ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

PHẠM NGUYÊN BÌNH

ỨNG DỤNG MÔ HÌNH MAXIMUM ENTROPY TRONG PHÂN LỚP QUAN ĐIỂM CHO DỮ LIỆU VĂN BẢN

LUẬN VĂN THẠC SĨ KỸ THUẬT PHẦN MỀM

Hà Nội – 2016

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

PHẠM NGUYÊN BÌNH

ỨNG DỤNG MÔ HÌNH MAXIMUM ENTROPY TRONG PHÂN LỚP QUAN ĐIỂM CHO DỮ LIỆU VĂN BẢN

Ngành: Công nghệ thông tin
Chuyên ngành: Kỹ thuật phần mềm
Mã số: 60480103

LUẬN VĂN THẠC SĨ KỸ THUẬT PHẦN MỀM

NGƯỜI HƯỚNG DẪN KHOA HỌC: PGS.TS. PHẠM BẢO SƠN

Hà Nội – 2016

Lời cam đoan

Tôi xin cam đoan đây là công trình nghiên cứu khoa học của riêng tôi và được sự hướng dẫn khoa học của PGS.TS. Phạm Bảo Sơn. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong luận văn còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung luận văn của mình.

Học viên Cao học

Phạm Nguyên Bình

Lời cảm ơn

Trước tiên, tôi xin bày tỏ sự biết ơn chân thành và sâu sắc nhất tới PGS. TS. Phạm Bảo Sơn – Giáo viên hướng dẫn trực tiếp của tôi, người đã hết lòng hỗ trợ và giúp đỡ tôi trong quá trình nghiên cứu và hoàn thiện luận văn thạc sĩ của mình.

Tôi cũng xin gửi lời cảm ơn chân thành tới các thầy, các cô là giảng viên của trường Đại học Công nghệ đã tận tình dạy dỗ và hướng dẫn cho tôi trong suốt quá trình học tập thạc sĩ tại trường.

Và tôi cũng xin gửi lời cảm ơn tới bố mẹ, vợ và những người thân trong gia đình vì đã nuôi nấng, dạy dỗ, chăm lo cho tôi, động viên tôi hoàn thành thật tốt khóa học thạc sĩ này.

Mặc dù đã hết sức cố gắng hoàn thành luận văn nhưng chắc chắn sẽ không tránh khỏi những sai sót. Kính mong nhận được sự cảm thông, chỉ bảo tận tình của các quý thầy cô và các bạn.

Tôi xin chân thành cảm ơn!

Mục lục

Danh sách hình vẽ

Danh sách bảng biểu

MỞ ĐẦU

1. Tính cấp thiết của đề tài luận văn

Ngày nay, xã hội của chúng ta đang chứng kiến sự bùng nổ của Internet và đặc biệt là sự phát triển đến chóng mặt của các mạng xã hội như Facebook, Twitter cũng như các diễn đàn, các trang thông tin mạng về đa dạng các lĩnh vực. Chúng ta thường gọi chúng với tên chung là các kênh truyền thông xã hội trực tuyến (social media online). Trên các kênh truyền thông này là một lượng dữ liệu về quan điểm, ý kiến khổng lồ (big data) tới trực tiếp từ hàng trăm triệu người dùng trong nước cũng như quốc tế. Vì lẽ đó, việc giám sát thương hiệu thông qua thu thập, phân tích những phản hồi, ý kiến, đóng góp của người sử dụng trên những kênh truyền thông này là vô cùng quan trọng và hữu ích với các công ty, doanh nghiệp và các tổ chức nói chung. Việc thu thập và xử lý kịp thời các thông tin này sẽ hỗ trợ tích cực cho các công ty, doanh nghiệp và tổ chức thực hiện được: (I) nắm bắt được mức độ phổ biến, lan tỏa và tầm ảnh hưởng của thương hiệu; (II) nắm bắt được tâm tư, nguyện vọng và cả những phản hồi, góp ý trực tiếp từ cộng đồng, những người sử dụng dịch vụ để từ đó đưa ra những điều chỉnh phù hợp; (III) nắm bắt và hiểu được những phản hồi và bình luận trên diện rộng đối với các vấn đề, sự kiện quan trọng của tổ chức; (IV) kịp thời bảo vệ thương hiệu của đơn vị trước những thông tin dư luận thiếu chính xác và sai lệch.

Chính vì lẽ đó, việc phát triển một hệ thống có thể tự động thu thập, phân tích và tổng hợp dữ liệu truyền thông là vô cùng cần thiết và hữu ích đối với sự phát triển của bất cứ một công ty, doanh nghiệp hay tổ chức nào, trong đó có cả Đại học Quốc gia (ĐHQG) Hà Nội. Mục tiêu của nhóm đề tài là xây dựng hệ thống tự động phân tích dữ liệu truyền thông xã hội trực tuyến phục vụ quản lý và hỗ trợ ra quyết định, kinh tế, chính trị, giáo dục và xã hội cho Đại học Quốc gia Hà Nội với tên gọi VNU-SMM (Vietnam National University-Social Media Monitoring).

2. Mục tiêu của luận văn

Luận văn tập trung vào tìm hiểu các mô hình học máy có giám sát phổ biến, được ứng dụng trong bài toán phân lớp quan điểm người dùng cho dữ liệu văn bản thu được từ các kênh truyền thông xã hội. Trong luận văn, chúng tôi cũng đã lựa chọn bộ phân lớp Maximum Entropy để cài đặt và thử nghiệm, đồng thời ứng dụng vào hệ thống tự động phân tích dữ liệu truyền thông xã hội trực tuyến phục vụ quản lý và hỗ trợ ra quyết định trong lĩnh vực đào tạo cho Đại học Quốc gia Hà Nội.

3. Cấu trúc của luận văn

Luận văn được tổ chức thành năm chương. Trong chương 1, chúng tôi sẽ giới thiệu về bài toán phân lớp quan điểm người dùng, các hướng tiếp cận và các giải pháp

đã và đang được nghiên cứu, sử dụng trên thế giới. Trong chương tiếp theo, chúng tôi sẽ mô tả tổng quan về hệ thống tự động thu thập và phân tích dữ liệu truyền thông xã hội trực tuyến cho Đại học Quốc gia Hà Nội - VNU-SMM và vai trò của thành phần phân lớp quan điểm người dùng trong hệ thống. Nội dung chi tiết về bộ phân lớp Maximum entropy và ứng dụng của nó trong bài toán phân tích quan điểm người dung sẽ được chúng tôi trình bày trong chương 3. Trong chương 4, chúng tôi sẽ tập trung trình bày về kết quả thực nghiệm, sau đó đánh giá, phân tích kết quả, những lỗi và điểm yếu còn tồn tại. Cuối cùng, chúng tôi sẽ tổng kết lại những nội dung đã thực hiện trong luận văn, từ đó đề xuất hướng nghiên cứu và phát triển trong tương lai.

Chương 1
Bài toán phân lớp quan điểm và các hướng tiếp cận

1.1 Bài toán phân tích quan điểm

Phân tích quan điểm (opinion mining hay sentiment analysis) là một lĩnh vực nghiên cứu về các ý kiến, quan điểm, đánh giá, thái độ và cảm xúc của mọi người về một đối tượng. Các đối tượng ở đây có thể là các cá nhân, các sự việc, sự vật, các dịch vụ, sản phẩm, các công ty, tổ chức, hoặc một chủ đề bất kỳ. Hai thuật ngữ Opinion Mining (OM) và Sentiment Analysis (SA) có thể được sử dụng thay thế cho nhau trong các ngữ cảnh sử dụng. Tuy nhiên, một số nhà nghiên cứu cho rằng OM và SA có một điểm khác nhau nhỏ [14]. Trong khi OM trích xuất và phân tích các ý kiến của về một đối tượng thì SA cần phải xác định các ý kiến từ một văn bản trước khi tiến hành trích xuất và phân tích chúng.

Phân tích quan điểm là một lĩnh vực thu hút được sự quan tâm lớn của cộng đồng nghiên cứu nói chung và cộng đồng xử lý ngôn ngữ nói riêng bởi ba yếu tố chính sau:

Thứ nhất, đó là sự đa dạng trong ứng dụng của nó vào nhiều lĩnh vực. Ví dụ như trong kinh doanh, việc phân tích và nắm được các ý kiến, quan điểm của khách hàng có thể giúp các công ty, tập đoàn xây dựng được những sản phẩm chất lượng cao, đáp ứng được nhu cầu của người dùng, có thể đưa ra những giải pháp kịp thời cho các vấn đề liên quan đến sản phẩm, chăm sóc khách hàng,.. trước khi mọi chuyện diễn biến xấu đi vượt tầm kiểm soát hoặc thậm chí giúp dự đoán sản lượng bán hàng trong thời gian tới. Hoặc như trong giáo dục, việc nắm được tâm tư, nguyện vọng, hoặc các đánh giá, góp ý quý báu của các phụ huynh, học sinh giúp cho các trường, cho các Bộ, Sở giáo dục hiệu chỉnh hệ thống đào tạo hiện thời, giúp cho người dạy và người học đạt chất lượng tốt nhất.

Thứ hai, đó là sự bùng nổ của thông tin và mạng xã hội. Trong lịch sử loài người, đây là thời điểm mà lượng thông tin, lượng quan điểm trên mạng internet đang ngày càng trở nên khổng lồ, cung cấp những dữ liệu phong phú, thời gian thực, đa dạng mà không có nó, việc nghiên cứu, phân tích quan điểm người dùng là vô nghĩa. Cộng đồng người dùng Internet ngày càng phát triển và hoạt động tích cực trên các kênh mạng xã hội như Facebook, Twitter, forums, các trang báo,.. với rất nhiều các ý kiến, quan điểm riêng về mọi vấn đề trong xã hội. Tuy nhiên, vấn đề đặt ra là: mặc dù kho dữ liệu khổng lồ này (big data) chứa rất nhiều thông tin, bên cạnh các thông tin hữu ích, được các cá nhân, công ty, tổ chức quan tâm là các thông tin rác không cần

thiết. Bài toán đặt ra là làm sao có thể lọc được các thông tin hữu ích này từ kho dữ liệu khổng lồ đó.

Thứ ba, đó là sự thách thức của bài toán. Phân tích quan điểm người dùng có thể được chia ra làm nhiều bài toán nhỏ hơn và cũng đầy thách thức với các nhà nghiên cứu như các bài toán phân lớp chủ quan và khách quan (subjectivity classification), phân lớp ý kiến trái chiều (sentiment polarity classification), phát hiện ý kiến rác (spam opinion detection) [10], tóm tắt và tổng hợp quan điểm (opinion summarization), phân tích tính đa diện của của một ý kiến (dual sentiment analysis) [15],…

Quan điểm được chia làm hai loại: tích cực (positive) và tiêu cực (negative). Ngoài hai trạng thái này, một câu hoặc văn bản được xếp vào dạng trung lập (neutral).

Bài toán phân tích quan điểm người dùng thường được tiếp cận và giải quyết ở ba mức độ [5]:

- **Mức độ văn bản, tài liệu** (Document level): ở mức độ này, bài toán cần phân loại xem một văn bản hay tài liệu thể hiện ý kiến tiêu cực hay tích cực. Ví dụ như một bài viết phân tích, đánh giá về kỳ thi đánh giá năng lực do Đại học Quốc gia tổ chức thể hiện ý kiến, nhận định chủ yếu là tốt hay không tốt, tích cực hay tiêu cực. Mức độ này được thực hiện với giả sử rằng tài liệu chỉ đưa ra các quan điểm, ý kiến về một thực thể duy nhất chứ không có sự so sánh giữa các thực thể khác nhau.
- **Mức độ câu** (Sentence level) [3]: các phương pháp được áp dụng cho mức độ tài liệu cũng có thể được áp dụng ở mức độ câu. Trong trường hợp đơn giản, các câu chỉ chứa một ý kiến, quan điểm về một thực thể. Trong các trường hợp phức tạp hơn, một câu có thể có nhiều quan điểm, đánh giá về các khía cạnh khác nhau của một đối tượng hoặc thậm chí có thể có sự thay đổi về quan điểm trong cùng một câu (polarity shifting) [16]. Mức độ phân tích quan điểm cho câu rất gần với bài toán phân lớp chủ quan và khách quan, trong đó chúng ta cần phân loại xem một câu đã cho là chủ quan (có quan điểm, ý kiến riêng) hay khách quan (câu chỉ đưa ra thông tin). Tuy nhiên, các câu khách quan cũng có thể từ đó suy ra quan điểm. Ví dụ như câu: Cơ sở hạ tầng của trường vừa được xây mới một năm nay đã trở nên xập xệ, tồi tàn. Trong câu nói này, cả hai mệnh đề đều là sự việc khách quan trong thực tế nhưng từ đó có thể suy luận ra ý kiến chê bai chất lượng cơ sở vật chất và cũng như cách quản lý chưa sát sao của nhà trường.
- **Mức độ khía cạnh** (Aspect level): nếu với hai mức độ nêu trên, vấn đề được tiếp cận theo hướng kiến trúc của văn bản, ngôn ngữ (câu, đoạn, tài liệu, cú pháp), thì ở mức độ khía cạnh, bài toán tập trung vào chính quan điểm, ý kiến

được đưa ra, phân tích ở mức độ sâu hơn, đó là phân tích xem ý kiến tiêu cực hay tích cực của là về chủ đề, đối tượng nào [4].

Bảng 1.1 tóm tắt lại ba mức độ của phân tích quan điểm người dùng cùng các nhiệm vụ mà mỗi mức độ cần giải quyết:

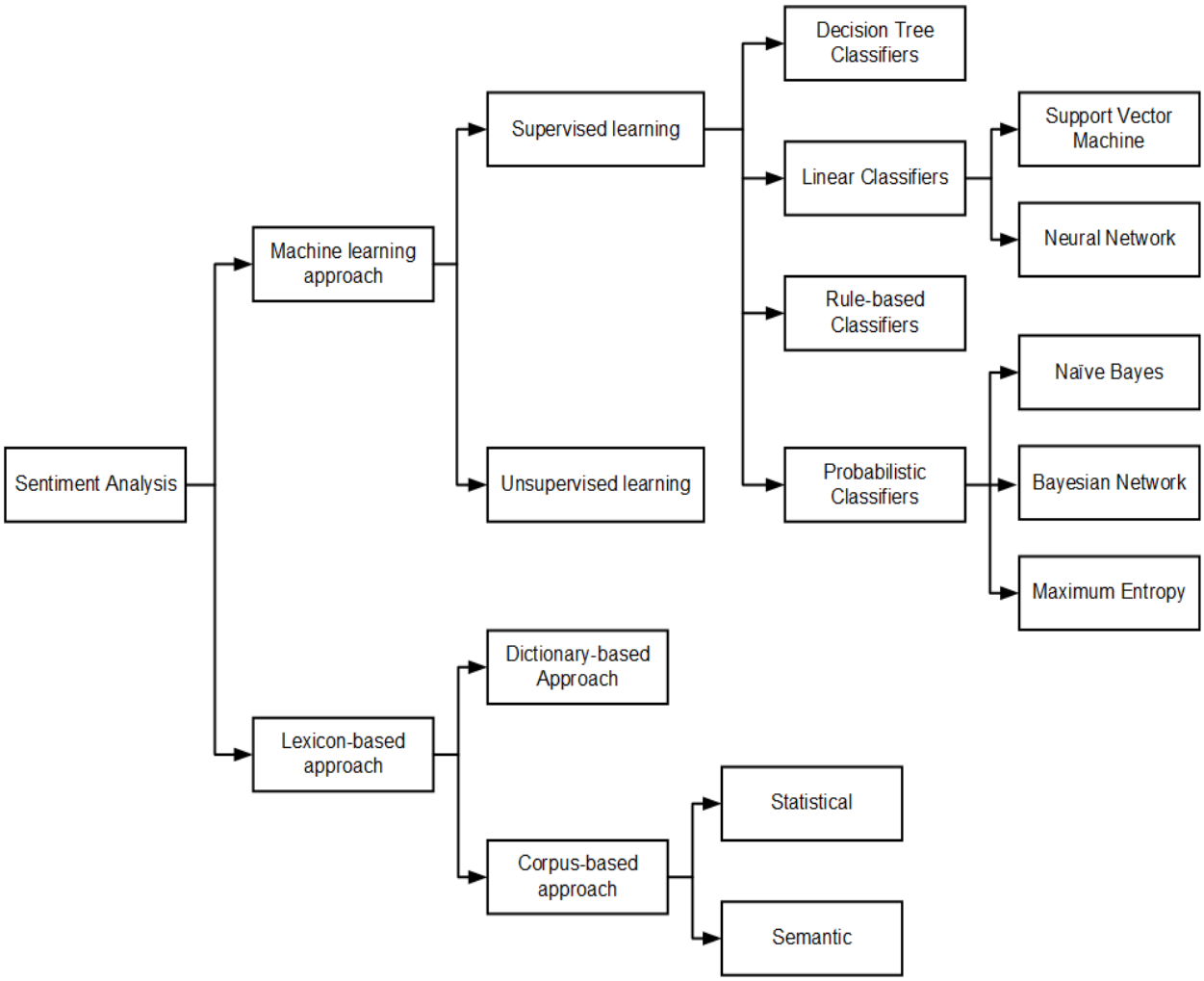
Bảng 1.1: Các mức độ trong phân tích quan điểm

Mức độ	Giả thiết đi kèm	Các nhiệm vụ cần giải quyết
Mức độ câu	1. Mỗi câu chỉ chứa quan điểm từ một người duy nhất. Tuy nhiên, giả thiết này không đúng trong nhiều trường hợp như câu phức hoặc các câu ghép 2. Các câu được phân tách rõ ràng trong văn bản	1. Xác định xem một câu là chủ quan hay khách quan. 2. Phân loại quan điểm của câu, xác định xem câu đã cho là tích cực, tiêu cực hay trung lập.
Mức độ văn bản/tài liệu	1. Mỗi văn bản chỉ tập trung vào một đối tượng duy nhất và chỉ chứa quan điểm của một người 2. Không thể áp dụng cho các bài viết trên blog, forum do có thể chứa nhiều bài viết về nhiều đối tượng trong những nguồn thông tin này	1. Phân loại quan điểm của câu, xác định xem câu đã cho là tích cực, tiêu cực hay trung lập.
Mức độ khía cạnh	1. Nguồn thông tin tập trung vào các khía cạnh của một đối tượng và chỉ chứa quan điểm của một người 2. Không thể áp dụng cho các bài viết trên blog, forum do có thể chứa nhiều bài viết về nhiều đối tượng trong những nguồn thông tin này	1. Xác định các đặc trưng của đối tượng được người viết đề cập tới 2. Xác định quan điểm của đặc trưng được đề cập đến (tích cực, tiêu cực hay trung lập) 3. Nhóm các từ đồng nghĩa chỉ cùng một đặc trưng. Sinh ra một tóm tắt các quan điểm về các đặc trưng từ nhiều đánh giá khác nhau.

1.2 Các hướng tiếp cận và giải quyết bài toán

Trong những năm gần đây, có rất nhiều bài báo và các công trình nghiên cứu cải tiến các thuật toán phân tích quan điểm người dùng. Các kỹ thuật này có thể được phân loại như trong Hình 1.1 [7]. Trong đó ta thấy, có hai hướng tiếp cận chính trong các kỹ thuật ứng dụng trong giải quyết bài toán phân lớp quan điểm người dùng, đó là: sử dụng các thuật toán học máy hoặc tiếp cận theo hướng sử dụng các kiến thức về từ vựng và ngữ nghĩa. Trong các thuật toán học máy lại có thể được chia ra thành các thuật toán học có giám sát hay học không giám sát. Ngoài ra, trong một, hai năm trở lại đây bắt đầu xuất hiện các ứng dụng thành công của deep learning vào trong bài toán phân tích quan điểm [12,13] đạt kết quả cao.

Các thuật toán học máy có giám sát phổ biến được sử dụng trong giải quyết bài toán phân lớp quan điểm là: Naïve Bayes, Maximum Entropy, Support Vector Machine (SVM) [9]. Các thuật toán này được đánh giá cao về tính chính xác và hiệu quả trong giải quyết bài toán phân lớp quan điểm người dùng. Trong mục này, chúng tôi sẽ giới thiệu tổng quan về các giải thuật học có giám sát này.



Hình 1.1: Các kỹ thuật sử dụng trong giải quyết bài toán phân lớp quan điểm

Một cách tổng quát, các bộ phân lớp thường gồm các bước chính sau:

- **Bước 1:** Tiền xử lý dữ liệu: tại bước này, dữ liệu sẽ được làm sạch (data cleaning), và chuẩn hóa (data normalization), làm đầu vào cho bước tiếp theo.
- **Bước 2:** Trích trọn đặc trưng và rút gọn đặc trưng (để giảm độ phức tạp).
- **Bước 3:** Xây dựng mô hình học
- **Bước 4:** Phân lớp
- **Bước 5:** Hậu xử lý kết quả phân lớp

Các giải thuật học máy có giám sát đều có hướng tiếp cận chung như sau:

Đầu vào của giải thuật gồm:

- d: tài liệu cần phân loại
- C: tập xác định các phân lớp cho tài liệu. C={c_1, c_2,…,c_n}. Trong bài toán phân tích quan điểm C = {tích cực, tiêu cực, khác}
- tập dữ liệu huấn luyện với các tài liệu đã được gán nhãn, phân loại thủ công

Đầu ra: bộ phân lớp đã học xong.

Trong các mục tiếp theo, chúng tôi sẽ trình bày giới thiệu về ba mô hình phân lớp phổ biến thường được sử dụng trong phân lớp quan điểm người dung cho dữ liệu văn bản.

1.3 Mô hình phân lớp Naïve Bayes

Bộ phân lớp quan điểm Naïve Bayes được xây dựng dựa trên lý thuyết Bayes về xác suất có điều kiện và sử dụng mô hình "bag of words" để phân loại văn bản:

$$P(\mathrm{c} \mid \mathrm{d}) = \mathrm{P(c)}.\frac{P(d \mid c)}{P(d)} \tag{1.1}$$

Mục tiêu là tìm được phân lớp c* sao cho P(c*|d) là lớn nhất hay xác suất của tài liệu d thuộc lớp c* là lớn nhất.

Từ công thức trên ta có thể nhận thấy P(d) không đóng vai trò gì trong việc quyết định phân lớp c → P(c|d) lớn nhất ⟺ P(c).P(d|c) lớn nhất.

Để có thể xấp xỉ giá trị của P(d|c), thuật toán Naïve Bayes giả sử rằng: các vector đặc trưng f_i của một tài liệu khi đã biết phân lớp là độc lập với nhau. Từ đó ta có công thức:

$$\begin{aligned} &\mathrm{P(c|d)\ max = argmax\ P(c).P(f_1, f_2,\ldots,f_n|c)} \\ \Leftrightarrow\ &\mathrm{P(c|d)\ max = argmax\ P(c).} \prod_{1\le i\le n} P(f_i \mid c) \\ \Leftrightarrow\ &\mathrm{P(c|d)\ max = argmax(log\ P(c) + } \sum_{1\le i\le n} \mathrm{logP}(f_i \mid c)\ ) \end{aligned} \tag{1.2}$$

Trong đó f là các vector đặc trưng cho tài liệu d.

Khi tiến hành huấn luyện, thuật toán sử dụng phương pháp xấp xỉ hợp lý cực đại MLE (Maximum Likelihood Estimation) để xấp xỉ P(c) và $P(f_i|c)$ cùng thuật toán làm mịn add-one (add-one smoothing). Ta có:

$$P(c) = \frac{N_c}{N} \tag{1.3}$$

Trong đó N_c là số văn bản được phân loại vào lớp c; N là tổng số văn bản trong tập huấn luyện.

$$P(f_i \mid c) = \frac{N_{cf_i}}{\sum_{f \in F} N_{cf}} \quad (1.4)$$

Trong đó N_{cf_i} là số lần xuất hiện của vector đặc trưng i trong tài liệu thuộc phân lớp *c*.

Đánh giá bộ phân lớp sử dụng thuật toán học máy Naive Bayes, ta nhận thấy phương pháp này các ưu điểm như: đơn giản, dễ cài đặt, bộ phân lớp chạy nhanh và cần ít bộ nhớ lưu trữ. Bộ phân lớp cũng không cần nhiều dữ liệu huấn luyện để xấp xỉ được bộ tham số. Tuy nhiên, bộ phân lớp này có nhược điểm là thiếu chính xác do giả thiết độc lập của các vector đặc trưng khi đã biết phân lớp là không có thực trong thực tế.

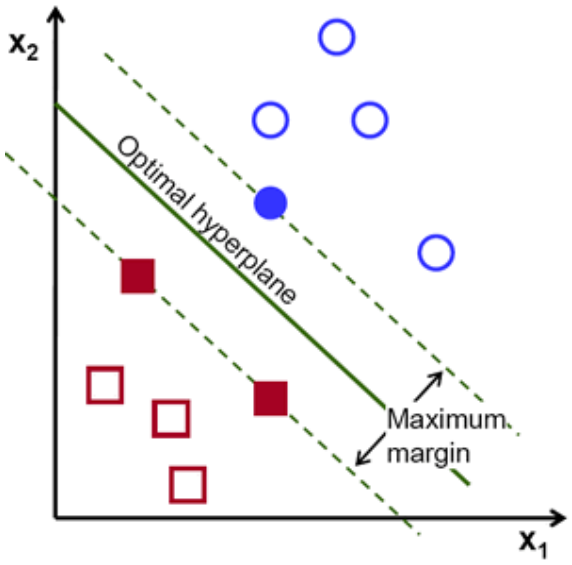
1.4 Mô hình phân lớp SVM

1.4.1 Giới thiệu về SVM

Máy vector hỗ trợ (Support Vector Machine – SVM) là một phương pháp học máy nổi tiếng được sử dụng để giải quyết bài toán phân lớp, thuật toán được Vladimir N. Vapnik tìm ra và thuật toán SVM tiêu chuẩn hiện nay sử dụng được tìm ra bởi Vapnik và Corinna Cortes vào năm 1995. SVM xuất phát từ lý thuyết học thống kê, dựa trên nguyên tắc tối thiểu rủi ro cấu trúc (Structural Risk Minimisation) và cố gắng phân loại dữ liệu sao cho lỗi xảy ra trên tập kiểm tra là nhỏ nhất (Test Error Minimisation). Nhiều bài toán trong đời sống thực được SVM giải quyết khá thành công như nhận dạng văn bản, hình ảnh, chữ viết tay, phân loại thư rác điện tử, virus…

Thuật toán SVM ban đầu chỉ được thiết kế để giải quyết bài toán phân lớp nhị phân, tức là số lớp hạn chế là hai lớp, với ý tưởng chính như sau:

Cho trước một tập huấn luyện, được biểu diễn trong không gian vector với mỗi điểm là biểu diễn của một dữ liệu, SVM sẽ tìm ra một siêu phẳng f quyết định tốt nhất có thể chia các điểm trên không gian này thành hai lớp riêng biệt, tương ứng là lớp "+" và lớp "-". Chất lượng của siêu phẳng được đánh giá bởi khoảng cách lề (margin) giữa hai lớp: khoảng cách càng lớn thì siêu phẳng quyết định càng tốt và chất lượng phân lớp càng cao.



Hình 1.2: Ví dụ về siêu phẳng trong SVM

Trong ví dụ như Hình 1.2, đường thẳng liền nét tô màu xanh lá chính là siêu phẳng tốt nhất để phân tách dữ liệu thành hai lớp khác nhau. Hai bên của siêu phẳng là hai lề, chứa các vector hỗ trợ (support vectors) – tức là các điểm dữ liệu gần siêu phẳng nhất.

1.4.2 Bài toán phân lớp nhị phân với SVM

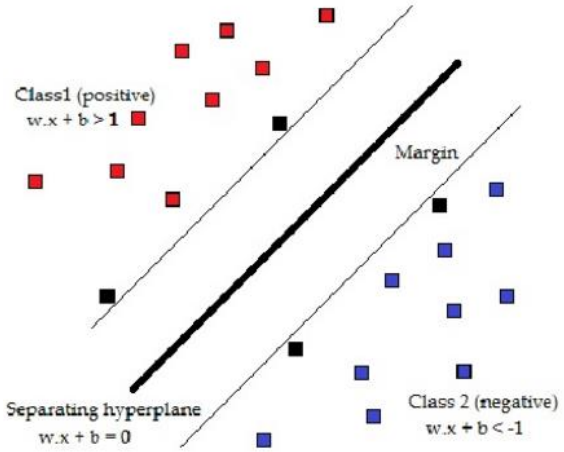
➢ **Phát biểu bài toán:**

Cho tập mẫu $\{(x_1, y_1), (x_2, y_2), \dots (x_D, y_D)\}$ trong đó $x_i \in R^D$ và $y_i \in \{-1, +1\}$. Giả sử dữ liệu là phân tách tuyến tính, tức là ta có thể phân tách dữ liệu thành hai lớp bằng cách vẽ một đường phẳng trên đồ thị của x_1, x_2 (với D = 2) hoặc một siêu phẳng trên đồ thị của x_1, x_2,… x_D (với D > 2). Mục đích của thuật toán phân lớp SVM là xây dựng siêu phẳng sao cho khoảng cách lề giữa hai lớp đạt cực đại bằng cách xác định phương trình mô tả siêu phẳng đó trên đồ thị.

➢ **Phương pháp giải bài toán:**

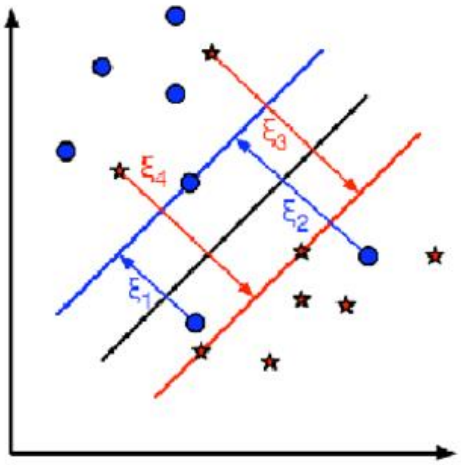
Bài toán này xảy ra ba trường hợp, mỗi trường hợp có một bài toán tối ưu và giải được bài toán này ta sẽ xây dựng được siêu phẳng cần tìm.

- Trường hợp 1: Tập dữ liệu có thể phân chia tuyến tính được mà không có nhiễu, tức là mọi điểm có nhãn “+1” nằm về phía dương trong khi mọi điểm có nhãn “-1” đều nằm về phía âm của mặt phẳng.



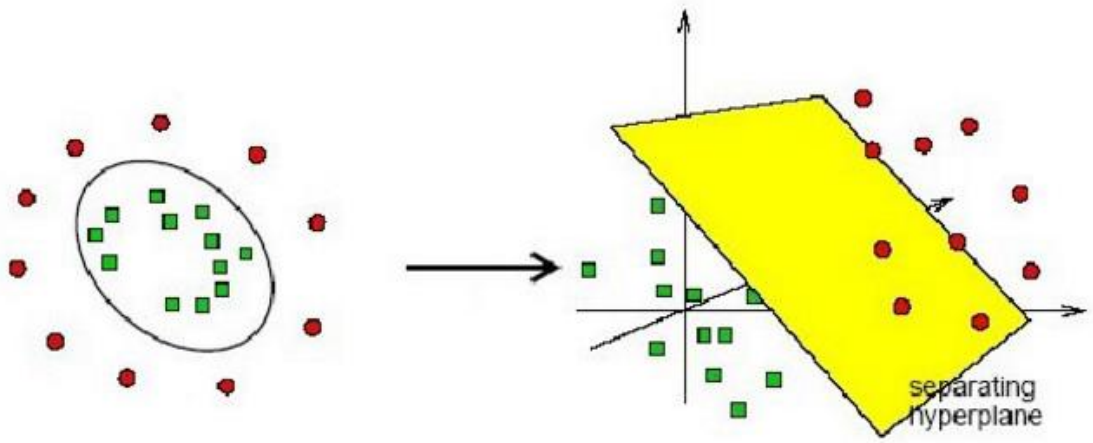
Hình 1.3: Trường hợp phân chia tuyến tính nhị phân sử dụng SVM không có nhiễu

- Trường hợp 2: Tập dữ liệu có thể phân chia tuyến tính được nhưng có nhiễu, tức là hầu hết các điểm được phân chia đúng bởi siêu phẳng nhưng có tồn tại một vài điểm nhiễu (điểm có nhãn "+1" lại nằm về phía âm của siêu phẳng, hoặc ngược lại).



Hình 1.4: Trường hợp phân chia tuyến tính nhị phân sử dụng SVM có nhiễu

- Trường hợp 3: Tập dữ liệu không thể phân chia tuyến tính được. Ta sẽ thực hiện phép ánh xạ các vector dữ liệu x vào một không gian khác có nhiều chiều hơn chiều không gian hiện tại sao cho trong không gian này, tập dữ liệu có thể phân chia tuyến tính được.



Hình 1.5: Trường hợp không thể phân chia tuyến tính nhị phân sử dụng SVM

1.4.3 Bài toán phân lớp đa lớp với SVM

Đối với bài toán phân lớp với số lớp nhiều hơn hai lớp, ta sử dụng kỹ thuật phân đa lớp dạng Multiple Binary Classification với hai chiến lược chính là One-vs-One và One-vs-Rest.

- Chiến thuật One-vs-One: Nếu k là số lớp cần phân tách, chiến lược này sẽ thực hiện k.(k-1)/2 lần phân lớp nhị phân SVM. Cụ thể: ta sẽ bắt cặp từng hai lớp một và sử dụng phương pháp chọn đa số để kết hợp các bộ phân lớp lại thành kết quả phân lớp cuối cùng.
- Chiến thuật One-vs-Rest: Chiến lược này sử dụng (k-1) bộ phân lớp đối với k lớp, tức là chuyển bài toán phân lớp k lớp thành k bài toán phân lớp nhị phân. Trong đó bộ phân lớp nhị phân thứ i được xây dựng trên lớp thứ i và các lớp còn lại.

1.4.4 Đánh giá bộ phân lớp SVM

Bộ phân lớp SVM có các ưu điểm như:

- Độ chính xác phân lớp cao, yêu cầu kích thước bộ dữ liệu huấn luyện nhỏ, dễ áp dụng cho nhiều bài toán.
- Hiệu quả với các bài toán phân lớp dữ liệu có số chiều lớn.
- Hiệu quả với các trường hợp số chiều dữ liệu lớn hơn số lượng mẫu.

Tuy nhiên, bộ phân lớp SVM còn có một số nhược điểm:

- Thời gian huấn luyện lâu, không gian bộ nhớ sử dụng lớn, được thiết kế cho phân lớp nhị phân (trong khi thực tế chủ yếu là phân loại đa lớp).
- Có thể bị overfit trên dữ liệu huấn luyện, nhạy cảm với nhiễu.

1.5 Mô hình phân lớp Maximum Entropy

Với những nhược điểm của hai bộ phân lớp trên, bộ phân lớp theo nguyên lý entropy cực đại ra đời, giải quyết tương đối tốt các bài toán phân lớp dữ liệu dạng văn

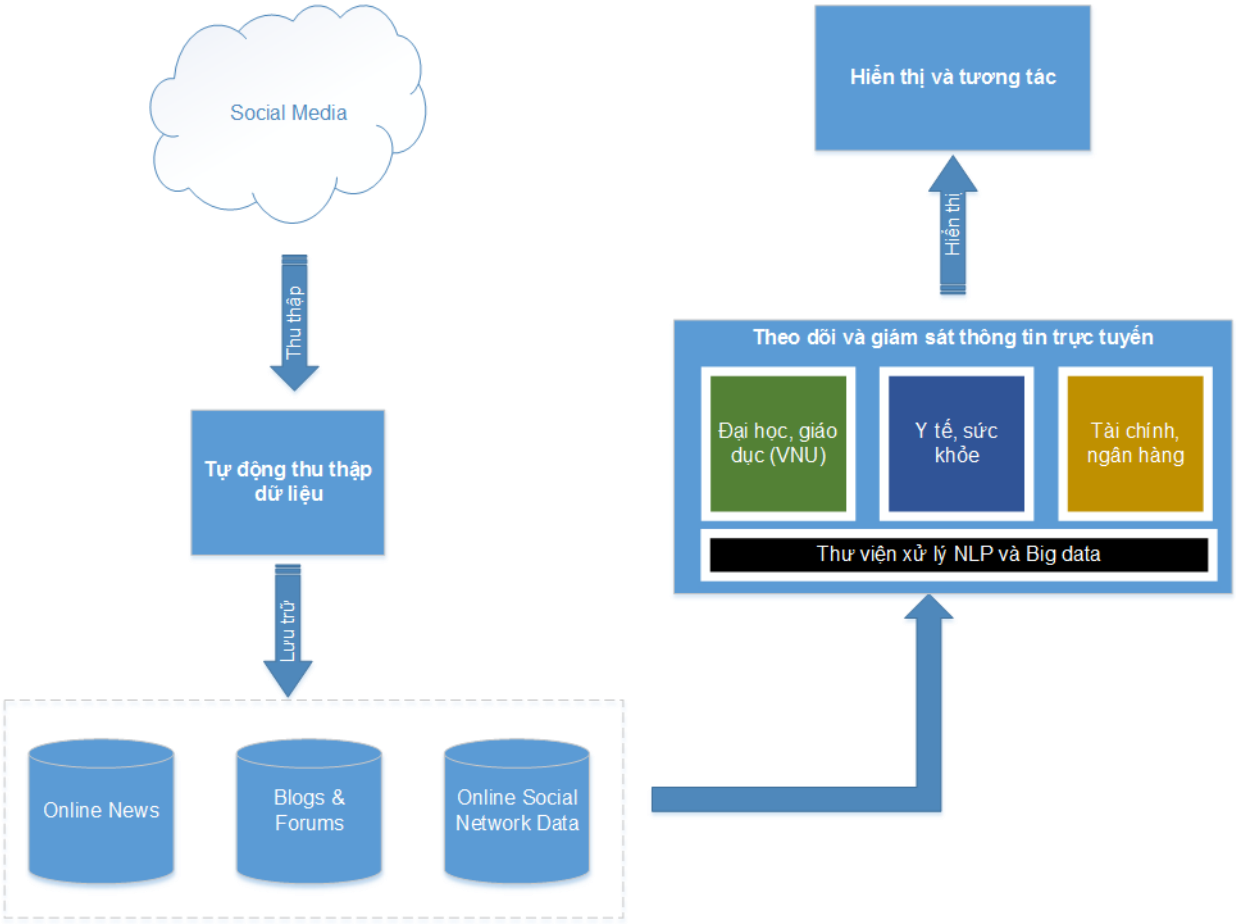
bản. Trong chương 3, chúng tôi sẽ trình bày chi tiết về bộ phân lớp này cũng như cách ứng dụng vào trong bài toán phân lớp quan điểm cho dữ liệu văn bản.

Chương 2
Tổng quan hệ thống VNU-SMM

Như chúng tôi đã đề cập ở phần Mở đầu, mục tiêu của chúng tôi là xây dựng một hệ thống tự động thu thập, phân tích và đánh giá quan điểm của người dùng chia sẻ trên các kênh truyền thông xã hội (social media) trong lĩnh vực giáo dục, ứng dụng trước mắt phục vụ cho quá trình nắm bắt thông tin và ra quyết định của Đại học Quốc gia Hà Nội. Hệ thống có tên gọi VNU-SMM.

2.1 Kiến trúc tổng thể của hệ thống

Hệ thống VNU-SMM được thiết kế với kiến trúc tổng quan như trong hình 2.1:



Hình 2.1: Thiết kế tổng quan của hệ thống VNU-SMM

Hệ thống cần thu thập, lưu trữ và xử lý, phân tích một lượng thông tin khổng lồ từ các kênh truyền thông xã hội với yêu cầu xử lý nhanh, kịp thời nên thiết kế của hệ thống cần đảm bảo được các yêu cầu này. Về công nghệ, hệ thống được tích hợp và cài đặt nhiều công nghệ hiện đại về điện toán đám mây và xử lý dữ liệu lớn. Thêm vào đó, hệ thống cũng được thiết kế theo kiến trúc mở, phục vụ việc linh động trong mở rộng

ứng dụng của hệ thống ra nhiều lĩnh vực khác ngoài giáo dục như y tế, sức khỏe hay tài chính, ngân hàng.

Từ Hình 2.1, ta có thể thấy hệ thống VNU-SMM được thiết kế với ba khối chức năng chính: khối chức năng tự động thu thập dữ liệu, khối chức năng theo dõi và giám sát thông tin trực tuyến và khối hiển thị, giao diện tương tác với người sử dụng.

2.1.1. Khối chức năng tự động thu thập dữ liệu

Khối chức năng tự động thu thập dữ liệu có các chức năng chính như: tự động thu thập dữ liệu từ các kênh truyền thông xã hội như facebook, twitter, các blog, forums. Sau đó, dữ liệu thu được sẽ được đi qua thành phần tiền xử lý dữ liệu (data preprocessing) để chuẩn hóa và làm sạch thông tin. Dữ liệu sau khi được chuẩn hóa và làm sạch sẽ được hệ thống lưu vào cơ sở dữ liệu, đồng thời tự động đánh chỉ mục phục vụ việc truy xuất dữ liệu nhanh chóng khi cần sử dụng.

Ngoài ra, khối chức năng này còn thực hiện nhiệm vụ phân tích sơ bộ dữ liệu (data shallow analysis). Dữ liệu trong và sau quá trình đánh chỉ mục sẽ được phân loại và gán nhãn (tags), đồng thời cũng được tiến hành một số bước xử lý ngôn ngữ tự nhiên mức nông như tách câu, từ, POS tagging, xác định tên riêng (NER – Named Entity Recognition),…

2.1.2. Khối chức năng lõi với chức năng theo dõi và giám sát thông tin trực tuyến

Khối chức năng tự động theo dõi và giám sát thông tin trực tuyến là khối chức năng lõi của hệ thống. Khối chức năng thực hiện các nhiệm vụ:

- Phân loại, phân lớp, thống kê và tổng hợp thông tin. Các dữ liệu sau khi được thu thập và lưu trữ trong cơ sở dữ liệu sẽ được khối chức năng này thực hiện phân loại, phân lớp, đồng thời thực hiện các thao tác tổng hợp thông tin, thống kê kết quả theo thời gian khi thu thập được thêm dữ liệu mới.
- Phân tích và so sánh thương hiệu: Sau khi phân loại và phân lớp dữ liệu thu được, khối chức năng có thể thực hiện tự động việc so sánh và phân tích giữa các thương hiệu khác nhau. Ví dụ như so sánh thương hiệu giữa các trường đại học, giữa các ngân hàng với nhau.
- Phân tích các khía cạnh: thành phần này cho phép hệ thống xác định được các khía cạnh khác nhau của một đối tượng được đề cập đến trong văn bản. Sau đó tiến hành phân tích quan điểm của người dùng cho từng khía cạnh cụ thể của đối tượng. Ví dụ phân tích về các khía cạnh như chất lượng đào tạo, cơ sở vật chất, chất lượng giáo viên,.. của một trường đại học.
- Phân tích và so sánh: khác với chức năng so sánh thương hiệu, chức năng này phân tích các so sánh trực tiếp do người dùng đưa ra trong các bình luận. Ví

dụ như người dùng có thể so sánh hai trường đại học trong câu nhận xét sau: "Trường Đại học Công nghệ có trang thiết bị giảng dạy, trình chiếu tốt hơn so với trường đại học Giao thông vận tải". Trong câu nhận xét này, hai thương hiệu trường đại học được so sánh với nhau trên khía cạnh trang thiết bị giảng dạy và trình chiếu.

- Phân tích bình luận/quan điểm: chức năng này thực hiện phân tích các phản hồi của cộng đồng mạng về một vấn đề, chủ đề hoặc sản phẩm nào đó. Ví dụ như trong một bài viết về kỳ thi đánh giá năng lực do Đại học Quốc gia tổ chức, hệ thống tự động thu thập được các bình luận và quan điểm của người dùng về kỳ thi này, trong đó có thể có các ý kiến ủng hộ, đánh giá cao, đồng thời cũng có thể có các ý kiến phê bình, phản đối hay các ý kiến trung lập. Chức năng này giúp phân loại các bình luận này, giúp cho người sử dụng hệ thống có thể có cái nhìn chi tiết hơn về quan điểm của cộng đồng mạng về chủ đề đang được quan tâm.
- Phân tích ý kiến góp ý: chức năng này cho phép hệ thống phân tích các ý kiến góp ý mang tính xây dựng, đề nghị, khuyến nghị,… từ người dùng trên các kênh truyền thông giúp người sử dụng hệ thống ý thức được và từ đó có thể đưa ra các quyết định xử lý kịp thời.
- Phân tích xu hướng: ngoài các chức năng trên, chức năng này cho phép phân tích các xu hướng thông tin, những chủ đề nổi bật trong cộng đồng mạng và sự thay đổi của chúng theo thời gian.

2.1.3. Khối hiển thị, giao diện tương tác với người dùng cuối

Khối giao diện hiển thị, tương tác có chức năng cung cấp cho người sử dụng cuối một giao diện trực quan, sinh động cho từng nội dung là kết quả của các bước phân tích nói trên. Người sử dụng có thể theo dõi thông tin cập nhật theo thời gian thực, khi có dữ liệu mới cập nhật, đồng thời có thể thực hiện các thao tác tìm kiếm, so sánh, thống kê, v.v đối với các dữ liệu đã thu thập được.

2.2 Thu thập và gán nhãn dữ liệu

Dữ liệu được thu thập tự động từ các trang mạng xã hội, blogs, forums và các trang báo chí trực tuyến. Sau khi được tiền xử lý và phân tích sơ bộ, các câu thu được sẽ được gán nhãn thủ công.

Việc gán nhãn dữ liệu là một bước rất quan trọng, ảnh hưởng lớn tới độ chính xác của bộ phân lớp. Hiện nay, các câu được gán một trong ba nhãn là positive (tích cực), negative (tiêu cực) hay other (khác).

Dữ liệu của chúng tôi thu được hệ thống gồm 9353 câu, trong đó có 2812 câu là positive, 2662 câu là negative và 3879 câu là gán nhãn other.

2.3 Phân lớp quan điểm

Thành phần phân lớp quan điểm thuộc khối chức năng lõi với khả năng tự động phân lớp quan điểm theo thời gian khi có dữ liệu mới thu thập được. Thành phần phân lớp quan điểm này được chúng tôi cài đặt theo mô hình entropy cực đại theo cách thức sau:

- Chia tập dữ liệu thu thập được và đã qua tiền xử lý, gán nhãn phân lớp thành 2 phần: dữ liệu huấn luyện và dữ liệu kiểm tra, trong đó dữ liệu huấn luyện chiếm 80% trong tổng số dữ liệu thu thập được.
- Xây dựng các hàm đặc trưng cho mô hình.
- Cài đặt giải thuật tính toán xấp xỉ tham số cho mô hình dựa vào dữ liệu tập mẫu.
- Đánh giá mô hình với bộ dữ liệu kiểm tra.
- Hiệu chỉnh lại các hàm đặc trưng sao cho kết quả đạt được là tốt nhất.

Chi tiết về cách cài đặt bộ phân lớp theo mô hình entropy cực đại sẽ được chúng tôi trình bày chi tiết trong chương 4 của luận văn.

Chương 3
Bộ phân lớp Maximum Entropy

Bài toán phân lớp nói chung và bài toán phân lớp quan điểm nói riêng bản chất là bài toán phân loại thống kê với mục tiêu tính toán xác suất một phân lớp *a* xuất hiện cùng một ngữ cảnh *b* hay ta cần tính *p(a,b)*. Trong bài toán phân lớp văn bản, ngữ cảnh thường là các từ hoặc cụm từ hoặc phức tạp hơn có thể bao gồm các nhãn về ngữ nghĩa của từ. Trong các tập dữ liệu huấn luyện, chúng ta có thể thống kê được số trường hợp *a* và *b* xuất hiện cùng nhau nhưng trong thực tế, tập dữ liệu huấn luyện không bao giờ đủ lớn để có thể tính chính xác được xác suất *p(a,b)* mà phải thông qua sử dụng các mô hình học, xác định một cách tương đối chính xác giá trị này [11]. Mô hình xác suất entropy cực đại là một kỹ thuật giúp ta xấp xỉ giá trị *p(a,b)* chỉ từ tập dữ liệu quan sát được.

Trước khi bắt đầu đi vào chương này, chúng tôi sẽ giới thiệu một số ký hiệu sau phục vụ cho việc dễ dàng đọc hiểu các nội dung trong chương:

3.1. Tổng quan về entropy cực đại

Trong mục này, chúng tôi sẽ giới thiệu về khái niệm entropy cực đại thông qua một ví dụ đơn giản. Giả sử chúng ta cần mô hình hóa lại các quyết định của một chuyên gia khi phân lớp chủ đề cho một bài báo. Mô hình *p* gán cho mỗi phân lớp *f* một giá trị xấp xỉ *p(f)* là xác suất mà chuyên gia sẽ chọn *f* là phân lớp của bài báo. Để có thể xây dựng được mô hình *p*, chúng ta trước tiên cần thu thập một lượng lớn các mẫu lựa chọn phân lớp của chuyên gia. Mục tiêu của chúng ta là (1) trích xuất các dữ liệu thực về quá trình ra quyết định từ tập mẫu thu thập được và (2) xây dựng mô hình *p* cho quá trình ra quyết định này.

Trong ví dụ này, chúng ta có các giá trị của phân lớp *f* là một trong các giá trị: {sport, music, politics, education}. Ta có ràng buộc đầu tiên của mô hình *p* như sau:

$$p(sport) + p(music) + p(politics) + p(education) = 1$$

Bây giờ ta tiếp tục xây dựng mô hình thỏa mãn ràng buộc này. Dễ dàng ta có thể thấy được có vô số mô hình *p* thỏa mãn điều kiện này. Ví dụ như ta có thể có mô hình *p(sport) = 1* (luôn phân lớp thành *sport*) hay như *p(sport)=1/2* và *p(music)=1/2*. Tuy nhiên, cả 2 mô hình này đều không hợp lý và đang đưa ra quá nhiều chi tiết hơn so với những gì chúng ta đã biết. Nếu chúng ta chỉ biết được là chuyên gia sẽ đưa ra 1 trong 3 lựa chọn và các lựa chọn là riêng biệt thì mô hình hợp lý nhất để biểu diễn tri thức này là:

$$p(sport) = p(music) = p(politics) = p(education) = \frac{1}{4}$$

Đây là mô hình đồng đều nhất dựa trên những gì ta đã biết. Giả sử chúng ta có thêm thông tin thứ 2 từ tập mẫu: chuyên gia chọn *sport* hoặc *music* trong 40% các trường hợp. Chúng ta có thể sử dụng tri thức này để cập nhật cho mô hình p đáp ứng 2 ràng buộc:

$$p(sport) + p(music) = 0.4$$
$$p(sport) + p(music) + p(politics) + p(education) = 1$$

Tương tự như trên, ta vẫn có vô số các mô hình thỏa mãn cả 2 ràng buộc này. Và tương tự như trên, phân phối hợp lý nhất khi chúng ta chỉ biết được 2 ràng buộc này là:

$$p(politics) = p(education) = 0.3$$
$$p(sport) = p(music) = 0.2$$

Giả sử chúng ta tiếp tục biết thêm được một thông tin khác từ tập mẫu là chuyên gia chọn sport và education trong 60% trường hợp. Ta tiếp tục update mô hình *p* thỏa mãn 3 ràng buộc:

$$p(sport) + p(music) = 0.4$$
$$p(sport) + p(politics) = 0.6$$
$$p(sport) + p(music) + p(politics) + p(education) = 1$$

Chúng ta tiếp tục tìm phân phối p đồng đều nhất thỏa mãn các ràng buộc này nhưng lần này giá trị của p không còn dễ thấy như trong 2 trường hợp trên. Khi thêm ràng buộc thứ 3, ta đã tăng độ phức tạp và dẫn tới 2 vấn đề cần giải quyết. Thứ nhất, định nghĩa tính đồng đều và cách để tính giá trị đồng đều này. Và thứ hai, làm thế nào để tìm ra được mô hình đồng đều nhất thỏa mãn các ràng buộc đã cho.

Mô hình entropy cực đại mà chúng tôi giới thiệu trong chương này sẽ trả lời 2 câu hỏi này. Nhưng trước tiên, chúng ta sẽ đi tìm hiểu khái niệm về entropy.

3.2. Entropy là gì?

Trong phần này, chúng tôi sẽ giới thiệu tổng quan về entropy cũng như một số tính chất cơ bản của nó.

Trong lý thuyết thông tin, chúng ta quan tâm tới việc phát triển một thước đo lượng thông tin thu được từ việc quan sát một sự kiện xảy ra với một xác suất *p*. Để đơn giản hóa vấn đề, chúng ta tạm bỏ qua các đặc trưng của sự kiện mà chỉ quan tâm

tới sự kiện xảy ra hay không. Chúng ta sẽ định nghĩa thông tin theo xác suất p là I(p) dựa trên bốn tiên đề sau [2]:

- Thông tin là một đại lượng không âm: I(p) ≥ 0
- Nếu sự kiện chắc chắn xảy ra hay p = 1, chúng ta không thu thêm được thông tin từ việc sự kiện xảy ra hay: I(1) = 0
- Nếu 2 sự kiện độc lập với các xác suất xảy ra lần lượt là p_1 và p_2 thì thông tin thu được từ việc cả 2 sự kiện xảy ra là tổng thông tin thu được từ từng sự kiện: I(p_1*p_2) = I(p_1) + I(p_2)
- Thước đo thông tin theo xác suất là một hàm liên tục (xác suất thay đổi dù rất nhỏ cũng sẽ thay đổi thông tin)

Từ các tiên đề trên ta thấy:

- I(p^m) = I(p*p*...*p) = m*I(p) với m là số nguyên
- I(p) = I(($p^{1/m}$)^m) = m*I($p^{1/m}$) ⇒ 1/m * I(p) = I($p^{1/m}$). Tổng quát lên ta có I($p^{n/m}$) = n/m * I(p)
- Thêm vào đó, theo tiên đề 4) thông tin là hàm liên tục ⇒ với a > 0 là một số thực, ta có:

$$I(p^a) = a * I(p)$$

Từ trên suy ra tính chất của thông tin:

$$I(p) = -\log(p) = \log(\frac{1}{p}) \text{ do } (0 \le p \le 1) \quad (3.1)$$

Ví dụ: tung đồng xu đồng chất với 2 mặt ngửa và sấp với xác suất ra sấp hoặc ngửa bằng nhau = 1/2. Mỗi lần tung (sự kiện xảy ra) ta thu được lượng thông tin:

$$I(\frac{1}{2}) = -\log(\frac{1}{2}) = 1 \text{ bit thông tin.}$$

Tung đồng xu n lần ta thu được n bit thông tin.

Trong ví dụ này, ta có 2 giá trị có thể có của sự kiện đó là sấp và ngửa với $p_{sấp}$ = $p_{ngửa}$ = 0,5. Tổng quát ta có một sự kiện X có n giá trị $\{a_1, a_2, ..., a_n\}$ với xác suất lần lượt là $\{p_1, p_2, ..., p_n\}$. Câu hỏi đặt ra là: lượng thông tin trung bình mà ta thu được là bao nhiêu khi ta thấy từng giá trị của một chuỗi sự kiện xảy ra?

Khi quan sát thấy giá trị a_i xuất hiện, ta thu được lượng thông tin bằng $\log(\frac{1}{p_i})$.

Trong một chuỗi sự kiện (giả sử gồm n sự kiện) thì ta sẽ quan sát thấy khoảng $n * p_i$ lần giá trị a_i. Vì vậy, trong một chuỗi n sự kiện xảy ra, ta sẽ thu được lượng thông tin:

$I = \sum_{i=1}^{n} n * p_i \log(\frac{1}{p_i})$ ⇒ lượng thông tin thu được từ mỗi giá trị quan sát được là:

$$\frac{I}{n} = \frac{1}{n}\sum_{i=1}^{n} n * p_i \log(\frac{1}{p_i}) = \sum_{i=1}^{n} p_i \log(\frac{1}{p_i}) = -\sum_{i=1}^{n} p_i \log(p_i) \quad (3.2)$$

Từ đây ta có định nghĩa về Entropy do Shannon đưa ra vào năm 1948:

Với một tập hợp các xác suất $P = \{p_1, p_2, ..., p_n\}$ ta có entropy của P được định nghĩa như sau:

$$H(P) = -\sum_{i=1}^{n} p_i \log p_i \quad (3.3)$$

Hoặc chúng ta cũng có thể định nghĩa entropy theo khái niệm kỳ vọng như sau: Entropy của một phân phối xác suất là giá trị kỳ vọng của thông tin của phân phối đó.

Ngoài công thức tính entropy cho một phân phối xác suất, ta có một số công thức khác như sau:

- **Entropy hợp của một cặp biến rời rạc (X,Y):**

$$H(X,Y) = -\sum_{x \in \chi} \sum_{y \in \gamma} p(x,y) \log p(\mathrm{x},\mathrm{y}) \quad (3.4)$$

- **Entropy có điều kiện**

Ta có định nghĩa về entropy có điều kiện của 2 biến ngẫu nhiên X, Y như sau:

$$H(X \mid Y) = -\sum_{x \in \chi} \sum_{y \in \gamma} p(x,y) \log p(\mathrm{y} \mid \mathrm{x}) \quad (3.5)$$

- **Một số tính chất của entropy**
  - $H(X) \geq 0 \forall p(x)$

  Chứng minh: $0 \leq p(x) \leq 1 \rightarrow -\log p(x) \geq 0 \ \ \forall x \in X$
  - $H(X) \leq \log(M)$ với M là tổng số giá trị ouput khác nhau.
  - $H(X) = 0 \Leftrightarrow p(\mathrm{x}) = 1$
  - $H(\mathrm{X}) \geq H(X \mid Y)$
  - Tính đối xứng: $H(x_1, x_2, x_3, ..., x_n) = H(x_2, x_1, x_3, ..., x_n)$
  - Luật xích: $H(\mathrm{X},\mathrm{Y}) = \mathrm{H}(\mathrm{X}) + \mathrm{H}(\mathrm{Y} \mid \mathrm{X}) = \mathrm{H}(\mathrm{Y}) + \mathrm{H}(\mathrm{X} \mid \mathrm{Y})$

3.3. Mô hình Maximum Entropy (ME)

3.3.1. Các ràng buộc và đặc trưng

Trong mô hình entropy cực đại, chúng ta sử dụng các tập mẫu huấn luyện (training data) để sinh ra các ràng buộc cho phân phối điều kiện. Mỗi ràng buộc thể hiện một đặc trưng của tập mẫu mà phân phối đã học cần có. Phân phối sau khi học xong phải thỏa mãn tất cả các ràng buộc sinh ra từ tập mẫu, ngoài ra không cho thêm bất kì giả thiết nào khác.

Các hàm đặc trưng $f(x,y)$ (còn gọi tắt là đặc trưng) là một hàm nhị phân với 2 tham số: $y \in$ tập các lớp cần phân loại và $x \in$ tập các ngữ cảnh:

$$f = \varepsilon \to \{0,1\}$$

Giá trị kỳ vọng của f có phân phối xác suất quan sát được $\tilde{p}(x,y)$ là:

$$E_{\tilde{p}} f_i = \sum_{a,b} \tilde{p}(x,y) f(x,y) \tag{3.6}$$

Mọi tri thức quan sát được từ tập mẫu đều có thể được biểu diễn dưới dạng giá trị kỳ vọng của một hàm đặc trưng f phù hợp.

Với k đặc trưng, các ràng buộc được biểu diễn dưới điều kiện:

$$E_p f_i = E_{\tilde{p}} f_i \tag{3.7}$$

với $0 \le i \le k$.

Việc chúng ta lựa chọn các hàm đặc trưng là tùy thuộc vào từng bài toán khác nhau và cách lựa chọn đặc trưng sẽ ảnh hưởng đến chất lượng của bộ phân lớp.

3.3.2. Nguyên lý Entropy cực đại

Giả sử ta có n đặc trưng f là những thống kê quan trọng từ tập mẫu để mô hình hóa quá trình ra quyết định phân lớp, ta sẽ muốn mô hình *p* sẽ đáp ứng được các ràng buộc thống kê này. Cụ thể, ta muốn mô hình *p* sẽ thuộc tập C là tập con của P định nghĩa như sau:

$$C \equiv \{p \in P \mid p(f_i) = \tilde{p}(f_i)\} \text{ với } i \in \{1,2,...,n\} \tag{3.8}$$

Theo nguyên lý của entropy cực đại, ta cần chọn ra mô hình *p* thuộc C có phân phối đồng đều nhất. Và câu hỏi đặt ra cần trả lời là "đồng đều" có nghĩa là gì và được tính như thế nào?

Từ những tính chất của entropy trong mục 2.1 ta có thể thấy entropy của một phân phối xác suất còn thể hiện sự đồng đều trong phân phối xác suất. Cụ thể, entropy càng cao thì phân phối càng đều. Nguyên lý Entropy cực đại cho rằng: *Với một tập các dữ liệu đã biết trước, phân phối xác suất tốt nhất trong tập các phân phối xác suất có thể để biểu diễn trạng thái hiện tại của tri thức, là phân phối xác suất có entropy cực đại và phân phối này là duy nhất.*

Ta có thể tóm tắt ý tưởng, bản chất của nguyên lý entropy cực đại như sau: Nguyên lý entropy cực đại không giả thiết bất cứ điều gì về phân phối xác suất ngoài những gì quan sát được từ tập dữ liệu, đồng thời luôn chọn phân phối xác suất đồng đều nhất phù hợp với các ràng buộc quan sát được này.

3.3.3. Dạng tham số

Bài toán đặt ra theo nguyên lý entropy cực đại có dạng: tìm *p** thuộc C sao cho entropy là lớn nhất. Bài toán có thể dễ dàng được giải quyết khi số ràng buộc là ít và đơn giản, tuy nhiên, trong thực tế số các ràng buộc tăng lên và chồng chéo nhau như trong ví dụ ở mục 2.1 thì ta cần một hướng giải quyết hiệu quả hơn.

Để giải quyết vấn đề này, chúng ta có thể áp dụng phương pháp thừa số Lagrange như sau [1]:

- Với mỗi đặc trưng f_i ta có một tham số λ_i (thừa số nhân Lagrange), hàm Lagrange được định nghĩa như sau:

$$\Lambda(p,\lambda) \equiv H(p) + \sum_i \lambda_i (p(f_i) - \tilde{p}(f_i)) \tag{3.9}$$

- Giữ λ cố định, ta tìm p_λ sao cho hàm không ràng buộc $\Lambda(p,\lambda)$ cực đại.

Ta định nghĩa hàm số $\Psi(\lambda)$ là giá trị của hàm Lagrange tại p_λ.

- Ta có công thức tính p_λ và $\Psi(\lambda)$ như sau:

$$p_\lambda(y \mid x) = \frac{1}{Z_\lambda(x)} \exp\left(\sum_i \lambda_i f_i(x,y) \right) \tag{3.10}$$

$$\Psi(\lambda) = -\sum_x \tilde{p}(x) \log Z_\lambda(x) + \sum_i \lambda_i \tilde{p}(f_i) \tag{3.11}$$

Trong đó $Z_\lambda(x)$ là hằng số chuẩn hóa thỏa mãn: $\sum_y p_\lambda(y \mid x) = 1 \ \forall x$. Ta có công thức tính $Z_\lambda(x)$:

$$Z_\lambda(x) = \sum_y \exp\left(\sum_i \lambda_i f_i(x,y) \right) \tag{3.12}$$

Theo lý thuyết thừa số Lagrange, nếu tìm được $\lambda^* = \arg\max \Psi(\lambda)$ thì p_{λ^*} là phân phối cần tìm. Hệ quả quan trọng của kết quả này là ta có thể kết luận: mọi giải thuật tìm λ^* cũng có thể được sử dụng để tìm *p** của H(*p*) với *p* thuộc C.

3.3.4. Tính toán các tham số

Đối với tất cả các vấn đề, trừ vấn đề đơn giản nhất, giá trị λ^* làm cho $\Psi(\lambda)$ cực đại không thể được tính toán bằng các phương pháp giải tích mà thay vào đó là các phương pháp số học. Có nhiều phương pháp số học được sử dụng, có thể kể đến như IIS (Improved Iterative Scaling), L-BFGS, GIS (Generalized Iterative Scaling).

Trong phần này, chúng tôi sẽ giới thiệu tổng quan về hai phương pháp phổ biến và tốt nhất hiện nay cho bộ phân lớp dựa trên mô hình entropy cực đại: IIS và L-BFGS [6] .

➢ Phương pháp Improved Iterative Scaling

Phương pháp này được hai nhà khoa học Darroch và Ratcliff giới thiệu vào năm 1972 để tính toán các xấp xỉ cực đại likelihood cho các tham số của các mô hình hàm mũ (exponential model). Thuật toán này được áp dụng với điều kiện các hàm đặc trưng $f_i(x,y)$ không âm:

$$f_i(x,y) \geq 0 \quad \forall \mathrm{x,y,i}$$

Trong bài toán phân lớp chúng ta đang giải quyết, điều kiện này hiển nhiên thỏa mãn do các hàm đặc trưng là các hàm nhị phân. Nội dung của thuật toán được trình bày như sau:

Input: Các hàm đặc trưng $f_i(x,y)$ và phân phối thực nghiệm $\tilde{p}(x,y)$

Output: Các tham số tối ưu λ_i* và mô hình tối ưu p_{λ^*}

Bước 1: Bắt đầu với λ_i = 0 với mọi i ∈{1,2,…,n}

Bước 2: Với mỗi i thực hiện:

- Gọi $\Delta\lambda_i$ là nghiệm của phương trình:

$$\sum_{x,y} \tilde{p}(x)\, p(y \mid x) f_i(x,y) \exp(\Delta\lambda_i f^{\#}(x,y)) = \tilde{p}(f_i) \qquad (3.13)$$

Trong đó: $f^{\#}(x,y) = \sum_{i=1}^{n} f_i(x,y)$

- Cập nhật lại giá trị của λ_i theo công thức: $\lambda_i = \lambda_i + \Delta\lambda_i$

Bước 3: Quay lại bước 2 nếu như tất cả các λ_i đều chưa hội tụ.

Từ các bước của giải thuật ta thấy bước 2a là bước then chốt để giải bài toán. Ta cần tìm được giá trị $\Delta\lambda_i$ là nghiệm của phương trình.

Nếu $f^{\#}(x,y)$ là hằng số (giả sử bằng M) thì ta có thể tính được giá trị : $\Delta\lambda_i$

$$\Delta\lambda_i = \frac{1}{M} \log \frac{\tilde{p}(f_i)}{p_\lambda(f_i)} \qquad (3.14)$$

Nếu $f^{\#}(x,y)$ không phải là hằng số thì giá trị của $\Delta\lambda_i$ phải được tính theo phương pháp số học. Một phương pháp đơn giản và hiệu quả là phương pháp Newton. Phương pháp này tính giá trị ∝* của phương trình g(∝*) = 0 lặp đi lặp lại bằng cách tính:

$$\alpha_{n+1} = \alpha_n - \frac{g(\alpha_n)}{g^{'}(\alpha_n)} \tag{3.15}$$

➢ Phương pháp L-BFGS (Limited-memory BFGS)

L-BFGS là một thuật toán tối ưu trong họ các phương pháp quasi-Newton cho phép xấp xỉ thuật toán BFGS gốc sử dụng bộ nhớ giới hạn của máy tính. Để hiểu rõ phương pháp này, chúng tôi sẽ giới thiệu tổng quan về phương pháp Newton và phương pháp Quasi-Newton trước khi giới thiệu về thuật toán L-BFGS

• Phương pháp Newton

Hầu hết các phương pháp tối ưu số học là các giải thuật lặp trong đó ta thử dần các giá trị của biến cần tìm, hội tụ dần về giá trị tối ưu của hàm số đã cho. Hay nói cách khác, với hàm số $x^* = \arg\max f(x)$, giả sử ta có một giá trị xấp xỉ x_n, ta mong muốn giá trị thử tiếp theo là x_{n+1} thỏa mãn: $f(x_n) < f(x_{n+1})$.

Phương pháp Newton tập trung vào xấp xỉ bậc 2 của hàm số cho các điểm xung quanh x_n. Giả sử hàm số f là khả vi hai lần (twice-differentiable), chúng ta có thể sử dụng xấp xỉ bậc 2 của hàm f cho các điểm 'gần' một điểm cố định bằng khai triển Taylor:

$$f(x+\Delta x) \approx f(x) + \Delta x^T \nabla f(x) + \frac{1}{2}\Delta x^T (\nabla^2 f(x))\Delta x \tag{3.16}$$

Trong đó $\nabla f(x)$ và $\nabla^2 f(x)$ lần lượt là gradient và ma trận Hessian của hàm số f tại điểm x. Xấp xỉ này đúng với giá trị Δx tiến dần tới 0. Không mất tính tổng quát, ta có thể viết: $x_{n+1} = x_n + \Delta x$ và viết lại phương trình trên như sau:

$$h_n(\Delta x) = f(x_n) + \Delta x^T g_n + \frac{1}{2}\Delta x^T H_n \Delta x \tag{3.17}$$

Trong đó g_n và H_n lần lượt là gradient và ma trận Hessian của hàm số f tại x_n.

Chúng ta cần chọn giá trị Δx để cực đại giá trị xấp xỉ địa phương của hàm f tại x_n. Lấy đạo hàm riêng với Δx ta có:

$$\frac{\partial h_n(\Delta x)}{\Delta x} = g_n + H_n \Delta x \tag{3.18}$$

Để tìm giá trị Δx sao cho hàm số đạt cực đại địa phương ta chỉ cần giải phương trình $\frac{\partial h_n(\Delta x)}{\Delta x} = 0$ hay ta có:

$$\Delta x = -H_n^{-1} g_n \tag{3.19}$$

Trong thực tế, chúng ta thường lấy giá trị $x_{n+1} = x_n + \alpha\Delta x = x_n - \alpha H_n^{-1} g_n$ với α là hằng số sao cho $f(x_n)$ vừa đủ nhỏ hơn $f(x_{n+1})$.

Từ lý luận trên, ta có giải thuật lặp NewtonRapshon như sau:

$$\begin{aligned}
&\textbf{NewtonRaphson}(f, x_0):\\
&\quad \text{For } n = 0, 1, \ldots \text{ (until converged)}:\\
&\quad\quad \text{Compute } \mathbf{g}_n \text{ and } \mathbf{H}_n^{-1} \text{ for } x_n\\
&\quad\quad d = \mathbf{H}_n^{-1}\mathbf{g}_n\\
&\quad\quad \alpha = \min_{\alpha \geq 0} f(x_n - \alpha d)\\
&\quad\quad x_{n+1} \leftarrow x_n - \alpha d
\end{aligned}$$

Hình 3.1: Giải thuật lặp NewtonRapshon

Giải thuật trên có thể được chứng minh luôn hội tụ tới điểm tối ưu cho hàm f cực đại nếu f là một hàm số lõm hay hội tụ tới f cực tiểu nếu f là hàm lồi với lựa chọn x_0 bất kỳ.

Trong thực tế với các bài toán học máy như chúng ta đang quan tâm, *f* thường là một hàm số nhiều chiều với số chiều tương ứng với số tham số của mô hình học. Số tham số này thường rất lớn, có thể lên tới hàng trăm triệu hoặc thậm chí hàng tỉ, điều này khiến cho việc thực hiện tính toán theo phương pháp Newton là không thể do không thể tính được ma trận Hessian hay nghịch đảo của nó. Chính vì vậy, trong thực tế, giải thuật NewtonRapshon rất ít khi được sử dụng với các bài toán lớn. Tuy nhiên, thuật toán trên vẫn đúng với ma trận Hessian xấp xỉ đủ tốt mà không cần chính xác tuyệt đối. Phương pháp được sử dụng để xấp xỉ ma trận Hessian này là Quasi-Newton.

- Quasi-Newton

Phương pháp Quasi-Newton sử dụng một hàm QuasiUpdate để sinh ra ma trận Hessian nghịch đảo tại x_{n+1} dựa trên ma trận Hessian nghịch đảo tại x_n:

$$
\begin{aligned}
&\textbf{QuasiNewton}(f, x_0, \mathbf{H}_0^{-1}, \text{QuasiUpdate}) : \\
&\quad \text{For } n = 0, 1, \ldots \text{ (until converged)} : \\
&\qquad // \text{ Compute search direction and step-size} \\
&\qquad d = \mathbf{H}_n^{-1} \mathbf{g}_n \\
&\qquad \alpha \leftarrow \min_{\alpha \geq 0} f(x_n - \alpha d) \\
&\qquad x_{n+1} \leftarrow x_n - \alpha d \\
&\qquad // \text{ Store the input and gradient deltas} \\
&\qquad \mathbf{g}_{n+1} \leftarrow \nabla f(x_{n+1}) \\
&\qquad s_{n+1} \leftarrow x_{n+1} - x_n \\
&\qquad y_{n+1} \leftarrow \mathbf{g}_{n+1} - \mathbf{g}_n \\
&\qquad // \text{ Update inverse hessian} \\
&\qquad \mathbf{H}_{n+1}^{-1} \leftarrow \text{QuasiUpdate}(\mathbf{H}_n^{-1}, s_{n+1}, y_{n+1})
\end{aligned}
$$

Hình 3.2: QuasiNewton Update

Ở đây, chúng ta giả sử rằng phương thức QuasiUpdate chỉ cần ma trận nghịch đảo tại điểm liền trước đó, độ lệch giữa 2 điểm và độ lệch gradient của chúng.

Bốn nhà nghiên cứu Broyden, Fletcher, Goldfarb và Shanno đã tìm ra phương thức tính xấp xỉ ma trận Hessian nghịch đảo H_n^{-1} mà ta gọi là phương thức BFGS Update:

$$
\begin{aligned}
&\textbf{BFGSMultiply}(\mathbf{H}_0^{-1}, \{s_k\}, \{y_k\}, d) : \\
&\quad r \leftarrow d \\
&\quad // \text{ Compute right product} \\
&\quad \text{for } i = n, \ldots, 1 : \\
&\qquad \alpha_i \leftarrow \rho_i s_i^T r \\
&\qquad r \leftarrow r - \alpha_i y_i \\
&\quad // \text{ Compute center} \\
&\quad r \leftarrow \mathbf{H}_0^{-1} r \\
&\quad // \text{ Compute left product} \\
&\quad \text{for } i = 1, \ldots, n : \\
&\qquad \beta \leftarrow \rho_i y_i^T r \\
&\qquad r \leftarrow r + (\alpha_{n-i+1} - \beta) s_i \\
&\quad \text{return } r
\end{aligned}
$$

Hình 3.3: BFGS Update

Ta chỉ cần sử dụng phương thức này ứng dụng vào trong phương thức QuasiNewton ở trên để xấp xỉ tham số.

Xấp xỉ BFGS Quasi-Newton có ưu điểm là không cần chúng ta phải tính toán ra ma trận Hessian của hàm số *f* mà thay vào đó, ta có thể liên tục cập nhật các giá trị xấp xỉ của nó. Tuy nhiên, chúng ta vẫn cần phải lưu lại lịch sử

của các vector s_n và y_n trong mỗi vòng lặp. Nếu vấn đề cốt lõi của phương pháp NewtonRapshon là bộ nhớ cần thiết để tính toán ma trận nghịch đảo Hessian là quá lớn thì phương pháp BFGS Quasi-Newton chưa giải quyết được vấn đề này do bộ nhớ liên tục tăng không có giới hạn. Chính vì lẽ đó, phương pháp L-BFGS ra đời với ý tưởng chỉ sử dụng m giá trị s_k và y_k gần nhất để tính toán hàm update BFGS thay vì toàn bộ số lượng vector. Việc này giúp cho bộ nhớ luôn là hữu hạn.

Chương 4
Kết quả thử nghiệm và đánh giá

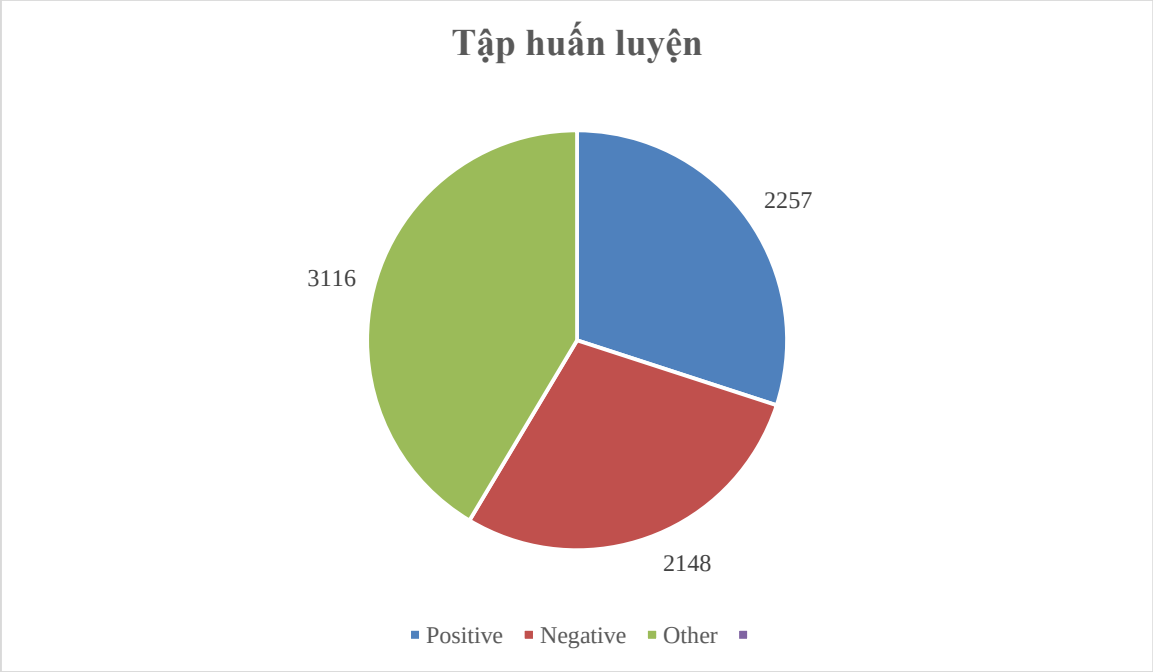
4.1. Tiến hành thử nghiệm

➢ **Bước 1:** Tự động thu thập dữ liệu từ các trang mạng trực tuyến: baomoi.com, vnexpress.net và dantri.com.vn.

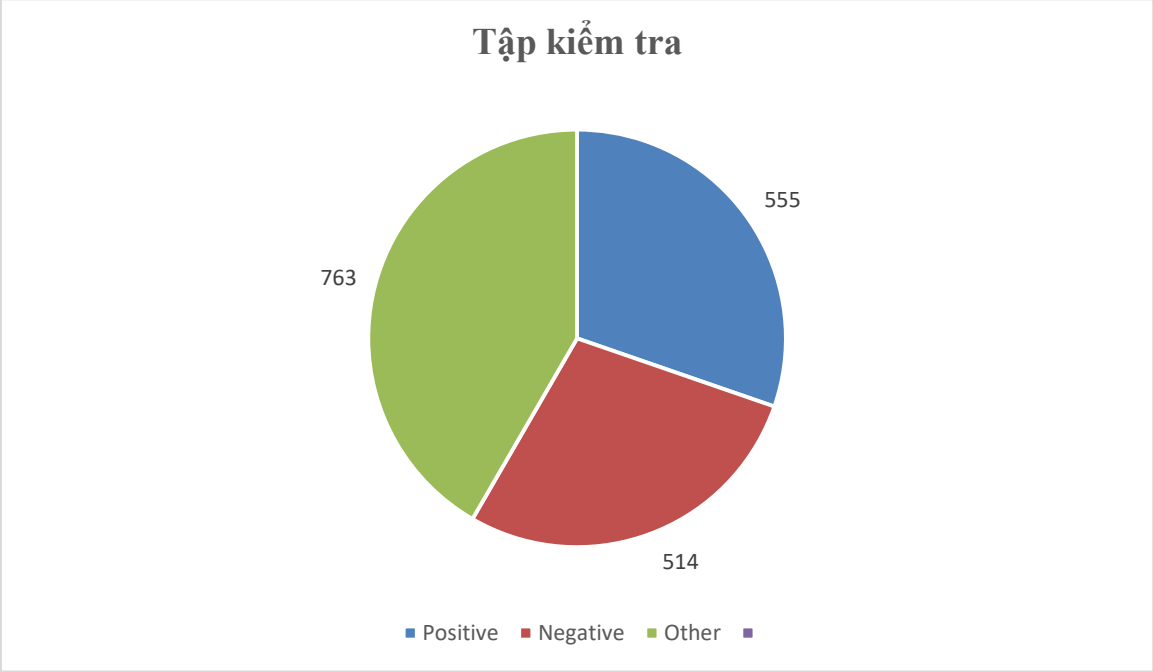
➢ **Bước 2:** Tiền xử lý dữ liệu thu thập được: làm sạch và chuẩn hóa dữ liệu, gán nhãn loại từ cho từng câu bình luận.

➢ **Bước 3:** Nhận dạng thủ công từng câu trong bộ dữ liệu mẫu và phân vào các lớp *positive* (tích cực), *negative* (tiêu cực) và *other* (khác)

➢ **Bước 4:** Tách 1832 câu trong bộ dữ liệu đã gán nhãn thành bộ test và 7521 câu còn lại là bộ huấn luyện. Thành phần các nhãn của các tập dữ liệu cụ thể như sau:



Hình 4.1: Thành phần các nhãn trong tập huấn luyện



Hình 4.2: Thành phần các nhãn trong tập kiểm tra

➢ **Bước 5:** Chạy bộ phân lớp và so sánh kết quả phân lớp tự động so với kết quả phân lớp thủ công.

4.2. Tiền xử lý dữ liệu

Dữ liệu sau khi được crawl tự động về sẽ được đưa qua bộ tiền xử lý dữ liệu trước khi đưa vào nhận dạng thủ công. Bộ tiền xử lý là JvnTextPro do các tác giả của trường Đại học Công nghệ phát triển.

Ta có một ví dụ sau:

Câu gốc thu được từ các kênh truyền thông như sau:

"*Những ảnh hưởng này xem chừng như không rõ ràng lắm, nhất là trong một thí nghiệm với vài chiếc bánh quy*"

Sau khi được tiền xử lý, ta có một câu như sau, mỗi câu là một quan sát về tập dữ liệu (observation):

"*những/L ảnh_hưởng/N này/P xem/V chừng_như/C không/R rõ_ràng/A lắm/R nhất_là/X trong/E một/M thí_nghiệm/N với/E vài/L chiếc/Nc bánh_quy/N*"

Trong ví dụ này, các dấu câu đã được loại bỏ, các từ được tokenize và gán nhãn loại từ. Bảng loại từ chi tiết trong Bảng 4.1 [8].

Bảng 4.1: Bảng nhãn từ và ý nghĩa

1	N: danh từ	10	R: phụ từ
2	Np: danh từ riêng	11	E: giới từ
3	Nc: danh từ chỉ loại	12	T: trợ từ, tiểu từ
4	Nu: danh từ đơn vị	13	B: từ mượn
5	V: động từ	14	Y: từ viết tắt

6	A: tính từ	15	X: các từ không phân loại được
7	P: đại từ	16	Mrk: dấu câu
8	L: định từ	17	C: liên từ
9	M: số từ	18	I: thán từ

4.3. Xây dựng mô hình

4.3.1. Lựa chọn đặc trưng

Như ta đã biết từ nội dung chương 2, các hàm đặc trưng f gồm hai tham số: ngữ cảnh và nhãn phân lớp.

Các hàm đặc trưng được xác định theo quy tắc sau:

- **Bước 1**: Tìm tất cả unigram, bigram của từng câu hay từng quan sát (observation).
- **Bước 2**: Sắp xếp danh sách các unigram và bigram thu được theo thứ tự giảm dần của loại từ (ưu tiên các tính từ, rồi đến danh từ, rồi đến động từ, rồi đến các loại từ khác).
- **Bước 3**: Lấy top 50 của danh sách sau khi sắp xếp làm đặc trưng cho câu hay quan sát đó.

4.3.2. Cài đặt thuật toán học

Chúng tôi cài đặt bộ phân lớp sử dụng hệ điều hành windows 10 và ngôn ngữ lập trình Java với công cụ lập trình Eclipse.

Hệ thống cài đặt thuật toán học ME sử dụng phương pháp L-BFGS để xấp xỉ tham số cho mô hình.

4.4. Kết quả thử nghiệm

4.4.1. Các chỉ số đo kiểm chất lượng bộ phân lớp

Hệ thống được đánh giá dựa trên bộ ba tiêu chí đánh giá sau:

- Độ chính xác (precision)

Độ chính xác của bộ phân lớp được định nghĩa như sau:

$$\text{Độ } chính\ xác = \frac{Số\ thực\ thể\ phân\ loại\ đúng}{Tổng\ số\ thực\ thể\ \text{đã}\ phân\ loại} \quad (4.1)$$

- Độ bao phủ (recall)

Độ bao phủ của bộ phân lớp được định nghĩa như sau:

$$\text{Độ } bao\ phủ = \frac{Số\ thực\ thể\ phân\ loại\ đúng}{Tổng\ số\ thực\ thể\ đúng\ trong\ thực\ tế} \quad (4.2)$$

- F_1

Độ đo F_1 của bộ phân lớp được định nghĩa như sau:

$$F_1 = 2 . \frac{\text{Độ } chính\ xác . \text{ Độ } bao\ phủ}{\text{Độ } chính\ xác + \text{Độ } bao\ phủ} \quad (4.3)$$

Độ đo F1 là giá trị trung hòa giữa hai giá trị độ chính xác và độ bao phủ. Chúng ta cần tính F1 bởi lẽ nếu chỉ căn cứ vào giá trị của độ chính xác và độ bao phủ, ta không thể so sánh và đánh giá các bộ phân lớp với nhau trong trường hợp bộ phân lớp này có độ chính xác cao, độ bao phủ thấp trong khi bộ phân lớp còn lại có độ chính xác thấp nhưng độ bao phủ cao.

Ta có ví dụ về ba giá trị đo này: Bộ phân lớp nhận dạng được 10 câu là thuộc phân lớp *tích cực* trong một bộ test gồm 15 câu thuộc phân lớp *tích cực* và còn lại là các câu thuộc phân lớp khác. Nếu trong 10 câu bộ phân lớp đã nhận dạng là thuộc phân lớp *tích cực* có 8 câu đúng và 2 câu nhận dạng sai thì ta có:

Độ chính xác = 8/10 = 80%

Độ bao phủ = 8/15 = 53,33%

F_1 = 2 . (0,8 . 0,53)/(0,8+0,53) = 63,76%

4.4.2. Kết quả thực nghiệm bài toán phân lớp mức độ câu

Kết quả phân loại với tập kiểm tra được thể hiện trong Bảng 4.2:

Bảng 4.2: Kết quả thực nghiệm bài toán phân lớp mức độ câu sử dụng ME

	Số thực thể	**Số thực thể nhận dạng được**	**Số thực thể nhận dạng đúng**	**Độ chính xác (%)**	**Độ bao phủ (%)**	**F_1 (%)**
	(1)	**(2)**	**(3)**	**(4)=(3)/(2)**	**(5)=(3)/(1)**	**(6)=2.(4)x(5)/((4)+(5))**
Positive	555	543	325	59.85	58.56	59.20
Negative	514	530	309	58.30	60.12	59.20
Other	763	759	460	60.61	60.29	60.45
All	1832	1832	1094	59.72	59.72	59.72

Từ bảng kết quả trên chúng ta có thể thấy, kết quả của bộ phân lớp tính theo tiêu chí độ chính xác của các nhãn positive, negative, other lần lượt là 59.85%, 58.30% và 60.61%. Các giá trị này xấp xỉ với kết quả tính theo độ bao phủ, lần lượt là 58.56%, 60.12% và 60.29%. Điều này cho thấy, bộ phân lớp tương đối ổn định khi đánh giá theo hai tiêu chí trên, kết quả là giá trị F1 theo từng nhãn cũng xấp xỉ nhau. Kết quả tính theo tiêu chí F1 đạt 59.72% nếu tính theo tổng toàn bộ nhãn của chương trình.

4.5. So sánh với bộ phân lớp Naïve Bayes

Để so sánh, chúng tôi cũng đã cài đặt bộ phân lớp Naïve Bayes và đánh giá trên cùng tập dữ liệu huấn luyện và kiểm tra như trên. Chúng tôi đã sử dụng thư viện mã nguồn mở để cài đặt và kiểm tra bộ phân lớp Naïve Bayes[1].

Kết quả cụ thể như trong Bảng 4.3.

Bảng 4.3: Kết quả thực nghiệm bài toán với bộ phân lớp Naïve Bayes

	Số thực thể	**Số thực thể nhận dạng được**	**Số thực thể nhận dạng đúng**	**Độ chính xác (%)**	**Độ bao phủ (%)**	**F_1 (%)**
	(1)	**(2)**	**(3)**	**(4)=(3)/(2)**	**(5)=(3)/(1)**	**(6)=2.(4)x(5)/((4)+(5))**
Positive	555	348	214	61.49	38.56	61.49
Negative	514	463	262	56.59	50.97	56.59
Other	763	1021	543	53.18	71.17	53.18
All	1832	1832	1019	55.62	55.62	55.62

Từ bảng kết quả trên, chúng ta có độ chính xác của ba nhãn positive, negative và other lần lượt là 61.49%, 56.59% và 53.18%. So sánh với độ bao phủ, ta thấy có sự chênh lệch lớn (38.56%, 50,97% và 55.62%) và đồng thời kết quả đo theo tiêu chí F1 đạt 55.62%, thấp hơn so với bộ phân lớp Maximum entropy. Sự không ổn định trong phân loại của bộ phân lớp Naïve Bayes có thể dẫn đến hiệu quả phân lớp rất khác nhau đối với các bộ dữ liệu khác nhau.

4.6. Đánh giá kết quả

Mặc dù bộ phân lớp Maximum entropy cho kết quả cao hơn so với bộ phân lớp sử Naïve Bayes, kết quả đạt được chưa cao (~60%). Kết quả này có thể do một số nguyên nhân sau:

+ Tập dữ liệu sử dụng để huấn luyện và kiểm tra gán nhãn còn chưa chính xác: bộ dữ liệu này sau khi được crawl về và chạy qua bộ tiền xử lý (lọc bỏ stopword, dấu câu, chữ số; đưa về dạng chữ viết thường (lowercase); phân tách từ và thực hiện pos tagging) đã được phân loại và gán nhãn bằng tay theo phương pháp crowdsourcing do khối lượng câu cần phân loại lớn. Điều này dẫn đến những bất thường và khó kiểm soát trong chất lượng nguồn dữ liệu.

+ Các đặc trưng lựa chọn chưa thực sự hiệu quả: đối với các thuật toán học máy có giám sát, việc chọn lựa được các đặc trưng hiệu quả là điểm mấu chốt quyết định đến chất lượng của cả bộ phân lớp. Trong hệ thống, chúng tôi đã sử dụng các đặc trưng phổ biến cho các bộ phân lớp chủ đề truyền thống (unigram và bigram), Part-of-

[1] https://github.com/datumbox/NaiveBayesClassifier

speech (POS) của từng từ, đồng thời kết hợp với sử dụng các đặc trưng riêng của bài toán phân lớp quan điểm như sử dụng từ điển các từ và cụm từ mang quan điểm (sentiment words and phrases) để tăng độ chính xác cho bộ phân lớp. Tuy nhiên, các đặc trưng được lựa chọn vẫn còn mang tính kinh nghiệm và đánh giá qua thực tế nên kết quả chưa được cao.

Chương 5
Tổng kết và hướng phát triển tiếp theo

Luận văn đã nghiên cứu và tìm hiểu về bài toán phân lớp quan điểm với dữ liệu là các comment, phản hồi, các góp ý từ các kênh truyền thông xã hội phổ biến, đánh giá thuật toán học maximum entropy với dữ liệu thực tế trong chủ đề giáo dục. Các kết quả chính mà luận văn đạt được như sau:

- Tìm hiểu, giới thiệu và đánh giá sơ bộ một số thuật toán học có giám sát ứng dụng trong xây dựng bộ phân lớp văn bản nói chung và phân lớp quan điểm người dùng nói riêng: thuật toán Naïve Bayes, SVM và Maximum Entropy.
- Giới thiệu và đi sâu vào thuật toán Maximum Entropy và cách ứng dụng trong hệ thống phân lớp quan điểm người dùng.
- Thử nghiệm với dữ liệu thật thu được từ các kênh truyền thông xã hội.

Tuy đã cố gắng nâng cao chất lượng bộ phân lớp, nhưng kết quả thử nghiệm với mức câu còn chưa cao (~60%) do một số nguyên nhân cả về khách quan và chủ quan, trong đó nguyên nhân chủ yếu do chất lượng của bộ dữ liệu huấn luyện và kiểm tra còn thấp, chưa đồng bộ, các đặc trưng được lựa chọn chưa hiệu quả. Trong tương lai, để cải tiến hiệu năng của bộ phân lớp, chúng tôi có thể giảm số lượng các câu trong tập huấn luyện để có thể tập trung nâng cao chất lượng gán nhãn của tập này. Bên cạnh đó, để nâng cao chất lượng của các đặc trưng, chúng tôi đề xuất sử dụng thêm các kiến thức chuyên gia về ngôn ngữ và hiểu biết về các lĩnh vực cụ thể để có thể tránh được các trường hợp phân lớp sai cơ bản nếu chỉ dựa vào việc đếm các từ trong câu. Ví dụ như chúng tôi có thể phân biệt các câu điều kiện để xử lý riêng, các câu ghép có sự so sánh, thay đổi về quan điểm để xử lý riêng, v.v. Ngoài ra, như đã trình bày trong chương 1, chúng tôi cũng cân nhắc một hướng nghiên cứu khả thi và rất có tiềm năng để tăng độ chính xác của bộ phân lớp là nghiên cứu và cài đặt phương pháp học máy deep learning cho bộ phân lớp.

Chương 6
Tài liệu tham khảo

1. Berger, A.L., Pietra, V.J.D. and Pietra, S.A.D. (1996), "A Maximum Entropy Approach to Natural Language Processing.", *Computational linguistics, 22*(1), pp. 39-71.

2. Carter, T. (2014), "An Introduction to Information Theory and Entropy.", *Complex systems summer school, Santa Fe*.

3. Devi, G.D. and Rasheed, A.A. (2015), "A Survey on Sentiment Analysis and Opinion Mining.", *International journal for research in emerging science and technology 2*(8), pp. 26-31.

4. Hu, M. and Liu, B. (2004), "Mining and Summarizing Customer Reviews.", *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining,* pp. 168-177.

5. Liu, B. (2012), *Sentiment Analysis and Opinion Mining,* Morgan & Claypool.

6. Malouf, R. (2002), "A Comparison of Algorithms for Maximum Entropy Parameter Estimation." *Proceedings of the 6th conference on Natural language learning 20,* pp. 1-7.

7. Medhat, W., Hassan, A. and Korashy, H. (2014), "Sentiment Analysis Algorithms and Applications: A Survey.", *Ain Shams Engineering Journal 5*(4), pp. 1093-1113.

8. Nguyen Cam Tu, Phan Xuan Hieu and Nguyen Thu Trang (2010), "Manual for JvnTextPro".

9. Pang, B., Lee, L. and Vaithyanathan, S. (2002), "Thumbs up?: Sentiment Classification Using Machine Learning Techniques.", *Proceedings of the ACL-02 conference on Empirical methods in natural language processing 10,* pp. 79-86.

10. Rădulescu, C., Dinsoreanu, M. and Potolea, R. (2014), "Identification of Spam Comments Using Natural Language Processing Techniques", *Intelligent Computer Communication and Processing (ICCP),* pp. 29-35.

11. Ratnaparkhi, A. (1997), "A Simple Introduction to Maximum Entropy Models for Natural Language Processing", *IRCS Technical Reports Series*.

12. Severyn, A. and Moschitti, A. (2015), "Twitter Sentiment Analysis with Deep Convolutional Neural Networks.", *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 959-962.

13. Tang, D., Qin, B. and Liu, T. (2015), “Deep Learning for Sentiment Analysis: Successful Approaches and Future Challenges.”, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 5*(6), pp. 292-303.

14. Tsytsarau, M. and Palpanas, T. (2012), “Survey on Mining Subjective Data on the Web.”, *Data Mining and Knowledge Discovery*, *24*(3), pp. 478-514.

15. Xia, R., Xu, F., Zong, C., Li, Q., Qi, Y. and Li, T. (2015), “Dual Sentiment Analysis: Considering Two Sides of One Review.”, *IEEE transactions on knowledge and data engineering, 27*(8), pp.2120-2133.

16. Zhang, X., Li, S., Zhou, G. and Zhao, H. (2011), “Polarity Shifting: Corpus Construction and Analysis.”, *Asian Language Processing (IALP)*, pp. 272-275.