp: trong thư mục `<MSSV>` gồm có:
  - 3 file `*.m` ứng với 3 hàm ở trên.
  - File báo cáo `Report.pdf`. Bạn cần báo cáo những điểm sau:

* Ý nghĩa của làm trơn Laplace?
* Báo cáo kết quả chạy ra của bạn với những tham số sau: `phi`, `phi0` (báo cáo 10 phần tử đầu tiên), `phi1` (báo cáo 10 phần tử đầu tiên), `acc`.

Happy learning!