Mục lục

Lời mở đầu

Trong kỷ nguyên thông tin hiện nay, khi mà lượng thông tin được đưa lên mạng gần như là vô tận, vậy làm thế nào ta có thể tận dụng lượng thông tin đó chuyển hóa thành tri thức phục vụ cho cuộc sống. Với dữ liệu dạng văn bản-chiếm phần lớn thông tin trên mạng, câu trả lời chính là: Xử lý ngôn ngữ tự nhiên.

Với lý do trên báo cáo xin được trình bày một ứng dụng của xử lý ngôn ngữ tự nhiên: phân loại văn bản, cụ thể là ứng dụng học máy cho bài toán phân loại văn bản. Báo cáo gồm: kiến thức cơ bản về học máy, giới thiệu về tài toán phân loại văn bản và kết quả thu được.

Phần 1:
Chương trình training,
chuỗi semina về Machine learning

1.1 Linear regession, Ridge regession

1.1.1 Học có giám sát (Supervised Learning)

Bằng cách xây dựng một mô hình phù hợp với bộ dữ liệu đã biết (gồm dữ liệu vào và dữ liệu ra tương ứng, còn gọi là tập train) ta có thể đưa ra dự đoán cho dữ liệu mới chưa biết đầu ra.
Với biểu diễn toán học, học có giám sát là tìm một hàm $f(x)$ từ bộ dữ liệu $X = x_1, x_2, \ldots, x_N$ với bộ nhãn tương ứng $Y = y_1, y_2, \ldots, y_N$ sao cho:

$$y_i \approx f(x_i), \quad \forall i = 1, 2, \ldots, N$$

Mục đích là xấp xỉ hàm số $f(x)$ thật tốt để khi có một dữ liệu x mới, chúng ta có thể tính được nhãn tương ứng của nó$y = f(x)$ Nếu bộ nhãn $Y = y_1, y_2, \ldots, y_N$ là rời rạc và hữu hạn thì bài toán thuộc bài toán phân loại (classification) còn bộ nhãn là liên tục thì bài toán là hồi quy (regression).

1.1.2 Thuận toán Linear regession và Ridge regession.

Nếu ta giả thuyết hàm cần tìm $f(x)$ có dạng tuyến tính:

$$f(x) = w_0 + w_1 x_1 + \ldots + w_n x_n$$

Thì thuận toán Linear regession chính là đi tìm bộ tham số $W = (w_0, w_1, \ldots, w_n)^T$ Với bô dữ liệu train, sao cho sai số $|yz - f(z)|$ là nhỏ nhất cho các dữ liệu trong tương lai. Bằng biểu diễn toán học

ta được:

$$w^* = argmin_w \mathbb{L}(w)$$

với

$$\mathbb{L}(w) = \frac{1}{2N}\sum_{i=1}^{N}(y_i - x_i^T w)^2$$

Tìm nghiệm w^* bằng cách giải phương trình $\mathbb{L}(w)^{'} = 0$. Thu được: $w^* = (A^T A)^{-1}A^T y$ Trong đó A là ma trận dữ liệu cỡ mx$(n+1)$ mà hàng thứ i là $A_i = (1, x_{i1}, x_{i2}, \ldots, x_{in}), y = (y_1, y_2, \ldots, y_M)^T$. Chú ý giả thuyết $A^T A$ tồn tại nghịch đảo. Phương pháp trên có một số nhược điểm:

- Nếu $A^T A$ không tồn tại nghịch đảo thì không học được.
- Độ phức tạp tính toán lớn do phải tính ma trận nghịch đảo. Do đó không làm việc được nếu số chiều n lớn.
- Khả năng overfitting cao vì việc học hàm $f(x)$ chỉ quan tâm tối thiểu lỗi đối với tập học đang có.

Thuận toán Ridge regresion

Để giảm overfiting cũng như tránh trường hợp ma trận $A^T A$ không tồn tại nghịch đảo, ta thêm một đại lượng hiệu chỉnh $\lambda||w||_2^2$ Khi đó w được tính bằng công thức

$$w^* = (A^T A + \lambda \mathbb{I}_{n+1})^{-1}A^T y$$

1.2 Phân cụm K-means

1.2.1 Học không giám sát.

Khi dữ liệu huấn luyện chỉ gồm dữ liệu đầu vào mà không có đầu ra tương ứng. Khi đó nhiệm vụ là ta tìm được các thông tin, quan hệ, tính chất ẩn trong dữ liệu từ đó có thể đưa ra đự báo cho dữ liệu mới trong tương lai.

Một trong những ví dụ điển hình của học không giám sát là phân cụm, phân bộ dữ liệu thành các cụm sao cho dữ liệu trong một cụm có cũng tính chất nào đó mong muốn.

1.2.2 Phương pháp K-means

Với giả sử rằng mỗi điểm dữ liệu chỉ thuộc duy nhất một cụm, khi đó với với bộ dữ liệu đầu vào và số lượng cụm cần tìm ta sẽ tìm được tâm của mỗi cụm bằng phương pháp K-means.

Bằng biểu diễn toán học, bộ dữ liệu $X = [x_1, x_2, ..., x_N]$ và $K < N$ là số cụm được xác định trước. Ta

cần tìm các tâm cụm $m_1, m_2, ..., m_K \in R^{d1}$ và với mỗi điểm dữ liệu x_i , ta cần tìm nhãn $y_i = k$ tương ứng, ở đây $k \in 1, 2, ..., K$.
Thuật toán của phương pháp K-means có 2 bước chính: Tính tâm từng cụm dựa vào các điểm thuộc cụm đó (ta có thể khởi tạo tâm ngẫu nhiên) sau đó gán lại nhãn cho mỗi điểm dữ liệu dựa trên tâm mới. Quá trình phân cụm kết thúc nếu một trong trong các điều kiện sau thỏa mãn:

- Số điểm dữ liệu thay đổi nhãn là không đáng kể
- Sự thay đổi tâm của mỗi cụm là không đáng kể
- Giảm không đáng kể về tổng lỗi phân cụm

$$Error = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, m_i)^2$$

  - C_i: Cụm thứ i
  - m_i: Tâm của cụm C_i
  - $d(x, m_i)$: Khoảng cách giữa điểm dữ liệu x và tâm m_i

Với ưu điểm đơn giản, dễ cài đặt, linh động (có thể dùng nhiều độ đo khác nhau tùy vào từng loại dữ liệu), K-means là phương pháp phân cụm được dùng phổ biến nhất.

1.3 Mạng neural nhân tạo.

Mạng neural nhân tạo (Artificial Neural network - ANN) là một mô hình toán học được xây dựng dựa trên mạng neural sinh học (bộ não). Nó được cấu tạo từ các nút (perceptron) nối với nhau nhằm đạt được một mục đính tính toán nào đó phù hợp với dữ liệu đưa vào.
Trong thực tế sử dụng, nhiều mạng neural là các công cụ mô hình hóa dữ liệu thống kê phi tuyến. Chúng có thể được dùng để mô hình hóa các mối quan hệ phức tạp giữa dữ liệu vào và kết quả hoặc để tìm kiếm các dạng/mẫu trong dữ liệu.
Khả năng tính toán của một ANN phụ thuộc vào: Kiến trúc của mạng neural, đặc tính của từng neural, thuật toán học và dữ liệu học.

Cấu trúc và hoạt động của một neural

Mỗi một neural (perceptron) hay một nút (node), có cấu trúc và cách thức hoạt động khá đơn giản: Với dữ liệu vào được cho qua hai hàm là hàm tổng hợp (thường là hàm tổng) và hàm kích hoạt sẽ

nhận được dữ liệu đầu ra.

$$a_j = \sum_{i=1}^{n} w_{ij} x_i + \theta_j$$
$$z_j = g(a_j)$$

- x_i là dữ liệu vào
- w_{ij} các trọng số tương ứng với x_i
- θ_j là hệ số điều chỉnh tại neural j
- z_j là đầu ra của neural
- $g(x)$ là hàm kích hoạt

Hàm kích hoạt

Hàm kích hoạt thường là một hàm phi tuyến, giúp tăng khả năng tính toán của ANN với những dữ liệu phức tạp hơn mức tuyến tính. Một số hàm kích hoạt phổ biến như: Sigmoid, Tanh, ReLu

Kiến trúc mạng

Một mạng neural có kiến trúc cơ bản gồm:

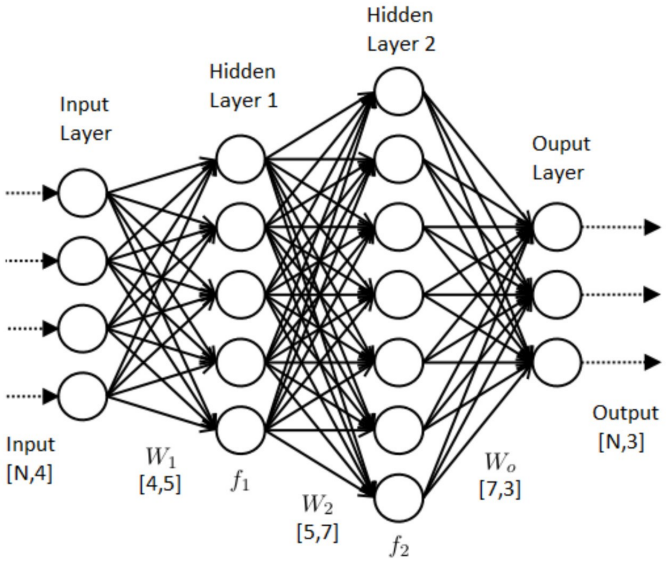
- Số lượng các tín hiệu đầu vào và đầu ra
- Số lượng các tầng
- Số lượng các neural trong mỗi tầng
- Số lượng các liên kết đối với mỗi nơ-ron
- Cách thức các neural (trong một tầng, hoặc giữa các tầng) liên kết với nhau

Một mạng khi ấy sẽ gồm nhiều tầng liên kết lần lượt, mỗi tầng chứa mốt số neural. Một ANN được gọi là liên kết đầy đủ (fully connected) nếu mọi đầu ra từ một tầng liên kết với mọi nơ-ron của tầng kế tiếp.

Huấn luyện mạng

Có 2 kiểu học trong mạng neural nhân tạo.

- Học tham số (Parameter learning): Mục tiêu là thay đổi thích nghi các trọng số (weights) của các liên kết trong mạng nearal



Hình 1.3.1: Mô hình mạng neural

- Học cấu trúc (Structure learning): Mục tiêu là thay đổi thích nghi cấu trúc mạng, bao gồm số lượng các nearal và các kiểu liên kết giữa chúng

Học cấu trúc thường phức tạp hơn khi thực hiện, báo cáo chỉ trình bày về học tham số.
Quá trình tính toán đầu ra cho một dữ liệu đi qua toàn bộ kiến trúc mạng được gọi là lan truyền thuận, được biểu diễn toán học như sau:

$$a^{(0)} = x$$
$$z^{(l)} = W^{(l)T} a^{l-1} + b^l, \quad l = 1, 2, ..., L$$
$$a(l) = f^{(l)}(z^{(l)}), \quad l = 1, 2, ..., L$$
$$\hat{y} = a^L$$

Với $l = 1, 2, ..., L$ là lần lượt các tầng.
Tương tự như các bài toán machine learning khác, ta định nghĩa hàm lỗi và tiến hành tìm tham số của ANN sao cho giá trị hàm lỗi là nhỏ nhất với dữ liệu học.

$$\mathbb{L}(w) = \frac{1}{N} \sum_{n=1}^{N} ||y_n - \hat{y}||_2^2$$

Theo các công thức này, việc tính toán trực tiếp các giá trị gradient tương đối phức tạp vì hàm mất mát không phụ thuộc trực tiếp vào các ma trận trọng số và vector điều chỉnh. Phương pháp phổ biến nhất được dùng có tên là lan truyền ngược (backpropagation) giúp tính gradient ngược từ tầng cuối cùng đến tầng đầu tiên. Tầng cuối cùng được tính toán trước vì nó ảnh hưởng trực tiếp tới đầu ra dự đoán và hàm mất mát. Việc tính toán gradient của các ma trận trọng số trong các tầng trước được

thực hiện dựa trên quy tắc chuỗi quen thuộc cho gradient của hàm hợp.

Phần 2:
Bài toán phân loại văn bản

2.1 Xử lý ngôn ngữ tự nhiên và bài toán phân loại văn bản.

Xử lý ngôn ngữ tự nhiên (natural language processing - NLP) là một nhánh của trí tuệ nhân tạo tập trung vào các ứng dụng trên ngôn ngữ của con người. Mục tiêu của lĩnh vực này là giúp máy tính hiểu và thực hiện hiệu quả những nhiệm vụ liên quan đến ngôn ngữ của con người như: tương tác giữa người và máy, cải thiện hiệu quả giao tiếp giữa con người với con người, hoặc đơn giản là nâng cao hiệu quả xử lý văn bản và lời nói.

Những bài toán cơ bản trong NLP

- **Truy xuất thông tin** (Information Retrieval – IR) có nhiệm vụ tìm các tài liệudưới dạng không có cấu trúc (thường là văn bản) đáp ứng nhu cầu về thông tin từ những nguồn tổng hợp lớn. Những hệ thống truy xuất thông tin phổ biến nhất bao gồm các công cụ tìm kiếm như Google, Yahoo, hoặc Bing search. Những công cụ này cho phép tiếp nhận một câu truy vấn dưới dạng ngôn ngữ tự nhiên làm đầu vào và cho ra một danh sách các tài liệu được sắp xếp theo mức độ phù hợp.

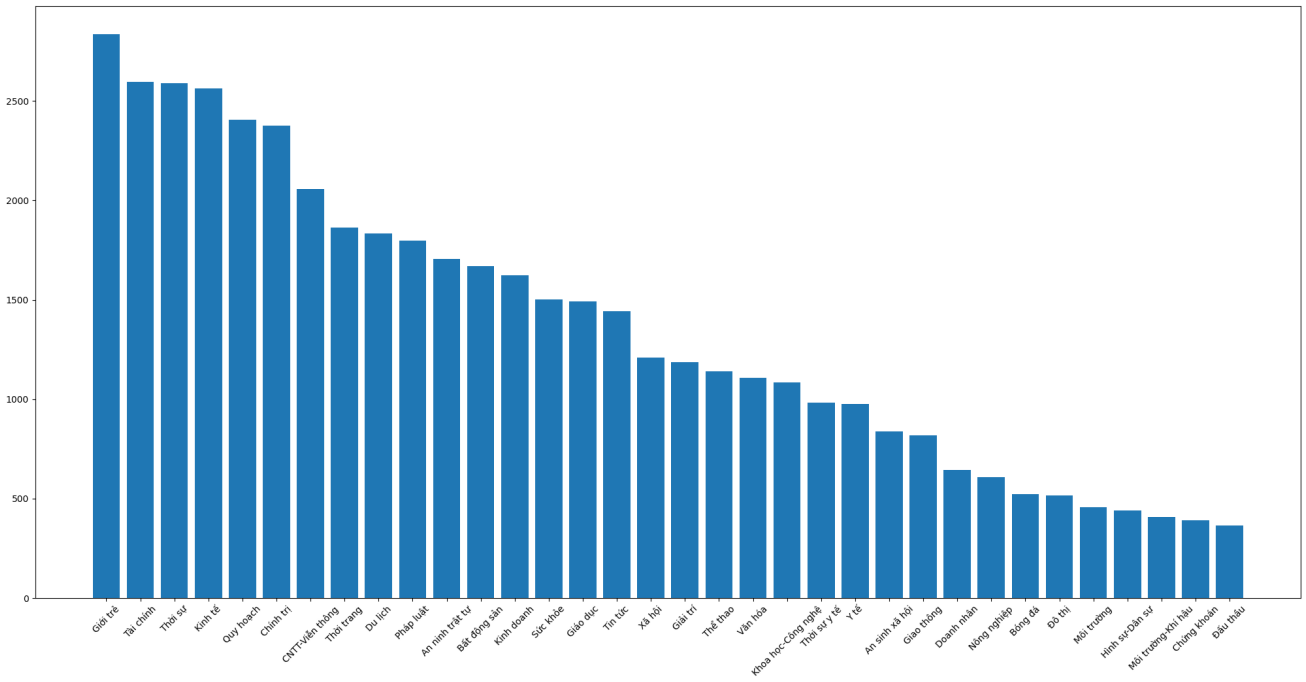
- **Trích chọn thông tin** (Information Extraction) nhận diện một số loại thực thể được xác định trước, mối quan hệ giữa các thực thể và các sự kiện trong văn bản ngôn ngữ tự nhiên. Khác với truy xuất thông tin trả về một danh sách các văn bản hợp lệ thì trích chọn thông tin trả về chính xác thông tin mà người dùng cần.

- **Trả lời câu hỏi** (QA) có khả năng tự động trả lời câu hỏi của con người ở dạng ngôn ngữ tự nhiên bằng cách truy xuất thông tin từ một tập hợp tài liệu. Một hệ thống QA đặc trưng thường bao gồm ba mô đun: Mô đun xử lý truy vấn (Query Processing Module) – tiến hành phân loại câu hỏi và mở rộng truy vấn; Mô đun xử lý tài liệu (Document Processing Module) – tiến hành

truy xuất thông tin để tìm ra tài liệu thích hợp; và Mô hình xử lý câu trả lời (Answer Processing Module) – trích chọn câu trả lời từ tài liệu đã được truy xuất.

- **Tóm tắt văn bản tự động** là bài toán thu gọn văn bản đầu vào để cho ra một bản tóm tắt ngắn gọn với những nội dung quan trọng nhất của văn bản gốc. Có hai phương pháp chính trong tóm tắt, là phương pháp trích xuất (extractive) và phương pháp tóm lược ý (abstractive).
- **Dịch máy** là việc sử dụng máy tính để tự động hóa một phần hoặc toàn bộ quá trình dịch từ ngôn ngữ này sang ngôn ngữ khác. Các phương pháp dịch máy phổ biến bao gồm dịch máy dựa trên ví dụ (example-based machine translation – EBMT), dịch máy dựa trên luật (rule-based machine translation – RBMT), và dịch máy thống kê (statistical machine translation – SMT).
- **Phân loại, gán nhãn văn bản** là bài toán xác định chủ đề chính hoặc liên quan của văn bản bằng các phương pháp học máy thông dụng hoặc deep learning.

Báo cáo sẽ trình bày về bài toán phân loại văn bản, bài toán với đầu vào là một đoạn văn bản bất kì có độ dài trung bình, Đầu ra của bài toán là nhãn tương ứng với văn bản đầu vào. Bộ dữ liệu training gồm nhiều đoạn văn bản đã có nhãn xác định như hình dưới.



2.2 Cơ sở lí thuyết

2.2.1 Tiền xử lý dữ liệu

Làm sạch dữ liệu

Với nhiệm vụ là phân loại cho văn bản, trước hết ta cần bằng cách nào đó chuyển dữ liệu dạng text thành số để có thể đưa vào các thuật toán phân loại, công việc sẽ trở nên dễ dàn hơn rất nhiều nếu dữ liệu text đã được làm sạch bằng tokenization loại bỏ kí tự đặc biệt, từ không quan trọng. Tokenization-đối với văn bản tiếng Việt ta cần tokeniza tức các từ ghép sẽ được gán lại với nhau. Có thể sử dụng thư viện ViTokenizer trong python.

Vector hóa dữ liệu

Có hai phương pháp phổ biến để Vector hóa dữ liệu là Bag of words và Tf-idf.

Với phương pháp Bag of words (BoW) mỗi văn bản sẽ được biểu diễn bằng một vector 1xn chiều với n là số từ trong từ điển (từ điển chỉ dành riêng cho bộ dữ liệu train) biểu diễn số lượng mỗi từ xuất hiện trong văn bản đó. Tuy nhiên với những văn bản có độ dài trên nhau lớn nhưng có cùng một nhãn thì Vector BoW có thể sẽ không phán ánh được điều đó, do đó thay vì số lượng ta có thể thay bằng tần suất một từ xuất hiện trong văn bản đó-Term frequency(Tf).

Cuối cùng, để xác định một từ có quan trọng cho việc phân loại hay không ta sẽ nhân Tf với tần số nghịch của mỗi từ tường ứng trong toàn bộ tập dữ liệu.

Ví dụ minh hoạ

- **Dữ liệu thô**

  "Người phát ngôn Bộ Ngoại giao Lê Thị Thu Hằng ˚Chiều 2/7, trả lời câu hỏi của phóng viên về việc Trung Quốc tiến hành tập trận ở khu vực quần đảo Hoàng Sa của Việt Nam, Người phát ngôn Bộ Ngoại giao Lê Thị Thu Hằng nêu rõ: ˚Ÿiệc Trung Quốc tập trận ở Hoàng Sa vi phạm chủ quyền của Việt Nam đối với quần đảo Hoàng Sa, đi ngược lại tinh thần tuyên bố Ứng xử của các bên ở Biển Đông (DOC), gây phức tạp tình hình không có lợi cho quá trình đàm phán hiện nay giữa Trung Quốc và ASEAN về Bộ Quy tắc ứng xử giữa các bên ở Biển Đông (COC) và việc duy trì môi trường hòa bình, ổn định, hợp tác ở biển Đông..."

- **Dữ liệu tiền xử lý**

  "người_phát_ngôn ngoại_giao lê_thị_thu hằng chiều trả_lời câu phóng_viên trung_quốc tiến_hành tập_trận khu_vực quần_đảo hoàng_sa việt_nam người_phát_ngôn ngoại_giao lê_thị_thu hằng nêu trung_quốc tập_trận hoàng_sa_vi_phạm chủ_quyền việt_nam quần_đảo hoàng_sa đi ngược_lại tinh_thần tuyên_bố ứng_xử biển_đông doc phức_tạp tình_hình lợi đàm_phán trung_quốc asean

quy_tắc ứng_xử biển_đông coc duy_trì môi_trường hòa bình ổn_định hợp_tác biển đông..."

- **BoW** Vị trí trong từ điển và giá trị BoW của từng từ:
  "102:1; 242:1; 246:2; 425:1; 616:1; 626:1; 910:1; 1178:1; 1395:1; 1584:1; 1704:1; 1893:1; 2299:1; 2551:1; 2730:4; 2802:1; 2874:1; 2904:1; 3034:3; 3160:1;... "
- **Tf-idf** Vị trí trong từ điển và giá trị Tf-idf của từng từ:
  "80:0.092; 204:0.057; 207:0.21; 387:0.074; 594:0.095; 602:0.094; 896:0.049; 1187:0.082; 1396:0.063; 1445:0.137; 1595:0.074; 1725:0.06; 1889:0.06; 2514:0.038; ... "

2.2.2 Các thuật toán sử dụng cho bài toán.

Naïve bayes

Với bài toán phân loại văn bản đã trình bày, giả sử $c \in \{1, \ldots, C\}$ là tập nhãn, x là vector hóa của một văn bản, khi đó x sẽ được phân vào lớp c với:

$$c = argmax_{c \in \{1,\ldots,C\}} p(c|x)$$

Nhìn chung, khó có cách tính trực tiếp $p(c|x)$. Thay vào đó, quy tắc Bayes thường được sử dụng:

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

$p(c)$ có thể được hiểu là xác suất để một điểm bất kỳ rơi vào nhãn c. Thành phần còn lại $p(x|c)$ là phân phối của các điểm dữ liệu trong nhãn c, nhằm đơn giản hóa việc tính toán người ta thường giả sử rằng các thành phần của biến ngẫu nhiên x độc lập với nhau khi đã biết c:

$$p(x|c) = p(x1, x2, ..., xd|c) = \prod_{i=1}^{d} p(xi|c)$$

Nhờ giả thiết độc lập mô hình có tốc độ huấn luyện và kiểm tra rất nhanh. Việc này rất quan trọng trong các bài toán với dữ liệu lớn, tuy nhiên lại đánh mất thông tin sự liên kết xuất hiện trong cùng một văn bản của các từ. Việc tính toán $p(xi|c)$ phụ thuộc vào loại dữ liệu. Có ba loại phân bố xác suất 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. Với mỗi chiều dữ liệu i và một nhãn c, xi tuân theo một phân phối chuẩn có kỳ vọng μ_{ci} và phương sai σ_{ci}^2:

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

Trong đó, bộ tham số $\theta = \mu_{ci}, \sigma_{ci}^2$ được xác định dựa trên các điểm trong tập huấn luyện thuộc nhãn c. Khi vecto hóa bằng Tf-idf ta sẽ cần sử dụng Gaussian naive Bayes

Multinomial Naive Bayes

Mô hình này chủ yếu được sử dụng trong bài toán phân loại văn bản mà vector đặc trưng được xây dựng dựa trên ý tưởng bag of words (BoW). Khi đó, p(xi ∣c) tỉ lệ với tần suất từ thứ i xuất hiện trong các văn bản có nhãn c. Giá trị này có thể được tính bởi

$$\chi_{ci} = p(x_i|c) = \frac{N_c i}{N_c}$$

Trong đó:

- $N_c i$ là tổng số lần từ thứ i xuất hiện trong các văn bản của nhãn c. Nó chính là tổng tất cả thành phần thứ i của các vector đặc trưng ứng với nhãn c
- N_c là tổng số từ, kể cả lặp, xuất hiện trong nhãn c. Nói cách khác, N_c là tổng độ dài của tất cả các văn bản thuộc nhãn c. Có thể suy ra rằng $N_c = \sum_{i=1}^{d} N_{ci}$, từ đó $\sum_{i=1}^{d} \chi_{ci} = 1$.

Bernoulli Naive Bayes

Mô hình này được áp dụng cho các loại dữ liệu mà mỗi thành phần là một giá trị nhị phân – bằng 0 hoặc 1. Ví dụ, cũng với loại văn bản nhưng thay vì đếm tổng số lần xuất hiện của một từ trong văn bản, ta chỉ cần quan tâm từ đó có xuất hiện hay không. Khi đó, $p(xi|c)$ được tính bởi

$$p(xi|c) = p(i|c)^{xi}(1p(i|c))^{1xi}$$

với $p(i|c)$ được hiểu là xác suất từ thứ i xuất hiện trong các văn bản của class c, xi bằng 1 hoặc 0 tuỳ vào việc từ thứ i có xuất hiện hay không.

Hồi quy Logistic

Tương tự với hồi quy tuyến tính, hồi quy Logistic chỉ khác ở chỗ thay vì đầu ra là hàm đồng nhất f(s) = s thì đầu ra sẽ là một hàm Logistic như hàm sigmoid hoặc tanh, ... Khi đó việc phân loại dữ liệu không thực sự tách biệt tuyến tính như dữ liệu văn bản tốt hơn.

Xây dựng hàm mất mát

Với các mô hình có hàm kích hoạt $f(s) \in (0, 1)$, ta có thể giả sử rằng xác suất để một điểm dữ liệu x_i có nhãn thứ nhất là $f(x_i^T w) = a_i$ nhãn còn lại là $1 - a_i$. Ta có thể viết:

$$p(y_i|x_i, w) = a_i^{y_i}(1 - a_i)^{1-y_i}, y_i = \{0, 1\}$$

Khi đó, cần tìm w sao cho:

$$w = argmax_w p(y|X, w)$$

Ta có thể giải bài toán này bằng cách giả sử các điểm dữ liệu độc lập

$$p(y|X, w) = \prod_{i=1}^{N} p(y_i|x_i, w) = \prod_{i=1}^{N} a_i^{y_i}(1 - a_i)^{1-y_i}$$

Lấy logarit tự nhiên, đổi dấu rồi lấy trung bình, ta thu được:

$$J(w) = \frac{1}{N} \log p(y|X, w) = -\frac{1}{N} \sum_{i=1}^{N} (y_i \log a_i + (1 - y_i) \log(1 - a_i))$$

Đến đây ta phải tìm w đề $J(w)$ nhỏ nhất bằng các phương pháp quen thuộc như Gradian Decent, Adam,... Tương tự như Linear Regresstion ta cũng có thể thêm một đại lượng hiệu chỉnh để tránh over fitting.

Máy Vector hỗ trợ

Máy Vector hỗ trợ (Support Vector Machine-SVM) là một phương pháp phân lớp tuyến tính (linear classifier), với mục đích xác định một siêu phẳng (hyperplane) để phân tách hai lớp của dữ liệu. khi dữ liệu không tách biệt tuyến tính ta có thể sử dụng các hàm hạt nhân (kernel functions) giúp biến đổi dữ liệu sao cho có thể tìm một siêu phẳng phân tách. SVM là một phương pháp tốt (phù hợp) đối với những bài toán phân lớp có không gian rất nhiều chiều (các đối tượng cần phân lớp được biểu diễn bởi một tập rất lớn các thuộc tính) được biết đến là một trong số các phương pháp phân lớp tốt nhất đối với các bài toán phân lớp văn bản (text classification).

Máy vector hỗ trợ lề cứng, lề mềm.

Gọi khoảng cách từ điểm gần nhất của một lớp tới siêu phẳng phân chia là lề (margin) khi đó siêu phẳng cần tìm sẽ có lề của hai lớp là như nhau. Hơn nữa, độ rộng của lề càng lớn thì khả năng phân loại càng tốt

Giả sử siêu phẳng phân chia có phương trình $w^T x + b = 0$, với cặp dữ liệu (x_n, y_n) bất kì có khoảng cách từ x_n tới siêu phẳng phân chia là $\frac{y_n(w^T x_n + b)}{||w||_2}$, lề khi đó sẽ được tính:

$$l = \min \frac{y_n(w^T x_n + b)}{||w||_2}$$

Bài toán tối ưu của SVM đi tìm w và b sao cho lề đạt giá trị lớn nhất:

$$(w, b) = \arg\max_{w,b} \left\{ \frac{1}{||w||_2} \min_n y_n(w^T x_n + b) \right\}$$

Nếu ta thay vector trọng số w bởi kw và b bởi kb trong đó k là một hằng số dương bất kỳ thì mặt phân chia không thay đổi, tức khoảng cách từ từng điểm đến mặt phân chia không đổi, tức lề không đổi.

Vì vậy, ta có thể giả sử:

$$y_m(w^T x_m + b) = 1$$

Như vậy, với mọi n ta luôn có

$$y_n(w^T x_n + b) \geq 1$$

Khi đó, ta có thể đưa về bài toán tối ưu có ràng buộc:

$$(w, b) = \arg\min_{w,b} \frac{1}{2}||w||_2^2$$

$$\text{Thỏa mãn } 1 - y_n(w^T x_n + b) \leq 0, \forall n = 1, 2, ..., N$$

Bài toán trên có thể giải bằng cách chuyển về dạng đối ngẫu, báo cáo sẽ không trình bày chi tiết.
Sau khi đã tìm được hệ số (w, b) của siêu phẳng $w^T + b = 0$, nhãn của một điểm bất kì mới sẽ được xác định bằng $(w^T + b)$
Với phân tính toán học trên máy vector chỉ làm việc khi dữ liệu tách biệt tuyến tính là được gọi là máy vector hỗ trợ lề cứng, để có thể làm việc với dữ liệu không tách biệt tuyến tính hoặc tách biệt nhỏ do nhiễu, ta sẽ sử dụng lề mềm. Từ mềm thể hiện sự linh hoạt, có thể chấp nhận một số điểm bị phân loại sai để mô hình hoạt động tốt hơn trên toàn bộ dữ liệu Như đã trình bày ở trên, máy vector hỗ trợ lề mềm ta cần chấp nhận phân loại sai một số điểm, nhưng sai bao nhiêu thì tốt nhất, ta cần có thêm một đại lượng đong đếm sự sai này:

$$\xi_i = |w^T x_i + b - y_i|$$

Với SVM lề mềm, hàm mục tiêu sẽ có thêm một số hạng nữa giúp tối thiểu tổng sự hy sinh. Từ đó ta có hàm mục tiêu:

$$\frac{1}{2}||w||_2^2 + C\sum_{n=1}^{N} \xi_n$$

trong đó C là một hằng số dương. Hằng số C được dùng để điều chỉnh tầm quan trọng giữa độ rộng lề và sự hy sinh. Điều kiện ràng buộc cũng được thay đổi so với SVM lề cứng. Với mỗi cặp dữ liệu (x_n, y_n), thay vì ràng buộc cứng $y_n(w^T x_n + b) \geq 1$, ta sử dụng ràng buộc mềm:

$$y_n(w^T x_n + b) \geq 1 - \xi_n \Leftrightarrow 1 - \xi_n - y_n(w^T x_n + b) \leq 0, \forall n = 1, 2, ..., n$$

Và ràng buộc phụ $\xi_n \geq 0, \forall n = 1, 2, ..., N$. Tóm lại, ta có bài toán tối ưu chính cho SVM lề mềm như sau:

$$(w, b) = \arg\min_{w,b} \frac{1}{2}||w||_2^2 + C\sum_{n=1}^{N} \xi_n$$

$$\text{Thỏa mãn: } 1 - \xi_n - yn(w^T xn + b) \leq 0, \forall n = 1, 2, ..., N$$

$$-\xi_n \leq 0, \forall n = 1, 2, ..., N$$

Bài toán trên cũng đc giải quyết bằng sử dụng bài toán đối ngẫu.

Máy vector hỗ trợ hạt nhân.

Ý tưởng cơ bản của SVM hạt nhân là tìm một phép biến đổi dữ liệu không tách biệt tuyến tính ở một không gian thành dữ liệu tách biệt hơn ở một không gian mới, sau đó có thể áp dụng SVM lề cứng hoặc mềm.

Hạt nhân ở đây là một hàm $k(x_i, x_j)$ mô tả quan hệ của hai điểm dữ liệu bất kì trong không gian mới. Một số hàm hạt nhân thông dụng như:

- Tuyến tính: Đây là trường hợp đơn giản với hàm hạt nhân chính là tích vô hướng của hai vector $k(x, z) = x^T z$.
- Đa thức: Hàm hạt nhân đa thức có dạng $k(x, z) = (r + \gamma x^T z)^d$ Với d là một số thực dương.
- Hàm cơ sở radial(radial basic function-RBF) có dạng: $k(x, z) = \exp(-\gamma ||x - z||_2^2), \gamma > 0$
- Hàm dạng Sigmoid: $k(x, z) = tanh(\gamma x^T z + r)$

2.3 Kết quả.

Kết quả độ chính xác (accuracy) trên tập test

	Số lượng từ vựng	Naive Bayes	Logistic Regression	SVM(linear kernel)
BoW	10k	64%	78.3%	80.14%
	15k	64.92%	81.29%	80.56%
	25k	66.57%	81.62%	80.49%
	30k	66.29%	81.91%	80.53%
Tf-idf	10k	79.79%	80,084%	83.604%
	15k	—[1]	80.33%	83.9%
	25k	—	81.33%	84.25%
	30k	—	81.34%	84.32%

Qua bảng trên ta có thể thấy SVM có kết quả tốt nhất, tuy nhiên như đã phân tích thời gian chạy là nhiều hơn hẳn hai phương pháp còn lại.

[1]Không tính được do tràn dữ liệu

Tổng kết

Sau quá trình thực tập ở công ty, em đã phần nào tìm hiểu về xử lý ngôn ngữ tự nhiên và ứng dụng trong bài toán thực tế, ngoài ra còn được học hỏi rất nhiều từ kiến thức cho đến phong cách làm việc, ứng xử từ các anh/chị tại công ty.
Báo cáo còn nhiêu thiếu sót, em mong nhân được sự góp ý của thầy cô, các anh/chị và các bạn để báo cáo được hoàn chỉnh hơn.
Em xin chân thành cảm ơn!

Tài liệu tham khảo

Vũ Hữu Tiệp, *Machine Learning cơ bản*, 20/01/2020.

Ngô Văn Linh, *Slide bài giảng về Machine learning*.

Internet.