HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN I

ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

Đề tài: "Phân loại quan điểm người dùng về sản phẩm công nghệ sử dụng phương pháp Support Vector Machine (SVM)"

Giảng viên hướng dẫn : Th.s. NGUYỄN THANH THỦY
Sinh viên thực hiện : LÊ TÔN ANH THƯ
Lớp : D10CNPM3
Khoá : 2010-2015
Hệ : Chính quy

Hà Nội, tháng 12/2014

LỜI CẢM ƠN

Để hoàn thành khóa luận này, em xin tỏ lòng biết ơn sâu sắc đến Th.s Nguyễn Thanh Thủy đã tận tình hướng dẫn em trong suốt quá trình viết khóa luận tốt nghiệp.

Em cũng xin chân thành cảm ơn các thầy, cô trong khoa Công Nghệ Thông Tin I – Học Viện Công Nghệ Bưu Chính Viễn Thông đã tận tình truyền đạt kiến thức cho em trong suốt quá trình học tập và làm Đồ án vừa qua. Với vốn kiến thức được tiếp thu trong quá trình học không chỉ là nền tảng cho quá trình nghiên cứu khóa luận mà còn là hành trang quí báu để sinh viên chúng em bước vào đời một cách vững chắc và tự tin.

Em cũng thầm biết ơn sự ủng hộ của gia đình, bạn bè – những người thân yêu luôn là chỗ dựa vững chắc, nguồn động viên to lớn, giúp đỡ em vượt qua những khó khăn trong suốt quá trình học tập và làm Đồ án.

Mặc dù đã cố gắng hoàn thiện khóa luận với tất cả sự nỗ lực của bản thân, nhưng chắc chắn em không thể tránh khỏi những thiếu sót, kính mong thầy, cô tận tình chỉ bảo.

Cuối cùng, em xin gửi lời chúc tốt đẹp nhất đến đến những người bạn đã đồng hành và các thầy cô giáo giúp đỡ em trong hơn bốn năm qua. Chúc cho mọi người luôn vui vẻ và thành công trong cuộc sống.

Em xin chân thành cảm ơn, Hà Nội, tháng 11 năm 2014

Sinh viên thực hiện

Lê Tôn Anh Thư

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM
(Của giảng viên hướng dẫn)

……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..

Điểm: ……(bằng chữ: …..…… ….)

……, ngày tháng năm 20

CÁN BỘ- GIẢNG VIÊN HƯỚNG DẪN

(ký, họ tên)

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM
(Của giảng viên phản biện)

……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..
……..

Điểm: ……..……(bằng chữ: …..…… ….)

MỤC LỤC

LỜI CẢM ƠN ..... **i**
DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH ..... **vi**
KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT ..... **vii**
MỞ ĐẦU **1**
CHƯƠNG 1: GIỚI THIỆU VỀ BÀI TOÁN PHÂN LOẠI QUAN ĐIỂM **3**
1.1. Một số khái niệm về phân lớp **3**
1.2. Một số kĩ thuật được sử dụng trong bài toán phân lớp dữ liệu **3**
1.3. Nhóm các giải thuật học máy **5**
1.3.1. Học có giám sát (Supervised Learning) **5**
1.3.2. Học không giám sát (Unsupervised Learning) **5**
1.3.3. Học bán giám sát (Semi - Supervised Learning) **6**
1.3.4. Học tăng cường **6**
1.4. Giới thiệu bài toán phân loại quan điểm người dùng về sản phẩm công nghệ **6**
1.4.1. Nhu cầu hiện nay **6**
1.4.2. Phát biểu bài toán **7**
1.4.3. Mô hình tổng quát **7**
CHƯƠNG 2: MỘT SỐ PHƯƠNG PHÁP SỬ DỤNG TRONG PHÂN LOẠI QUAN ĐIỂM VÀ CÁC NGHIÊN CỨU LIÊN QUAN **10**
2.1. Một số phương pháp sử dụng trong phân loại quan điểm **10**
2.1.1. Phương pháp Naïve (NB) **10**
2.1.1. Phương pháp K - Nearest Neighbor (kNN) **12**
2.1.2. Phương pháp sử dụng cây quyết định (Decision tree) **14**
2.1.3. Phương pháp SVM **17**
2.2. Các nghiên cứu liên quan **23**
2.3. Kết luận **23**
CHƯƠNG 3: XÂY DỰNG MÔ HÌNH PHÂN LOẠI QUAN ĐIỂM NGƯỜI DÙNG VỀ SẢN PHẨM CÔNG NGHỆ **25**
3.1. Yêu cầu dữ liệu **25**
3.2. Thu thập và tiền xử lý dữ liệu **25**
3.2.1. Google Custom Search **25**
3.2.2. Thu thập dữ liệu tự động **26**
3.2.3. Tiền xử lý dữ liệu **27**
3.3. Xây dựng bộ từ đặc trưng **29**
3.3.1. Giới thiệu VietSentiWordNet **30**

3.3.2. Bộ từ đặc trưng 30
3.4. Phân loại quan điểm 30
CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ 34
4.1. Phương pháp đánh giá 34
4.1.1. Phương pháp kiểm tra chéo Cross Validation 34
4.1.2. Bộ thư viện hỗ trợ LibSVM 34
4.2. Dữ liệu đầu vào 35
4.3. Quá trình đánh giá 35
4.4. Kết quả 38
KẾT LUẬN 39
DANH MỤC TÀI LIỆU THAM KHẢO 41

DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH

Bảng 2.1 Biểu diễn văn bản vector nhị phân 16
Bảng 3.1: Kết quả sau các bước xử lý dữ liệu 29
Bảng 4.1: Phân chia dữ liệu 35
Bảng 4.2: Kết quả đánh giá của tập dữ liệu với bộ đặc trưng 400 từ 37
Bảng 4.3: Kết quả đánh giá của tập dữ liệu với bộ đặc trưng 300 từ 37
Bảng 4.4: Kết quả đánh giá của tập dữ liệu với bộ đặc trưng 200 từ 38
Bảng 4.5: Kết quả đánh giá của hệ thống với mỗi bộ từ đặc trưng khác nhau.....38

Hình 1.1: Ví dụ xây dựng mô hình 4
Hình 1.2: Sử dụng mô hình.....4
Hình 1.3: Mô hình chi tiết giai đoạn huấn luyện 8
Hình 1.4: Mô hình chi tiết giai đoạn phân lớp.....9
Hình 2.1. Xây dựng cây quyết định cho tập mẫu dùng để huấn luyện.....16
Hình 2.2 : Quá trình tìm kiếm lời giải trên cây quyết đinh 17
Hình 2.3: Một đường thẳng tuyến tính phân chia hai lớp điểm.....18
Hình 2.4: Độ rộng biên lớn nhất được tính toán bởi một SVM tuyến tính. 19
Hình 2.5: Ảnh hưởng của hằng số biên mềm C trên ranh giới quyết định.....21
Hình 2.6: Mức độ tác động của kernel đa thức.....22
Hình 3.1: Kết quả thu được sau bước thu thập dữ liệu tự động 27
Hình 3.2: Định dạng chuẩn cấu trúc vector của SVM^{light} 28

KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

Kí hiệu	Diễn giải
SVM	Support Vector Machine
kNN	K – Nearest neighbor
NB	Naïve Bayes
TB	Trung bình

MỞ ĐẦU

Trong thời gian gần đây, song song với sự bùng nổ của Internet là sự phát triển rõ rệt của Web và thương mại điện tử, những người dùng Internet ngày càng trở nên gần gũi với các trang blog, diễn đàn, hay các trang Web thương mại điện tử. Từ đó, nảy sinh ra số lượng lớn dữ liệu mang quan điểm tồn tại trên các trang này. Bên cạnh đó, nhu cầu về thương mại, quản lí, xã hội ngày càng tăng, để giải quyết bài toán trên, các nghiên cứu về phân loại quan điểm ngày càng được chú trọng và quan tâm nhiều hơn và đối tượng được nghiên cứu cũng ngày càng đa dạng hơn. Đối tượng có thể là dịch vụ, sản phẩm, thậm chí là một sự kiện, cá nhân, hay tổ chức nào đó…Bằng những thành tựu nghiên cứu đã đạt được ở nhiều loại đối tượng, điều này không những cho thấy tầm quan trọng của vấn đề phân loại quan điểm trong nhiều lĩnh vực mà còn là vấn đề đầy mới mẻ và tiềm năng khai thác.

Thực tế, mỗi cá nhân đều đã ít nhất một thậm chí là thường xuyên tham khảo ý kiến cộng đồng mạng về một đối tượng, một vấn đề hay một dịch vụ nào đấy. Đối với nhà sản xuất các tổ chức, họ quan tâm đến việc sản phẩm được khác hàng đón nhận ra sao, đánh giá của khách hàng là cơ sở để tiến hành cải tiến và hoàn thiện sản phẩm công nghệ của mình. Thậm chí là với các tổ chức này có thể nghiên cứu, đánh giá mức độ sử dụng của khách hàng đối với các sản phẩm của họ, xác định thị hiếu khách hàng cũng như chất lượng, đồng thời có thể dự đoán xu hướng tiêu dùng tương lai để đưa ra những chiến lược sản xuất hay kinh doanh phù hợp.

Nhu cầu tìm hiểu, tham khảo để sở hữu các sản phẩm công nghê luôn tăng cao là hệ quả tất yếu của công nghệ bủng nổ với các sản phẩm thông minh, gân gũi và thân thiện từ máy tính để bàn, laptop (máy tính cá nhân), tablet (máy tính bảng), note, smartphone … Nên đối tượng được lựa chọn mà đồ án đề cập tới sẽ là các sản phẩm công nghệ, ngôn ngữ đánh giá được áp dụng là tiếng Việt.

Mục tiêu của đồ án là nghiên cứu và lựa chọn phương pháp phù hợp cho bài toán phân loại quan điểm người dùng về sản phẩm công nghệ và xây dựng hệ thống áp dụng phương pháp và kĩ thuật mà đồ án đã đưa ra.

Nội dung đồ án được chia làm 4 chương với nội dung tóm tắt như sau: Chương 1: "*Giới thiệu chung về bài toán phân loại quan điểm"*.

Trong chương 1 cung cấp cái nhìn khai quát nhất về bài toán phân loại, phân loại dữ liệu và giới thiệu về bài toán phân loại quan điểm.

Chương 2: "*Phương pháp phân loại quan điểm và các nghiên cứu liên quan"*.

Trình bày một số phương pháp phân loại quan điểm và các nghiên cứu có liên quan cùng với hiệu quả mà chúng đem lại dựa trên những phương pháp đã nêu.

Chương 3: *"Xây dựng hệ thống phân loại quan điểm người dùng đối với sản phẩm công nghệ"*.

Trong chương này sẽ trình bày quá trính xây dựng hệ thống cho bài toán phân loại quan điểm người dùng một cách chi tiết.

Chương 4: *"Thực nghiệm và đánh giá kết quả"*

Trình bày phương pháp thực nghiệm, đánh giá kết quả của *hệ thống phân loại quan điểm người dùng đối với sản phẩm công nghệ* và đưa ra nhận xét cũng như phương hướng phát triển sau này.

CHƯƠNG 1: GIỚI THIỆU VỀ BÀI TOÁN PHÂN LOẠI QUAN ĐIỂM

Khoảng nửa thế kỉ trở lại đây, phân loại (phân lớp) quan điểm nói riêng và khai phá quan điểm nói chung đã trở thành một trong những lĩnh vực nghiên cứu của xử lý ngôn ngữ tự nhiên và được nghiên cứu rộng rãi trong khai phá dữ liệu, khai phá Web và khai phá văn bản. Mặc dù việc phân loại quan điểm phải dựa trên văn bản ngôn ngữ tự nhiên là rất khó, vì văn bản ngôn ngữ tự nhiên là loại dữ liệu phi cấu trúc nhưng kết quả mà nó đem lại không hề nhỏ và một trong những kỹ thuật được sử dụng rộng rãi nhất hiện nay đề cập để giải quyết bài toán phân loại quan điểm là kỹ thuật phân lớp. Việc giới thiệu về phân lớp và bài toán phân loại quan điểm và một số khái niệm liên quan sẽ được trình bày trong chương 1.

1.1. Một số khái niệm về phân lớp

Phân lớp (hay phân loại) là tiến trình xử lý nhằm xếp các mẫu dữ liệu hay các đối tượng vào một trong các lớp đã được định nghĩa trước. Tuy nhiên, phân lớp là một hoạt động tiềm ẩn trong tư duy con người khi nhận dạng về thế giới thực, đóng vai trò quan trọng làm cơ sở đưa ra các dự báo, các quyết định. Phân lớp và cách mô tả các lớp giúp cho tri thức được định dạng và lưu trữ trong đó.

Đối với một đối tượng, hiện tượng, chúng ta dựa vào các đặc trưng của chúng, tức là, chúng ta chỉ xem xét biểu diễn của đối tượng, hiện tượng này trong một không gian hữu hạn chiều, mỗi chiều tương ứng với một đặc trưng đã được lựa chọn.

Cụ thể, bài toán phân lớp được biểu diễn như sau:

- Cho tập các mẫu đã được phân lớp từ trước.
- Mục đích: Gán các mẫu mới vào các lớp với độ chính xác cao nhất có thể
- Cho cơ sở dữ liệu $X = \{x_1, x_2, x_3, …., x_n\}$ và tập các lớp $C = \{c_1, c_2, …., c_n\}$, phân lớp là bài toán xác định hàm ánh xạ f: $X \rightarrow C(X)$ sao cho mỗi t_i được gán vào một lớp.

1.2. Một số kĩ thuật được sử dụng trong bài toán phân lớp dữ liệu

Phân lớp dữ liệu là kĩ thuật dựa trên tập huấn luyện và những giá trị hay là nhãn của lớp trong một t huộc tính phân lớp và sử dụng nó trong việc phân lớp dữ liệu mới.

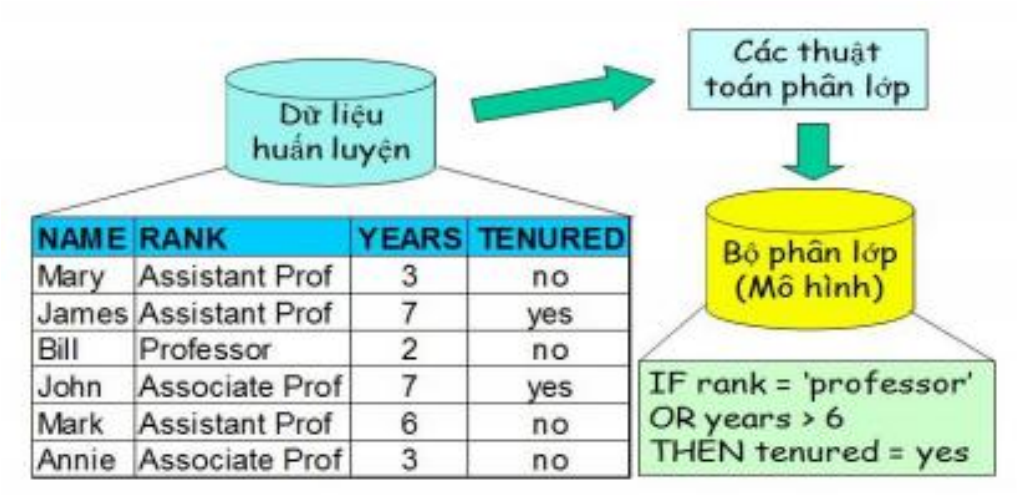
Qui trình phân lớp:

- *Bước 1: Học (learning) - xây dựng mô hình* từ tập huấn luyện.

Mỗi bộ/mẫu dữ liệu được phân vào một lớp được xác định trước. Lớp của một bộ/mẫu dữ liệu được xác định bởi thuộc tính gán nhãn lớp.

Tập các bộ/mẫu dữ liệu huấn luyện- tập huấn luyện- được dùng để xây dựng mô hình.

Mô hình được biểu diễn bởi các luật phân lớp, các cây quyết định hoặc các công thức toán học.



NAME	RANK	YEARS	TENURED
Mary	Assistant Prof	3	no
James	Assistant Prof	7	yes
Bill	Professor	2	no
John	Associate Prof	7	yes
Mark	Assistant Prof	6	no
Annie	Associate Prof	3	no

Hình 1.1: Ví dụ xây dựng mô hình

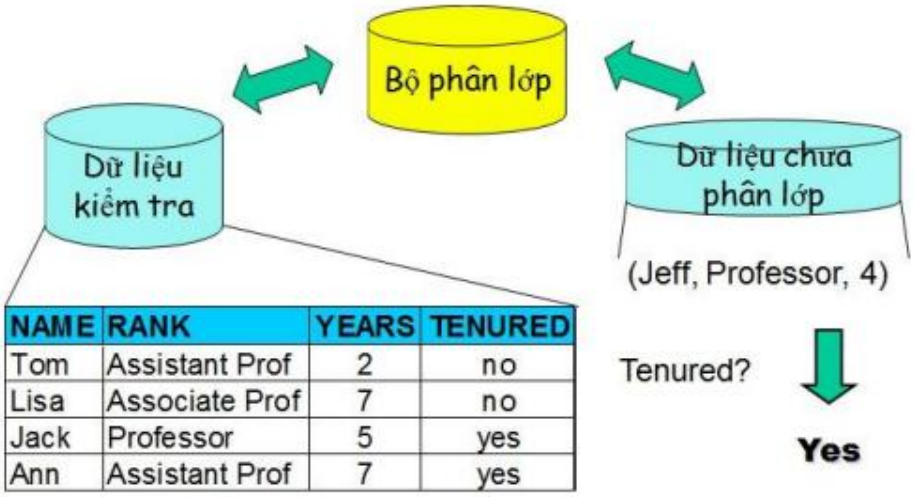
- *Bước 2: Phân lớp (Classification) - sử dụng mô hình* đã xây dựng từ quá trình "học". Tức là kiểm tra tính đúng đắn của mô hình và dùng nó để phân lớp dữ liệu.

Phân lớp cho những đối tượng mới hoặc chưa được phân lớp

Đánh giá độ chính xác của mô hình

Lớp biết trước của một mẫu/bộ dữ liệu đem kiểm tra được so sánh với kết quả thu được từ mô hình.

Tỉ lệ chính xác bằng phần trăm các mẫu/bộ dữ liệu được phân lớp đúng bởi mô hình trong số các lần kiểm tra



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Lisa	Associate Prof	7	no
Jack	Professor	5	yes
Ann	Assistant Prof	7	yes

Hình 1.2: Sử dụng mô hình

Như vậy, yếu tố đầu tiên để quyết định việc phân lớp có độ chính xác cao hay không là phụ thuộc vào quá trình học (learning). Nói cách khác, quá trình xây dựng mô hình sẽ quyết định mỗi đối tượng, hiện tượng, vật thể sẽ thuộc vào lớp nào từ chính các đặc trưng dữ liệu của chúng. Do đó, để có một mô hình học được cho là đem lại độ chính xác cao trong phân lớp thì việc lựa chọn phương pháp học sao cho phù hợp với tập dữ liệu huấn luyện ban đầu cũng rất quan trọng.

1.3. Nhóm các giải thuật học máy

Học máy (hay máy học – Machine learning) là một thành phần quan trọng của trí tuệ nhân tạo nhằm nghiên cứu và phát triển các phương pháp, kỹ thuật giúp cho các hệ thống hay máy tính có khả năng học.

Học máy có ứng dụng rộng khắp trong các ngành khoa học/sản xuất, đặc biệt những ngành cần phân tích khối lượng dữ liệu khổng lồ. Trong đó, xử lý ngôn ngữ tự nhiên (Natural Language Processing) bao gồm: xử lý văn bản, giao tiếp người – máy,… là một trong những ứng dụng hàng đầu. Ngoài ra còn có một số ứng dụng khác như: nhận dạng tiếng nói, chữ viết tay, vân tay, thị giác máy (Computer Vision), tìm kiếm (Search Engine), chẩn đoán trong y tế: phân tích ảnh X-quang, các hệ chuyên gia chẩn đoán tự động…

1.3.1. Học có giám sát (Supervised Learning)

Học có giám sát là học với tập dữ liệu huấn luyện ban đầu hoàn toàn được gán nhãn từ trước. Học có giám sát là phương pháp học sử dụng cho lớp bài toán phân lớp, phân loại (Classification).

Để thực hiện phân lớp, trước tiên phải chuẩn bị một tập dữ liệu huấn luyện (trainning data set), để có *tập dữ liệu huấn luyện* phải thực hiện gán nhãn cho dữ liệu ban đầu, đây được gọi là quá trình thu thập tập huấn luyện.

Lựa chọn một thuật toán phân lớp xây dựng bộ phân lớp để *học* tập dữ liệu huấn luyện. Hay nói cách khác, dùng tập dữ liệu huấn luyện để huấn luyện bộ phân lớp. Thuật ngữ *học có giám sát* được hiểu là *học* tập dữ liệu đã được gán nhãn trước (các dữ liệu kèm theo nhãn tương ứng này coi như đã được giám sát bởi người thực hiện gán nhãn).

Sử dụng một tập dữ liệu kiểm tra (test data set) đã được gán nhãn trước, để kiểm tra tính đúng đắn của bộ phân lớp. Sau đó, có thể dùng bộ phân lớp để phân lớp cho các dữ liệu mới.

1.3.2. Học không giám sát (Unsupervised Learning)

Học không giám sát là học với tập dữ liệu huấn luyện ban đầu hoàn toàn chưa được gán nhãn. Học không giám sát là phương pháp học sử dụng cho lớp bài toán gom cụm, phân cụm (Clustering). Tức là, máy tính chỉ được xem các mẫu không có đầu ra, sau đó máy tính phải tự tìm cách phân loại các mẫu này và các mẫu mới.

Để thực hiện phân cụm, trước tiên cần một tập dữ liệu huấn luyện (training dataset) – là một tập các ví dụ học (training examples/instances). Trong đó, mỗi ví dụ học chỉ chứa thông tin biểu diễn (ví dụ: một vector các giá trị thuộc tính), mà không có bất kỳ thông tin gì về nhãn lớp hoặc giá trị đầu ra mong muốn (expected output).

1.3.3. Học bán giám sát (Semi - Supervised Learning)

Học bán giám sát là học với tập dữ liệu huấn luyện gồm cả dữ liệu đã được gán nhãn và dữ liệu chưa được gán nhãn. Tuỳ vào từng mục đích cụ thể, học bán giám sát có thể được áp dụng cho bài toán phân lớp hoặc phân cụm.

Trong thực tế, để có được một tập dữ liệu có chất lượng và đã được gán nhãn của một lĩnh vực, thường được thực hiện thủ công bằng tay bởi người có nhiều kinh nghiệm về lĩnh vực đó. Vì vậy, dữ liệu đã được gán nhãn thường ít và đắt. Trong khi đó, dữ liệu chưa được gán nhãn lại rất nhiều và phong phú. Phương pháp học bán giám sát (hay học nửa giám sát) được đặt ra để tận dụng cả hai nguồn dữ liệu này.

1.3.4. Học tăng cường

Máy tính đưa ra quyết định hành động (*action*) và nhận kết quả phản hồi (*response/reward*) từ môi trường (*environment*). Sau đó máy tính tìm cách chỉnh sửa cách ra quyết định hành động của mình.

1.4. Giới thiệu bài toán phân loại quan điểm người dùng về sản phẩm công nghệ

1.4.1. Nhu cầu hiện nay

Trước khi Internet và web trở nên gần gũi với con người như hiện nay, việc tiếp cận ý kiến người dùng, chuyên gia tư vấn về một lĩnh vực nào đó là điều khó khăn. Tuy nhiên, với sự bùng nổ Internet ngày nay, và sự phát triển của Web đã giúp chúng ta có thể dễ dàng tham khảo ý kiến cũng như kinh nghiệm của người dùng khác, người có chuyên môn một cách thuận tiện và dễ dàng. Và ngược lại, ngày càng có nhiều và nhiều hơn nữa những người sẵn sàng chia sẻ những ý kiến, quan điểm của mình, đưa ra những đánh giá về bất kì một đối tượng nào như các sản phẩm trên các trang web thương mại điện tử, hay trình bày quan điểm về hầu hết thứ gì trên diễn đàn, nhóm thảo luận, blog hoặc đóng góp ý kiến ngay dưới một bài báo vừa đọc. Điều này đồng nghĩa với việc lượng thông tin phản hồi khách quan thu được từ chính người sử dụng Internet là rất lớn, đem lại rất nhiều lợi ích với các cá nhân nói riêng và các tổ chức nói chung. Với cá nhân, lượng thông tin này cung cấp cho người mua hàng cái nhìn toàn diện hơn về sản phẩm mà mình định mua. Đối với nhà sản xuất, dựa trên lượng thông tin phản hồi khách hàng, họ sẽ nâng cấp, nghiên cứu và đề xuất chiến lược kinh doanh sao cho phù hợp với thị trường.

Tuy nhiên, với lượng thông tin khổng lồ và đa dạng đến từ nhiều nguồn khác nhau, làm thế nào để khai thác một cách khoa học và triệt để trong khi lượng thông tin dư thừa là không hề nhỏ, các ý kiến nhận xét thường ẩn trong các bài viết dài và gây khó khăn cho người đọc.

Như vậy, không chỉ các cá nhân mà các công ty, nhà sản xuất, các tổ chức cũng đều quan tâm đến "một hệ thống" có khả năng tự động phân loại quan điểm người dùng.

1.4.2. Phát biểu bài toán

Cho trước một câu hay một tài liệu chứa quan điểm về một sản phẩm công nghệ bất kì, hãy phân loại xem câu hay tài liệu đó thể hiện quan điểm mang xu hướng tích cực (positive) hay tiêu cực (negative).

Input: Một tập dữ liệu chứa ý kiến đánh giá về một đối tượng sản phẩm công nghệ cụ thể.

Output: Mỗi văn bản được chia vào một lớp (tích cực hay tiêu cực) dựa theo sự phân loại về mặt ý nghĩa của câu/tài liệu mang quan điểm.

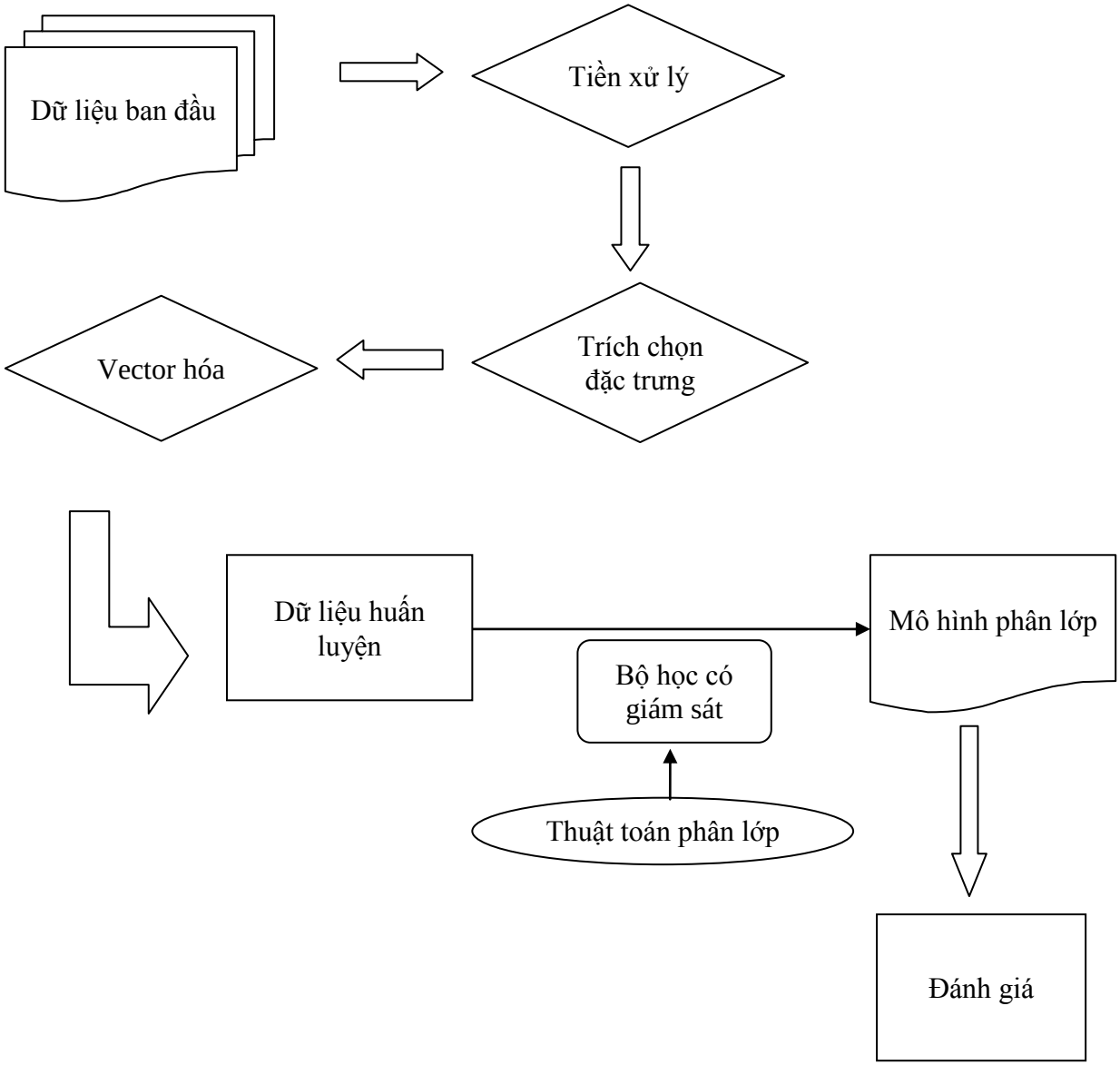
Có thể nhận thấy, bài toán phân loại quan điểm, thực chất là chính bài toán phân lớp (phân loại) dữ liệu, với phân lớp là một hình thức học được giám sát, quá trình "học" được "giám sát" bởi tri thức của các phân lớp cùng các mẫu huấn luyện và quan điểm trong bài toán được đề cập ở trên được thể hiện ở dạng dữ liệu văn bản text.

Phân loại tài liệu theo hướng quan điểm thật sự là vấn đề thách thức và khó khăn trong lĩnh vực xử lý ngôn ngữ bởi đó chính là bản chất phức tạp trong ngôn ngữ của con người, đặc biệt là sự đa nghĩa và nhập nhằng nghĩa của ngôn ngữ. Sự nhập nhằng này rõ ràng sẽ ảnh hưởng đến độ chính xác bộ phân loại của chúng ta một mức độ nhất định.

1.4.3. Mô hình tổng quát

Dựa trên qui trình phân lớp đã trình bày ở mục 1.2 để đưa ra quá trình xử lí cho bài toán phân loại quan điểm người dùng về một sản phẩm công nghệ. Quá trình xử lý này dựa trên kiểu học có giám sát bao gồm 2 giai đoạn: giai đoạn huấn luyện (training) và giai đoạn phân lớp.

- *Giai đoạn 1: Huấn luyện*



Hình 1.3: Mô hình chi tiết giai đoạn huấn luyện

Input: Tập dữ liệu huấn luyện và thuật toán huấn luyện

Output: Mô hình phân lớp

Ban đầu, dữ liệu chúng ta có là tài liệu mang quan điểm người dùng nhưng ở dưới dạng văn bản text. Trong tập dữ liệu này có chứa rất nhiều dữ liệu dư thừa, thường là các từ không mang quan điểm, còn được gọi là từ dừng (stop word). Không những thế, dữ liệu dạng text ban đầu cũng cần phải được chuyển đổi thành dạng thích hợp cho việc phân loại sau này. Việc đầu tiên cần làm đối với tập dữ liệu này là tiền xử lý, với mục đích làm thu gọn dung lượng dữ liệu thuân tiện cho lưu trữ dữ liệu cũng như tăng độ chính xác của dữ liệu đầu vào.

Sau bước tiền xử lý dữ liệu, mỗi phần tử trong tập dữ liệu này sẽ được gán nhãn vào một trong hai lớp (cụ thể là 2 nhãn 1 và -1 thể hiện nghĩa tích cực và tiêu cực). Mỗi phần tử trong tập dữ liệu đầu vào sẽ được biểu diễn dưới dạng ($\vec{x}$,c), trong đó $\vec{x}$ là vector biểu diễn văn bản trong tập dữ liệu đầu vào.

Sau đó, mô hình phân lớp sẽ được xây dựng dựa trên việc phân tích các phần tử dữ liệu đã được gán nhãn ở trên. Tập mẫu các dữ liệu này được gọi là tập dữ liệu huấn luyện (training data set). Nhãn của tập dữ liệu này được con người xác định trước khi xây dựng mô hình nên phương pháp này được gọi là học có giám sát. Trong mô hình phân lớp, thuật toán phân lớp giữ vai trò quyết định tới sự thành công của mô hình phân lớp.

Tuy nhiên, không thể bỏ qua bước đánh giá độ chính xác của mô hình và sử dụng một tập dữ liệu kiểm tra (test data set), nếu độ chính xác là chấp nhận được, mô hình sẽ được sử dụng để xác định nhãn lớp cho các dữ liệu mới khác.

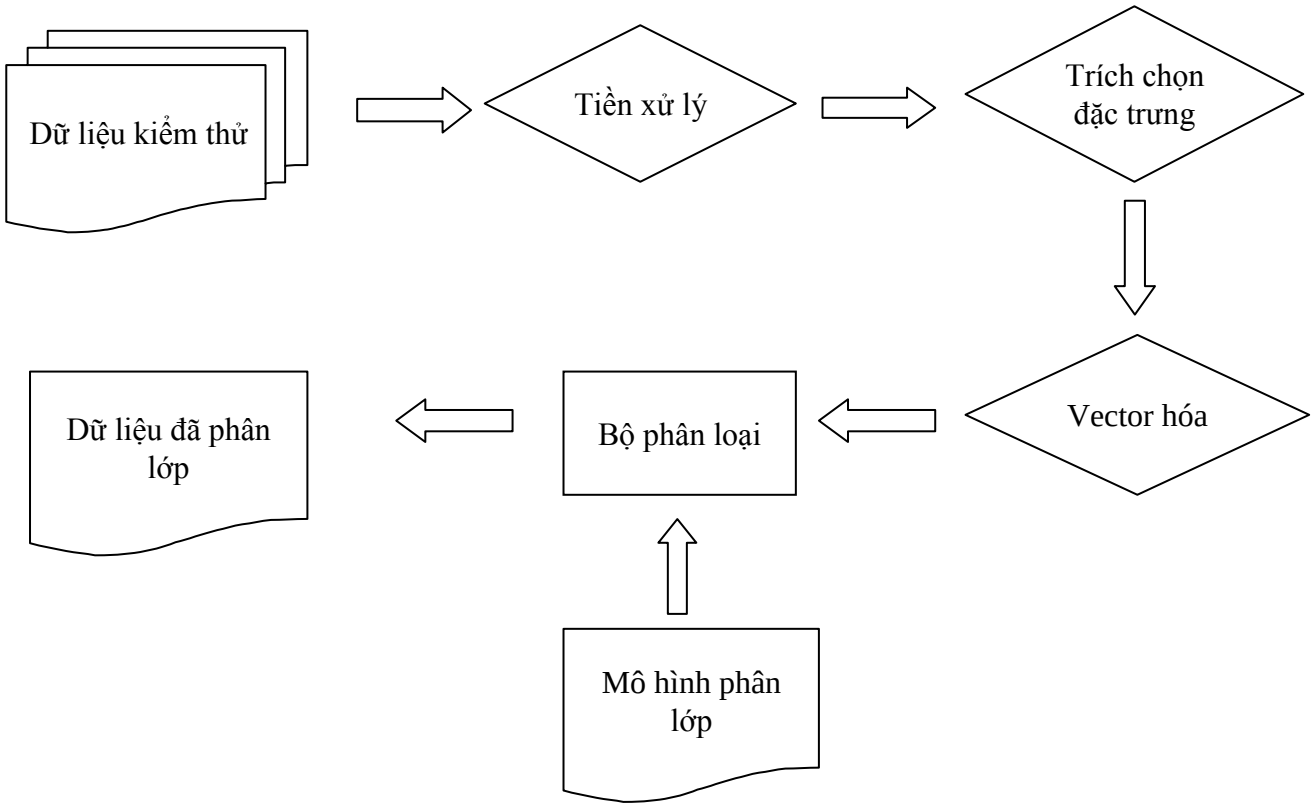
Để giải quyết bài toán phân lớp dữ liệu, có nhiều phương pháp tiếp cận được đề xuất như phương pháp Bayes, phương pháp cây quyết định (Decision tree), phương pháp Maximun Entropy, phương pháp SVM… Trong đó, phương pháp SVM – Support Vector Machine đạt độ chính xác tương đối cao trong khoảng 78,7% đến 82,9%, đây cũng là phương pháp sẽ được đề cập đến trong chương 3 và sử dụng để giải quyết bài toán phân loại quan điểm người dùng.

- *Giai đoạn 2: Phân lớp*

Mô hình phân lớp sau khi được tạo ra từ giai đoạn huấn luyện sẽ được áp dụng cho dữ liệu mang quan điểm mới cần phân loại.

Input: Tập dữ liệu chưa phân lớp.

Output: Tập dữ liệu đã được phân lớp.



Hình 1.4: Mô hình chi tiết giai đoạn phân lớp.

CHƯƠNG 2: MỘT SỐ PHƯƠNG PHÁP SỬ DỤNG TRONG PHÂN LOẠI QUAN ĐIỂM VÀ CÁC NGHIÊN CỨU LIÊN QUAN

Trong những năm qua, có nhiều phương pháp phân lớp (phân loại) dữ liệu đã được các nhà khoa học trong nhiều lĩnh vực khác nhau đề xuất để giải quyết bài toán phân lớp như phương pháp Bayes, cây quyết định, K – láng giềng gần nhất, Maximum Entropy cực đại, máy hỗ trợ vector (Support Vector Machine - SVM) … Tùy theo mục tiêu bài toán và dữ liệu đầu vào mà các nhà nghiên cứu sẽ lựa chọn phương pháp thích hợp. Một số phương pháp phân loại cụ thể sẽ được trình bày trong chương 2 nhằm mục đích cung cấp cái nhìn tổng quan nhất định cùng các nghiên cứu liên quan trước sẽ làm rõ hiệu năng của từng phương pháp phân loại.

2.1. Một số phương pháp sử dụng trong phân loại quan điểm

2.1.1. Phương pháp Naïve (NB)

Naïve Bayes là phương pháp phân loại dựa vào xác suất được sử dụng rộng rãi trong lĩnh vực máy học. Ban đầu phương pháp NB được sử dụng trong lĩnh vực phân loại, sau đó được dùng phổ biến trong nhiều lĩnh vực khác như trong các công cụ tìm kiếm, các bộ lọc mail…

- *Ý tưởng*

Ý tưởng cơ bản của cách tiếp cận Naïve Bayes là sử dụng xác suất có điều kiện giữa từ và chủ đề để dự đoán xác xuất chủ đề của một văn bản cần phân loại. Mấu chốt của phương pháp này là giả định sự xuất hiện của tất cả các từ trong văn bản đều độc lập với nhau. Theo giả định này, NB không sử dụng sự phụ thuộc của nhiều từ vào một chủ đề, không sự dụng việc kết hợp các từ để đưa ra phán đoán chủ đề và do vậy việc tính toán NB chạy nhanh hơn các phương pháp khác với độ phức tạp theo hàm số mũ.

- *Thuật toán NB*

Thuật toán Naïve Bayes dựa trên định lý Bayes được phát biểu như sau:

$$P(Y|X) = \frac{P(XY)}{P(X)} = \frac{P(X|Y)P(Y)}{P(X)} \quad (2.1)$$

Áp dụng trong bài toán phân loại, các dữ kiện gồm có:

- D: tập dữ liệu huấn luyện đã được vector hóa dưới dạng $\vec{x} = (x_1, x_2, \dots, x_n)$
- C_i: phân lớp i, với i = {1,2,…,m}
- Các thuộc tính độc lập điều kiện đôi một với nhau

Theo định lý Bayes:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \quad (2.2)$$

Theo tính chất độc lập điều kiện:

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) \quad (2.3)$$

Trong đó:

$P(C_i|X)$ là xác suất thuộc phân lớp i khi biết trước mẫu X.

$P(C_i)$ xác suất là phân lớp i.

$P(x_k|C_i)$ xác suất thuộc tính thứ k mang giá trị x_k khi đã biết X thuộc phân lớp i.

Các bước thực hiện thuật toán Naïve Bayes:

Bước 1: Huấn luyện Naïve Bayes (dựa vào tập dữ liệu), tính $P(C_i)$ và $P(x_k|C_i)$

Bước 2: Phân lớp $X^{new} = (x_1, x_2, \dots, x_n)$, ta cần tính xác suất thuộc từng phân lớp khi đã biết trước X^{new}. X^{new} được gán vào lớp có xác suất lớn nhất theo công thức

$$\max_{C_i \in C}(P(C_i) \prod_{k=1}^{n} P(x_k|C_i)) \quad (2.4)$$

- *Thuật toán NB cho bài toán phân lớp*
  - Huấn luyện: tính $P(C_i)$ và $P(x_k|C_i)$

    Đầu vào:
    - Các vector đặc trưng của văn bản trong tập huấn luyện (Ma trận MxN, với M là số vector đặc trưng trong tập huấn luyện, N là số đặc trưng của vector).
    - Tập nhãn/lớp cho từng vector đặc trưng của tập huấn luyện

    Đầu ra: Các giá trị xác suất $P(C_i)$ và $P(x_k|C_i)$.

    Công thức tính $P(C_i)$:

$$P(C_i) = \frac{|docs_i|+1}{|total\ docs|+m} \quad (2.5)$$

Trong đó:

|docs$_i$|: số văn bản của tập huấn luyện thuộc phân lớp i.

|total docs|: số văn bản trong tập huấn luyện

m số phân lớp

Cài đặt:

- Khởi tạo mảng A, B có kích thước m.
- Duyệt qua các văn bản trong tập dữ liệu, đếm số văn bản trong mỗi phân lớp lưu vào A.
- Tính xác suất cho từng phân lớp theo công thức (2.1) và lưu vào mảng B.

Công thức tính $P(x_k|C_i)$:

$$P(x_k|C_i) = \frac{|docs_{x_k i}|+1}{|docs_i|+d_k} \quad (2.6)$$

Trong đó:

$|docs_{x_k i}|$: Số văn bản trong trong phân lớp i có đặc trưng thứ k mang giá trị x_k. (hay số văn bản trong lớp i, có xuất hiện/không xuất hiện đặc trưng k)

$|docs_i|$: Số văn bản của tập huấn luyện thuộc phân lớp i.

d_k: Số giá trị có thể có của đặc trưng thứ k

Cài đặt:

- Với vector đặc trưng như mô tả bên trên, d_k ở đây mang giá trị là 2, tương ứng với xuất hiện và không xuất hiện. Do chỉ có 2 giá trị, ta có thể tính nhanh xác suất không xuất hiện theo công thức $P(\bar{x}) = 1 - P(x)$
- Khởi tạo mảng 3 chiều C, chiều 1 có kích thước là m (số phân lớp), chiều 2 có kích thước là N (số đặc trưng), chiều 3 có kích là 2 (d_k) để lưu các giá trị $P(x_k|C_i)$.
- Duyệt qua các văn bản trong tập dữ liệu, tiến hành thống kê các chỉ số cần thiết để tính xác suất $P(x_k|C_i)$ theo công thức trên và lưu vào mảng C.

- Phân lớp:
  - Đầu vào:

  Vector đặc trưng của văn bản cần phân lớp.

  Các giá trị xác suất $P(C_i)$ và $P(x_k|C_i)$.

  - Đầu ra:

  Nhãn/lớp của văn bản cần phân loại.

  Công thức tính xác suất thuộc phân lớp i khi biết trước mẫu X

$$P(C_i|X) = P(C_i)\prod_{k=1}^{n} P(x_k|C_i) \quad (2.7)$$

Dựa vào vector đặc trưng của văn bản cần phân lớp, áp dụng công thức trên tính xác suất thuộc từng phân lớp cho văn bản, và chọn ra lớp có xác suất cao nhất.

- Naïve Bayes là một công cụ rất hiệu quả trong một số trường hợp, đặc biệt là đối với bài toán phân loại văn bản nhiều chủ đề vì NB là một thuật toán phân loại tuyến tính. Tuy nhiên trong trường hợp dữ liệu huấn luyện ít và các tham số dự đoán (hay còn gọi là không gian đặc trưng) có chất lượng kém thì việc sử dụng phương pháp Naïve Bayes sẽ đem lại kết quả không cao, thậm chí là hiệu quả rất thấp.

Naïve Bayes có ưu điểm là cài đặt đơn giản, tốc độ nhanh, dễ dàng cập nhật dữ liệu huấn luyện mới và có tính độc lập cao với tập huấn luyện, có thể kết hợp sử dụng nhiều tập huấn luyện khác nhau.

2.1.1. Phương pháp K - Nearest Neighbor (kNN)

K-Nearest Neighbors algorithm (KNN) được sử dụng rất phổ biến trong lĩnh vực Data Mining. KNN là phương pháp để phân lớp các đối tượng dựa vào khoảng cách gần nhất giữa đối tượng cần xếp lớp với tất cả các đối tượng trong Training Data.

Một đối tượng được phân lớp dựa vào k láng giềng của nó. K là số nguyên dương được xác định trước khi thực hiện thuật toán. Người ta thường dùng khoảng cách Cosine để tính khoảng cách giữa các đối tượng.

- *Thuật toán KNN dùng trong phân lớp (phân loại) văn bản được mô tả như sau:*
  - Xác định giá trị tham số K (số láng giềng gần nhất)
  - Tính khoảng cách giữa đối tượng cần phân lớp với tất cả các đối tượng trong training data (thường sử dụng khoảng cách Euclidean, Cosine…).

- Sắp xếp khoảng cách theo thứ tự tăng dần và xác định k láng giềng gần nhất với đối tượng cần phần lớp.
- Lấy tất cả các lớp của k láng giềng gần nhất đã xác định.
- Dựa vào phần lớn lớp của láng giềng gần nhất để xác định lớp cho đối tượng.

- *Ý tưởng :*

Khi cần phân loại một dữ liệu mới, thuật toán sẽ tính khoảng cách của tất cả các văn bản trong tập huấn luyện đến văn bản này để tìm ra *k* văn bản gần nhất (gọi là k "láng giềng"), khoảng cách này thường là khoảng cách Euclide, Cosine…Sau đó, dùng khoảng cách này đánh trọng số cho tất cả các chủ đề. Trọng số của một chủ đề chính là tổng tất cả các khoảng cách ở trên của các văn bản trong k láng giềng có cùng chủ đề, chủ đề nào không xuất hiện trong k láng giềng sẽ có trọng số bằng 0. Các chủ để sẽ được sắp xếp theo mức độ trọng số giảm dần và các chủ đề có trọng số cao sẽ được chọn là chủ đề của văn bản cần phân loại.

Khoảng cách giữa 2 văn bản chính là độ tương tự giữa 2 văn bản đó, 2 văn bản có giá trị độ tương tự càng lớn thì khoảng cách càng gần nhau.

Ví dụ: Dùng công thức Cosine để tính độ tương tự giữa 2 văn bản:

$$sim(\vec{x},\vec{y}) = \cos(\vec{x},\vec{y}) = \frac{\vec{x}.\vec{y}}{\|\vec{x}\|.\|\vec{y}\|} \quad (2.8)$$

Văn bản A: Tôi là học sinh.

Văn Bản B: Tôi là sinh viên.

Văn bản C: Tôi là giáo viên.

Biểu diễn văn bản theo vector:

	Tôi	là	học	sinh	viên	giáo
Văn bản A	1	1	1	1	0	0
Văn bản B	1	1	0	1	1	0
Văn bản C	1	1	0	0	1	1

Vector A = (1,1,1,1,0,0)

Vector B = (1,1,0,1,1,0)

Vector C = (1,1,0,0,1,1)

$$sim(\vec{A},\vec{B}) = \cos(\vec{A},\vec{B}) = \frac{3}{\sqrt{4*4}} = 0.75$$

$$sim(\vec{A},\vec{C}) = \cos(\vec{A},\vec{C}) = \frac{2}{\sqrt{4*4}} = 0.5$$

Điều đó cho thấy văn bản A tương tự văn bản B hơn so với C.

Để chọn được tham số *k* tốt nhất cho việc phân loại, thuật toán phải chạy thử nghiệm trên nhiều giá trị khác nhau, giá trị *k* càng lớn thì thuật toán càng ổn định và sai sót thấp.

Hướng dẫn cài đặt:

Thông thường các thuật toán sẽ gồm 2 giai đoạn huấn luyện và phân lớp, riêng đối với thuật toán KNN do thuật toán này không cần tạo ra mô hình khi làm trên tập huấn luyện các văn bản đã có nhãn/lớp sẵn, nên không cần giai đoạn huấn luyện (giai đoạn huấn luyện của KNN là gán nhãn cho các văn bản trong tập huấn luyện bằng cách gom nhóm các văn bản có vector đặc trưng giống nhau thành cùng 1 nhóm).

Mô tả vector đặc trưng của văn bản: Là vector có số chiều là số đặc trưng trong toàn tập dữ liệu, các đặc trưng này đôi một khác nhau.Nếu văn bản có chứa đặc trưng đó sẽ có giá trị 1, ngược lại là 0.

Đầu vào:

- Vector đặc trưng của văn bản cần phân lớp.
- Các vector đặc trưng của văn bản trong tập huấn luyện (Ma trận MxN, với M là số vector đặc trưng trong tập huấn luyện, N là số đặc trưng của vector).
- Tập nhãn/lớp cho từng vector đặc trưng của tập huấn luyện.

Đầu ra:

- Nhãn/lớp của văn bản cần phân loại.

Quá trình phân lớp chi tiết gồm các bước sau:

- Xác định giá trị tham số K (số láng giềng gần nhất). Tùy vào mỗi tập huấn luyện (số lượng mẫu trong tập huấn luyện, không gian tập mẫu có phủ hết các trường hợp…) mà việc chọn số K sẽ ảnh hưởng đến kết quả phân lớp.
- Lần lượt duyệt qua các văn bản (được đại diện bằng vector đặc trưng của văn bản) trong tập huấn luyện và tính độ tương tự của văn bản đó với văn bản cần phân lớp.
- Sau khi đã có mảng các giá trị lưu độ tương tự của văn bản cần phân lớp với các văn bản trong tập huấn luyện, ta sắp xếp độ tương tự các văn bản theo thứ tự giảm dần (lưu ý đây là độ tương tự, độ tương tự càng lớn tức là khoảng cách càng gần) và lấy ra k văn bản đầu tiên trong mảng (tức là k văn bản gần với văn bản cần phân lớp nhất).
- Khởi tạo mảng A có độ dài bằng số phân lớp để lưu số văn bản của mỗi lớp. Duyệt qua k văn bản, đếm số văn bản trong từng phân lớp và lưu vào mảng.
- Duyệt qua mảng A, tìm lớp có số văn bản nhiều nhất và chọn là lớp cho văn bản mới.

2.1.2. Phương pháp sử dụng cây quyết định (Decision tree)

Phương pháp cây quyết định được đưa ra từ năm 1966 và được sử dụng rộng rãi nhất trong việc học quy nạp từ tập mẫu lớn. Đây là phương pháp học xấp xỉ các hàm mục tiêu có giá trị *rời* rạc. Phương pháp cây quyết định thường được sử dụng cho hai nhiệm vụ trong khai phá quan điểm là phân loại và dự báo. Mặc khác, cây quyết định còn có thể chuyển sang dạng biểu diễn tương đương dưới dạng cơ sở tri thức là các luật *nếu-thì*.

- *Ý tưởng:*

Bộ phân lớp cây quyết định biểu dễn dưới dạng cây mà mỗi nút được gán nhãn là một đặc trưng, mỗi nhánh là giá trị trọng số xuất hiện của đặc trưng trong văn bản cần phân lớp và mỗi lá là nhãn của phân lớp tài liệu. Việc phân lớp của một tài liệu d_j sẽ được duyệt đệ qui theo trọng số của những đặc trưng có xuất hiện trọng văn bản d_j. Thuật toán lặp đệ qui đến khi đạt đến nút lá và nhãn của d_j chính là nhãn của nút lá tìm được. Thông thường, việc phân lớp văn bản nhị phân sẽ tương thích với việc dùng cây nhị phân.

- Cách thực hiện:

Cấu trúc của cây quyết đinh gồm: các nút trong được gán nhãn bởi các thuật ngữ, nhãn của các cung tương ứng với tọng số của thuật ngữ trong tài liệu mẫu, nhãn của các lá tương ứng với nhãn của các lớp.

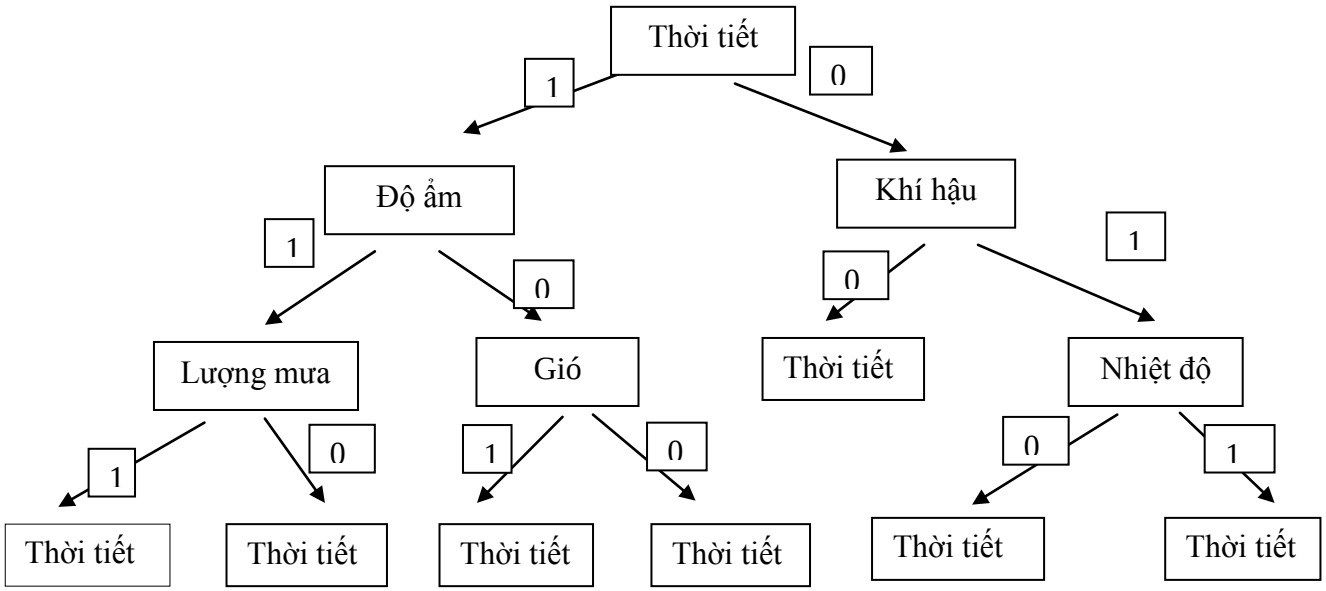
Cho tài liệu d_j, thực hiện việc so sánh các nhãn của một cung xuất phát từ một nút trong (tương ứng với một thuật ngữ nào đó) với trọng số của thuật ngữ này trong d_j, để quyết định nút trong nào sẽ được duyệt tiếp. Quá trình này được lặp từ nút gốc của cây cho tới khi nút được duyệt là một lá của cây. Kết thúc quá trình, nhãn của nút sẽ là nhãn của lớp được gán cho văn bản.

Ví dụ:

Ta có bảng dữ liệu gồm 10 tài liệu được mô tả bằng vector nhị phân thông qua 7 thuật ngữ "*thời tiết", "độ ẩm", "lượng mưa", "gió", "khí hậu", "thuyền", "nhiệt độ"*. Trong đó cột cuối cùng trong bảng là nhãn được gán cho từng tài liệu với chủ đề "*thời tiết"* , giá trị của tài liệu d_1 trong cột này bằng 1 tương ứng d_1 thuộc chủ đề thời tiết, nếu giá trị này băng 0 thì d_1 không thuộc chủ đề thời tiết.

Tài liệu	Thời tiết	Độ ẩm	Lượng mưa	Gió	Khí hậu	Thuyền	Nhiệt độ	Thời tiết
D_1	1	1	1	0	0	0	0	1
D_2	1	1	0	0	0	1	0	0
D_3	1	1	1	0	0	0	1	1
D_4	1	1	1	0	0	0	0	1
D_5	1	0	0	1	0	0	0	1
D_6	1	0	0	1	1	1	0	1
D_7	1	0	0	0	0	1	0	0
D_8	0	1	0	0	0	1	0	0
D_9	0	0	0	0	1	0	1	1
D_{10}	0	0	0	0	1	0	0	0

Bảng 2.1 Biểu diễn văn bản vector nhị phân



Hình 2.1. Xây dựng cây quyết định cho tập mẫu dùng để huấn luyện

Từ cây quyết định trên ta xây dựng được cơ sở tri thức dưới dạng luật *Nếu – Thì* như sau :

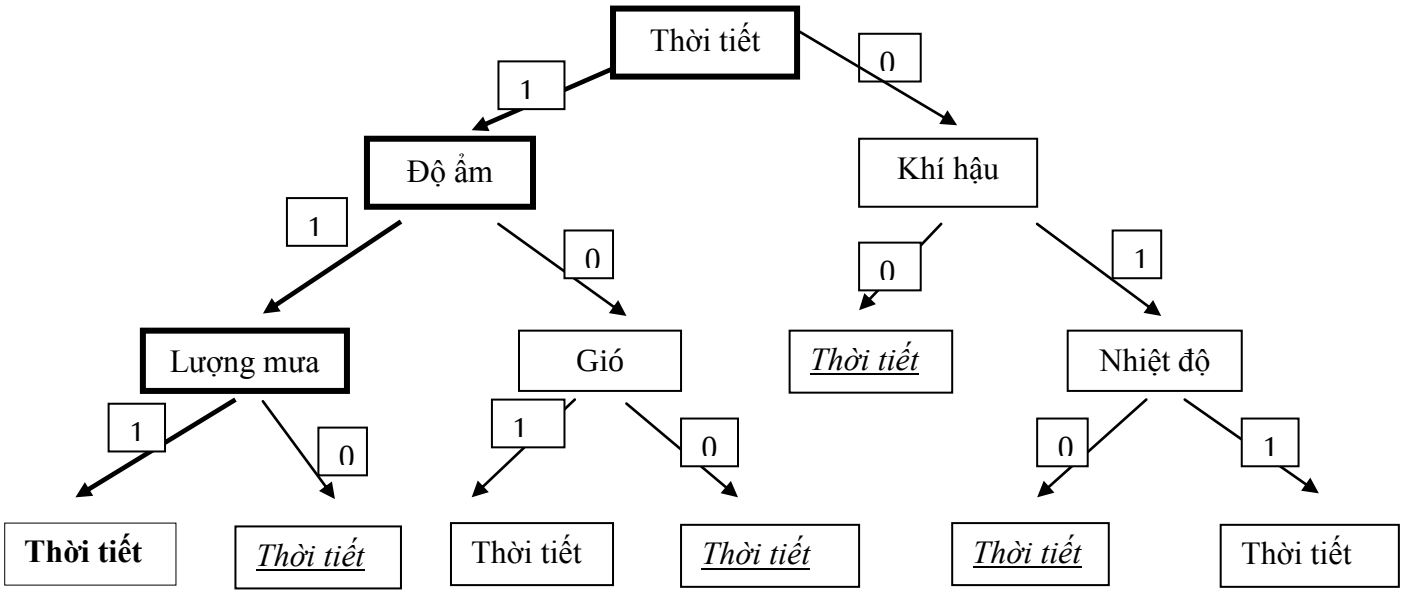
Nếu (thời tiết = 1 *)* và *(lượng mưa* = 1*)* và *(độ ẩm* = 1 *) Thì* class *thời tiết* = 1

Nếu (thời tiết = 1 *)* và *(lượng mưa* = 0*)* và *(độ ẩm* = 1 *) Thì* class *thời tiết* = 0

Nếu (thời tiết = 1 *)* và *(gió* = 0*)* và *(độ ẩm* = 0 *) Thì* class *thời tiết* = 0

Nếu (thời tiết = 1 *)* và *(gió* = 1*)* và *(độ ẩm* = 0 *) Thì* class *thời tiết* = 1

Nếu (thời tiết = 0 *)* và *(khí hậu* = 1*)* và *(nhiệt độ* = 0 *) Thì* class *thời tiết* = 0
Nếu (thời tiết = 0 *)* và *(khí hậu* = 1*)* và *(nhiệt độ* = 1 *) Thì* class *thời tiết* = 1
Nếu (thời tiết = 0) và *(khí hậu* = 1*) Thì* class *thời tiết* = 1
D = *(thời tiết, lượng mưa, khí hậu, độ ẩm, gió, thuyền, nhiệt độ)*= (1,1,1,0,0,1,0)
Quá trình tìm kiếm lời giải trên cây quyết định sẽ như sau :



Hình 2.2 : Quá trình tìm kiếm lời giải trên cây quyết đinh

Class *thời tiết = 1* , hay nói các khác văn bản d thước lớp văn bản nói về chủ đề thời tiết (lớp thời tiết).

Các thuật toán cây quyết định ngày càng được phát triển và cải tiến. Nhưng hầu hết các thuật toán này đều dựa vào cách tiếp cận từ trên xuống và chiến lược tìm kiếm tham lam trong không gian tìm kiếm của cây quyết đinh. Ưu điểm khác biệt nhất của phương pháp cây quyết định so với các phương pháp khác đó là nó có thể xử lý cả dữ liệu có giá trị bằng số và dữ liệu có giá trị là tên thể loại trong khi các phương pháp khác thường chuyên để phân tích các bộ dữ liệu chỉ gồm một loại biến.

2.1.3. Phương pháp SVM

Theo [1, 2], SVM là phương pháp phân lớp rất hiệu quả được Vapnik giới thiệu vào năm 1995 để giải quyết nhận dạng văn bản mẫu hai lớp sử dụng nguyên lý *Cực tiểu hóa rủi ro Cấu trúc* (Structural Risk Minization).

SVM sử dụng thuật toán học nhằm xây dựng một siêu phẳng làm cực tiểu hóa độ phân lớp sai của một đối tượng dữ liệu mới. Độ phân lớp sai của một siêu phẳng đặc trưng bởi khoảng các bé nhất tới siêu phẳng đấy.

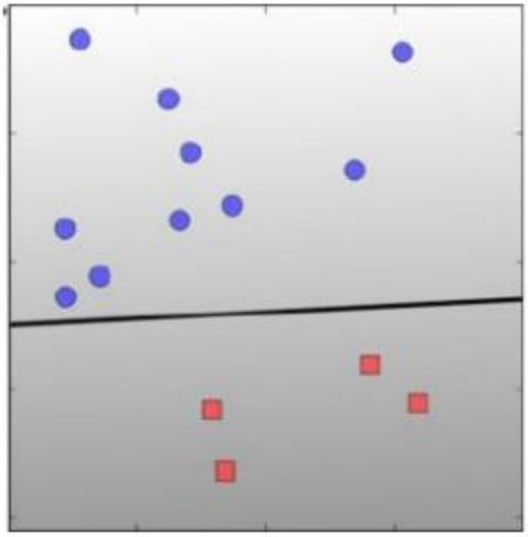
Như đã biết, phân lớp văn bản là một cách tiếp cận mới tạo ra tập phân lớp văn bản từ các mẫu cho trước. Phương pháp SVM có khả năng tính toán sẵn sang và phân lớp, nó trở thành lý thuyết học mà có thể chỉ dẫn những ứng dụng thực tế trên toàn cầu.

Đặc trưng cơ bản quyết định khả năng phân lớp là khả năng phân lớp những dữ liệu mới dựa vào những tri thức đã tích lũy được trong quá trình huấn luyện. Sau quá trình huấn luyện, nếu hiệu suất tổng quát hóa của bộ phân lớp cao thì thuật toán huấn luyện được đánh giá tốt. Hiệu suất tổng quát hóa phụ thuộc vào hai tham số là *sai số huấn luyện* và *năng lực* của máy học. Trong đó, sai số huấn luyện là tỉ lệ lỗi phân lớp trên tập dữ liệu huấn luyện. Còn năng lực của máy học được xác định bằng kích thước Vapnik – Chervonenkis (kích thước VC) [3]. Kích thước VC là một khái niệm quan trọng đối với một tập phân lớp. Đại lượng này được xác định bằng số điểm cực đại mà tập phân lớp có thể phân tách hoàn toàn trong không gian đối tượng. Một tập phân lớp tốt là tập phân lớp có năng lực thấp nhất (nghĩa là đơn giản nhất) và đảm bảo sai số huấn luyện nhỏ.

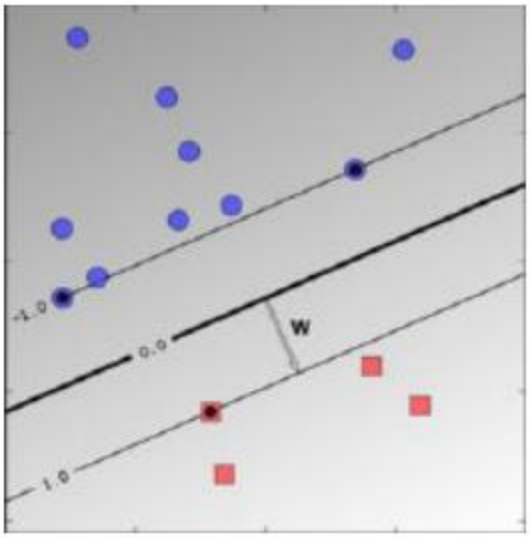
a, SVM cho bài toán phân lớp tuyến tính

Hình thức đơn giản của việc phân lớp (phân loại) là phân lớp nhị phân. Tức là phân biệt giữa các đối tượng thuộc về một trong hai lớp +1 (tích cực) và -1 (âm). SVM sử dụng hai khai niệm để giải quyết vấn đề này là phân lớp biên rộng và hàm kernel.

Ý tưởng của phân lớp biên rộng có thể được minh họa bởi sự phân lớp của các điểm trong không gian hai chiều. (hình 2.3) Một cách đơn giản để phân lớp các điểm này là sử dụng một đường thẳng để phân tách các điểm nằm ở một bên là dương và các điểm bên kia là âm. Nếu có hai đường thẳng phân chia tốt thì tâ có thể phân tách khá xa hai tập dữ liệu (hình 2.4). Đây là ý tưởng về sự phân chia biên rộng.



Hình 2.3: Một đường thẳng tuyến tính phân chia hai lớp điểm (hình vuông và hình tròn) trong không gian hai chiều. Ranh giới quyết định chia không gian thành hai tập tùy thuộc vào dấu của hàm f(x) = <w, x> +b.



Hình 2.4: Độ rộng biên lớn nhất được tính toán bởi một SVM tuyến tính. Khu vực giữa hai đường mảnh xác định miền biên với -1 ≤ <w, x> + b ≤ 1. Những điểm sáng hơn với chấm đen ở giữa gọi là các điểm Support Vectors, đó là những điểm gần biên quyết định nhất. Trong hình ví dụ, có ba Support Vectors trên các cạnh của vùng biên (fx)= -1 hoặc f(x)= 1) [10].

Các dữ liệu sử dụng gồm có các đối tượng có nhãn (là một trong hai nhãn +1 và -1). Lấy x biểu thị một vector với M phần tử x_j, (j = 1,…M) tức là một điểm trong một không gian vector M chiều. Các x_i ký hiệu biểu thị vector thứ *i* trong một tập dữ liệu $\{(X_i Y_i)\}_{i=1}^{n}$

Trong đó y_i là nhãn liên quan x_i. Các đối tượng x_i được gọi là đặc tính đầu vào.

Bên cạnh đó, một khái niệm quan trọng để xác định một phân lớp tuyến tính là tích vô hướng giữa hai vector$\langle w, x\rangle = \sum_{j=1}^{M} w_j x_i$, còn được gọi là tích *trong.* Phân lớp tuyến tính đươc dựa trên một hàm tuyến tính dạng:

$$f(x) = \langle w, x\rangle + b \quad (2.9)$$

Hàm *f(x)* là hàm đầu vào x, *f(x)* được sử dụng để quyết định làm thế nào để phân lớp *x*. Vector *w* được gọi là vector trọng số, và b được gọi là độ dịch. Trong không gian hai chiều các điểm ứng với phương trình $\langle w, x\rangle = 0$, tương ứng với một đường qua gốc tọa độ, trong không gian ba chiều thì nó là một mặt phẳng qua gốc tọa độ. Biến *b* sẽdịch chuyển mặt phẳng đi một lượng so với mặt phẳng qua gốc toạ độ. Mặt phẳng phân chia không gian thành hai không gian theo dấu của *f(x)*, nếu *f(x) >0* thì quyết định cho một lớp dương, lớp kia là lớp âm. Ranh giới giữa các vùng được phân lớp là dương và âm được gọi là ranh giới quyết định của các phân lớp. Ranh giới quyết định được xác định bởi một mặt phẳng (phương trình (1.1)) được cho là được tuyến tính bởi vì nó là tuyến tính đầu vào. Phân lớp với một ranh giới quyết định tuyến tính được gọi là phân lớp tuyến tính.

Với bất kỳ một tập dữ liệu khả tách tuyến tính có tồn tại một mặt phẳng phân lớp tất cả các điểm dữ liệu. Có nhiều mặt phẳng như vậy nhưng phải lựa chọn mặt phẳng

nào để đảm bảo thời gian huấn luyện ngắn về phân lớp một cách chính xác. Thực tế quan sát cũng như thống kê cho thấy rằng phân lớp siêu phẳng tách biệt chính xác với một biên độ lớn. Ở đây, biên của môt phân lớp tuyến tính được định nghĩa là khoảng cách gần nhất để quyết định ranh giới, như thể hiện trong hình 2.4. Có thể điều chính *b* siêu phẳng tách các điểm tương ứng. Hơn nữa nếu cho phương trình (1.1) các giá trị ±1, thì biên độ sẽ là 1 / ||w|| (trong đó 1 / ||w|| là độ dài của vector *w*) còn được gọi là chuẩn, được tính là $\sqrt{\langle w, w\rangle}$.

- *SVM biên cứng*

SVM biên cứng được áp dụng đối với dữ liệu khả tách tuyến tính và nó cho kết quả phân lớp một cách chính xác với tất cả dữ liệu như hình 2.4. Để tính toán *w* và *b* tương ứng với các biên cực đại, ta phải giải quyết bài toán tối ưu sau:

$$\underset{w,b}{minimize}\ \frac{1}{2}||w^2|| \tag{2.10}$$

với ràng buộc $y_i(\langle w, x_i\rangle + b) \geq 1,\ i = 1, \dots, n$

Các ràng buộc là để đảm bảo sự phân lớp chính xác, và cực tiểu $||w||^2$, tương đương với biên cực đại. Đây là bài toán tối ưu bậc hai, trong đó nghiệm tối ưu *(w, b)* thỏa mãn các ràng buộc $y_i(\langle w, x_i\rangle + b) \geq 1$, với *w* càng nhỏ càng tốt. Bài toán tối ưu hóa này có thể giải ằng cách sử dụng các công ục tiêu chuẩn từ tối ưu hóa lồi.

- *SVM biên mềm*

Trong thực tế, dữ liệu thường không phân chia tuyến tính (hình 3). Kết quả lý thuyết và thực nghiệm cho thấy với biên lớn hơn thì SVM biên mềm sẽ cho hiệu quả tốt hơn so với SVM biên cứng. Để chấp nhận một số lỗi, người ta thay thế các ràng buộc dạng bất đẳng thức (1.2) với $y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i, i = 1, \dots, n$ trong đó $\xi_i \geq 0$ là các biến phụ không âm. $C\sum_{i=1}^{n} \xi_i$ được thêm vào hàng tối ưu hóa:

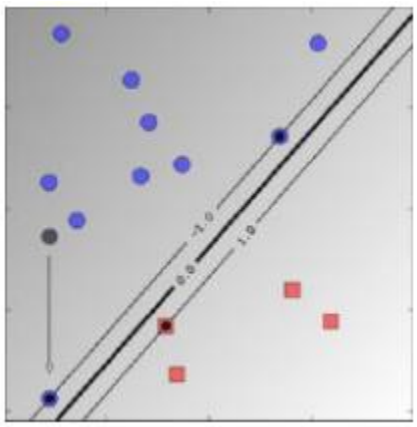
$$\underset{w,b}{minimize}\ \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \xi_i$$

với ràng buộc :

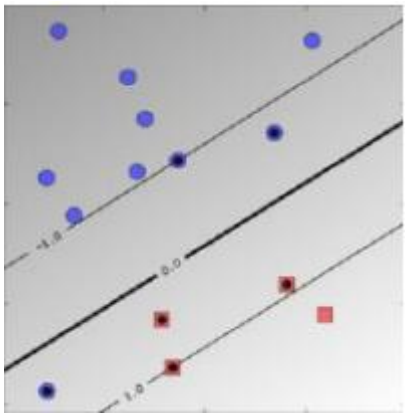
$$y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i, \xi_i \geq 0 \tag{2.11}$$

Hằng số *C >0* thiết lập mức độ quan trọng của việc cực đại biên và giảm số lượng biến phụ ξ_i. Công thức này được gọi là SVM biên mềm.

Ảnh hưởng của sự lựa chọn *C* được minh họa trong hình 3 dưới đây. Với một giá trị *C* lớn (hình 2.5a), hai điểm gần siêu phẳng nhất bị ảnh hưởng lớn hơn các điểm dữ liệu khác. Khi *C* giảm (hình 2.5b), những điểm chuyển động bên trong lề, và hướng của siêu phẳng được thay đổi, dẫn đến một biên lớn hơn cho dữ liệu. Có một lưu ý là giá trị của *C* không ý nghĩa trực tiếp, và có một công thức của SVM trong đó sử dụng một tham số trực quan hơn 0 <1 ≤ 1. Tham số v kiểm soát các vector hỗ trợ, và lỗi biên.



Hình 2.5a



Hình 2.5b

Hình 2.5: Ảnh hưởng của hằng số biên mềm C trên ranh giới quyết định.

Dữ liệu có thể được thay đổi bằng cách di chuyển điểm bóng mờ màu xám đến một vị trí mới theo mũi tên, điều đó làm giảm biên đáng kể mà một SVM biên cứng khó có thể phân tách dữ liệu. Hình 2.5a, biên quyết định cho một SVM với một giá trị rất cao của C mà bắt chước hành vi của SVM biên cứng và do đó dẫn tới lỗi huấn luyện. Một giá trị C nhỏ hơn (bên phải) cho phéo bỏ qua điểm gần ranh giới, và làm tăng biên. Ranh giới quyết định giữa các điểm dương và điểm âm được thể hiện bằng dòng đậm. Các dòng nhạt hơn là biên độ (giá trị bằng -1 hoặc +1) [10].

b, SVM cho phân lớp phi tuyến

Trong nhiều ứng dụng, một bộ phân lớp phi tuyến có độ chính xác cao hơn. Có một cách đơn giản để chuyển phân lớp tuyến tính sang phi tuyến hoặc sử dụng cho phân lớp dữ liệu không biểu diễn dưới dạng vector. Đó là ánh xạ dữ liệu cho một không gian vector nào đó mà sẽ được đề cập đến như là không gian đặc trưng, bằng cách sử dụng hàm ϕ. Hàm đó là:

$$f(x) = \langle w, \phi(x) \rangle + b \quad (2.12)$$

Lưu ý rằng *f(x)*là tuyến tính trong không gian đặc trưng được định nghĩa bởi ánh xạ ϕ, nhưng khi nhìn trong không gian đầu vào ban đầu nó là một hàm số phi tuyến *x* nếu $\phi(x)$ là một hàm phi tuyến. Ví dụ đơn giản nhất của ánh xạ là xem xét tất các các tích của các cặp (liên quan đến kernel đa thức). Kết quả là một bộ phân loại có dạng hàm phân tách bậc hai. Cách tiếp cận tính toán trực tiếp các đặc trưng phi tuyến này khỏ mở rộng cho số lượng đầu vào lớn.

Các vector trọng số của một mặt phẳng phân tách với biên độ lớn có thể được biểu diễn như một tổ hợp tuyến tính của các điểm huấn luyện, tức là $w = \sum_{i=1}^{n} y_i \alpha_i x_i$. Điều này cũng đúng cho một lớp lớn của các giải thuật tuyến tính. Hàm phân tách trở thành:

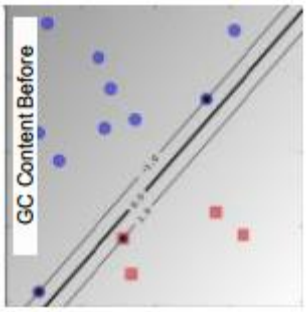
$$f(x) = \sum_{i=1}^{n} y_i \alpha_i \langle \emptyset(x_i), \emptyset(x) \rangle + b \quad (2.13)$$

Việc biểu diễn dưới dạng α_i được gọi là dạng đối ngẫu (dual), đại diện hai hàm đặc biệt phụ thuộc vào các dữ liệu chỉ thông qua các tích vô hướng trong không gian. Các quan sát tương tự cũng đúng cho bài toán tối ưu hóa đối ngẫu (phương trình (2.12)) khi thay thế x_i với $\emptyset(x_i)$.

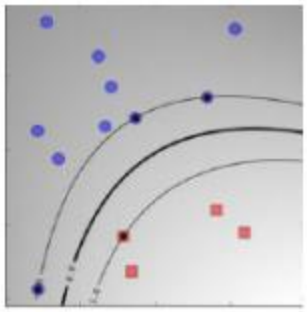
Nếu hàm kernel *k(x, x')* được định nghĩa là :

$$k(x, x') = \langle \emptyset(x), \emptyset(x') \rangle \quad (2.14)$$

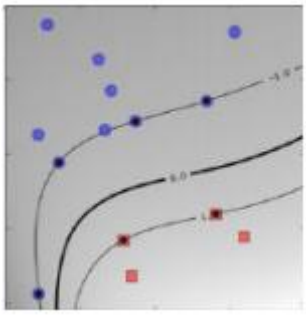
Hàm này có thể được tính toán một cách hiệu quả. Dạng đối ngẫu cho phép giải quyết vấn đề mà không cần thực hiện ánh xạ ϕ vào một không gian có nhiều chiều. Các vấn đề tiếp theo là xác định độ đo tương tự (hàm kernel) có thể được tính một cách hiệu quả.



Hình 2.6a: kernel tuyến tính



Hình 2.6b: Kernel đa thức có d=2



Hình 2.6c: Kernel đa thức có d=5

Hình 2.6: Mức độ tác động của kernel đa thức.
Kernel đa thức dẫn đến một sự phân tách tuyến tính (2.6a). Kernel đa thức cho phép một ranh giới quyết định linh họa hơn (2.6b-c) [10].

- Hàm kernel

Hai hàm kernel phổ biến nhất được sử dụng cho các dữ liệu thực là đa thức kernel và Gaussian kernel. Bậc d của đa thức kernel được định nghĩa là :

$$k_{d,k}^{polynomial}(x, x') = (\langle x, x' \rangle + k)^d \quad (2.15)$$

k thường được chọn là 0 (đồng nhất) hoặc 1 (không đồng nhất). Không gian đặc trưng cho các hàm kernel không đồng nhất bao gồm tất cả các đơn thức bậc nhỏ hơn d. Nhưng thời gian tính toán của nó là tuyến tính với số chiều của không gian đầu vào. Kernel với *d=1* và *k = 0*, biểu hiện bằng k^{linear}, là kernel tuyến tính dẫn đến một hàm phân tách tuyến tính.

Bậc của kernel đa thức kiểm soát sự linh hoạt của bộ phân lớp (hình 2.6). Đa thức bậc thấp nhất là kernel tuyến tính. Hàm kernel này không đủ tốt nếu không gian đặc trưng là phi tuyến. Đối với các dữ liệu trong hình 2.6 ở đa thức bậc 2 đã đủ linh hoạt để phân biệt giữa hai lớp với một biên tót. Đa thức bậc 5 định lượng một ranh giới quyế định tương tự, với độ cong lớn hơn. Quá trình chuẩn hóa có thể giúp cải thiện hiệu suất và ổn định *d* (đơn thức có bậc là d).

Kernel thứ hai được sử dụng rộng rãi là Gaussian kernel được xác định bởi công thức:

$$k_{\sigma}^{Gaussian}(x, x') = \exp(-\frac{1}{\sigma}||x - x'||^2) \quad (2.16)$$

Trong đó, $\sigma > 0$ là một tham số điều khiển độ rộng của Gaussian, Nó đóng một vai trò tương tự như bậc của kernel đa thức trong việc kiểm soát sự linh hoạt của bộ phân lớp. Gaussian kernel cơ bản là bằng không nếu khoảng cách bình phương $||x- x'||^2$ là

lớn hơn nhiều so với σ, tức là cho x' cố định là một vùng xung quanh x' với các giá trị kernel cao.

Như vậy, việc sử dụng một kernel phi tuyến, hoặc Gaussian hoặc đa thức, dẫn đến một cải tiến nhỏ trong việc thực hiện phân lớp kernel tuyến tính. Đối với đa thức bậc cao và Gaussian kernel nhỏ, độ chính xác được thu giảm.

2.2. Các nghiên cứu liên quan

Dựa trên các phương pháp đã trình bày ở trên, đã có nhiều đề tài nghiên cứu xoay quanh bài toán phân loại (phân lớp) quan điểm được công bố tiêu biểu như sau:

Hu và Liu đã có nhiều nghiên cứu trong lĩnh vực khai phá quan điểm, các nghiên cứu của họ tập trung vào các từ mang quan điểm. Theo [4] họ đưa ra phương pháp xác định quan điểm thông qua việc hướng ngữ nghĩa của từ mang quan điểm. Việc xác định quan điểm ở mức câu đã cho thấy tính hiệu quả trong phương pháp của họ, họ thực hiện xác định từ mang quan điểm và thống kê dựa trên hướng quan điểm của chúng. Ngoài vấn đề phân lớp quan điểm thì Hu và Liu còn quan tâm đên đặc trưng của sản phẩm và những quan điểm được bày tỏ với chúng. Điều nay chứng tỏ độ phức tạp của phân lớp quan điểm trong nghiên cứu của họ được tăng thêm. Kết quả trong nghiên cứu này độ chính xác (precision) đạt 64,2%, độ bao phủ (recall) đạt 69,3% nhưng độ chính xác hướng câu (sentences oriented accuracy) lại đạt 82,4% và nghiên cứu này được đánh giá là nghiên cứu độ chính xác cao nhất hiện nay.

Cùng sử dụng phương pháp học máy, Bo Pang và Lilian Lee [5] thực hiện khảo sát một vài thuật toán học máy có giám sát cho phân lớp quan điểm các đánh giá về phim. Họ kết luận các thuật toán học máy này hoạt động tốt và đem lại kết quả cao hơn so với các phương pháp có sử dụng dữ liệu gán nhãn thủ công. Trong đó thuật toán SVM đạt kết quả cao nhất với độ chính xác giao động trong khoảng 78,2% đến 82,9%.

Trong khi đó, Turney [6] lại áp dụng thuật toán học không giám sát dựa trên các thông tin qua lại giữa hai đoạn văn bản với các từ tích cực và tiêu cực, các thông tin qua lại này sẽ được xác định bởi một cỗ máy tìm kiếm. Độ chính xác trong phương pháp của Turney đạt 74%.

Thông qua các nghiên cứu trên có thể thấy rằng, mỗi phương pháp nghiên cứu đều có những ưu điểm riêng và phù hợp trong những trường hợp cụ thể khác nhau. Nhờ vào việc phân tích đối tượng nghiên cứu để lựa chọn ra phương pháp thích hợp cho bài toán nghiên cứu.

2.3. Kết luận

Dựa vào việc tìm hiểu về các phương pháp cũng như các đề tài nghiên cứu khoa học đã trình bày ở trên, do đặc thù bộ dữ liệu thu thập được tương đối nhiều cùng với việc xử lý ngôn ngữ tự nhiên sẽ gặp nhiều khó khăn trong việc gán nhãn lớp nên phương pháp được đề xuất để giải quyết bài toán phân loại quan điểm người dùng trong đồ án này là phương pháp học máy SVM (Support Vector Machine)

Vấn đề được đặt ra là một kết quả phân loại tốt phụ thuộc vào những yếu tố gì? Yếu tố nào là quan trọng nhất?

Dựa vào các phương pháo nghiên cứu đã trình bày và một số nghiên cứu liên quan, có thể thấy một kết quả phân loại tốt sẽ được sinh ra từ mô hình phân lớp mà có khả năng phân tách (phân lớp) được nhiều nhất số các mẫu dữ liệu thành các lớp riêng biệt (trong bài toán đồ án đề cập đến là 2 lớp). Kết quả này sẽ phụ thuộc vào 3 yếu tốt quan trọng hàng đầu đó là:

- Tập dữ liệu huấn luyện đủ lớn và chính xác. Trong quá trình huấn luyện, khi có tập dữ liệu đủ lớn và chính xác thì quá trình huấn luyện sẽ tốt, đồng nghĩa với việc sẽ có một mô hình phân lớp tốt phục vụ cho giai đoạn phân lớp sau này. Nếu như tập dữ liệu quá ít thì sẽ gây ra tình trạng có nhiều đặc trưng chưa được "học" trong quá trình huấn luyện. Và nếu tập dữ liệu không chính xác ngay từ đầu thì mô hình học dù có sử dụng đúng phương pháp thì mô hình phân lớp cũng không thể tốt, dẫn đến hệ quả là quá trình phân lớp sau này sẽ sai theo.
- Quá trình xử lý dữ liệu cũng là một bước quan trọng không kém. Có thể dễ dàng nhận thấy một điều là các phương pháp ở trên hầu hết đều sử dụng mô hình vector để biểu diễn văn bản. Do đó, việc lựa chọn phương pháp tách từ sẽ ảnh hưởng rất lớn đến quá trình biểu diễn văn bản bằng vector. Đối với một số ngôn ngữ như tiếng Anh, việc tách từ sẽ dễ dàng hơn vì các từ tách biệt nhau chỉ bởi dấu cách (khoảng trắng). Tuy nhiên, đối với một số đa âm như tiếng Việt thì việc tách từ dựa trên khoảng trắng là không chính xác, hơn nữa trong tiếng Việt có rất nhiều tính từ ở dạng từ ghép, loại từ này chính là từ mang quan điểm nhiều nhất. Vậy nên việc lựa chọn phương pháp tách từ là một yếu tố quan trọng quyết định đến độ chính xác cảu tập dữ liệu huấn luyện.
- Một thuật toán để sử dụng trong quá trình phân lớp (phân loại) sẽ là hợp lý khi thuật toán phân loại có thời gian xử lý hợp lý. Tức là thời gian học, thời gian phân loại văn bản là hợp lý. Bên cạnh đó, thuật toán cần phải có tính tăng cường (incremental function), nghĩa là chỉ phân loại các văn bản mới mà không phân loại lại toàn bộ văn bản khi thêm một số văn bản mới vào tập dữ liệu. Khi đó, thuật toán có khả năng giảm độ nhiễu (noise) khi phân loại văn bản.

Với tập dữ liệu đầu vào của đề tài nghiên cứu là các nhận xét cá nhân mang quan điểm bằng ngôn ngữ tiếng Việt, việc xử lý sẽ gặp nhiều khó khăn, hơn nữa các nhận xét cá nhận lại có độ dài không ổn đinh (quá ngắn hoặc quá dài). Trên thực tế, SVM là phương pháp khi dùng trong việc phân loại văn bản (tiếng Anh) đem lại độ chính xác phân loại cao và trội hơn các phương pháp khác. Dựa trên hai cơ sở này, đồ án sẽ áp dụng SVM vào giải quyết bài toán phân loại quan điểm người dùng (tiếng Việt) đối với các sản phẩm công nghệ.

CHƯƠNG 3: XÂY DỰNG MÔ HÌNH PHÂN LOẠI QUAN ĐIỂM NGƯỜI DÙNG VỀ SẢN PHẨM CÔNG NGHỆ

Xuyên suốt nội dung mà chương 1 và chương 2 đã trình bày, có thể thấy mục đích đồ án là giải quyết bài toán phân loại quan điểm người dùng về sản phẩm công nghệ sử dụng phương pháp Support Vector Machine. Quá trình xây dựng mô hình phân loại quan điểm một chi tiết sẽ được trình bày trong chương 3 dưới đây.

3.1. Yêu cầu dữ liệu

- *Input*: Bộ dữ liệu lưu nội dung ý kiến đánh giá của người dùng về một sản phẩm công nghệ cụ thể dưới dạng văn bản text và ngông ngữ tiếng Việt
- *Output*: Phân loại đánh giá tổng quát về sản phẩm công nghệ.
  Độ chính xác của hệ thống.

3.2. Thu thập và tiền xử lý dữ liệu

3.2.1. Google Custom Search

Google Custom Search: cho phép người dùng tạo ra một công cụ tìm kiếm cho trang web của mình, blog, hoặc một bộ sưu tập của các trang web. Người dùng có thể tùy chỉnh các công cụ tìm kiếm để tìm kiếm cả các trang web và hình ảnh. Bên cạnh đó, có thể chỉnh các bảng xếp hạng, tùy biến giao diện và cảm nhận của kết quả tìm kiếm, và mời bạn bè hoặc người sử dụng tin cậy để giúp xây dựng công cụ tìm kiếm tùy chỉnh của chính người dùng đó.

- *Đặc điểm chính*

Có hai trường hợp sử dụng chính cho Custom Search - tạo ra một công cụ tìm kiếm chỉ tìm kiếm các nội dung của một trang web (trang web tìm kiếm), hoặc tập trung vào một chủ đề cụ thể từ nhiều trang web. Người dùng có thể sử dụng chuyên môn của mình về một chủ đề mà các trang web để tìm kiếm, ưu tiên, hoặc bỏ qua. Vì với Custom search, người dùng có thể:

- Tạo công cụ tìm kiếm tùy chỉnh mà tìm kiếm trên một bộ theo quy định của các trang web hoặc các site.
- Kích hoạt tính năng tìm kiếm hình ảnh cho trang web của người dùng.
- Tùy chỉnh giao diện của kết quả tìm kiếm.
- Tận dụng cấu trúc dữ liệu trên trang web của người dùng để tùy chỉnh tìm kiếm.
- Có hai phiên bản Google Custom Search
  Custom Search Engine (phiên bản basic)
  Google Site Search (phiên bản business)

- *Custom Search APIs*

Sau khi tạo một CustomSearch Engine, người dùng có thể sử dụng một trang mặc định lưu trữ bởi Google để hiển thị kết quả, hoặc có thể nhúng chức năng tìm kiếm trực tiếp trong trang web của người dùng.

- Đăng kí tại: *https://www.google.com.vn/cse/all*

3.2.2. Thu thập dữ liệu tự động

Như trong chương 1 đã trình bày, việc thu thập dữ liệu sao cho nguồn dữ liệu đáng tin cậy và đem lại hiệu quả là bước cần thiết quan trọng đầu tiên trong việc phân loại ý kiến đánh giá của người dùng. Dưới đây là một số cách tiếp cận thu thập dữ liệu:

- *Thu thập dữ liệu thủ công bằng tay:* thông qua việc, người dùng sẽ nhập từ khóa vào các công cụ tìm kiếm như google…, sau đó các công cụ này sẽ trả về kết quả là các link Url, người dùng phải mở lần lượt từng link và xem dưới dạng html rồi lưu thành văn bản. Tuy nhiên có thể nhận thấy rằng cách này không phù hợp với dữ liệu là nguồn thông tin lớn.
- *Thu thập dữ liệu bán tự động*: người dùng search tương tự trên các công cụ tìm kiếm để lấy được các link Url trả về, sau đó copy những đường dẫn này vào 1 arrayList, sau đó viết hàm phân tích đọc nội dung các link trong ArrayList này rồi lưu kết quả vào cơ sở dữ liệu.

Cách này có thể giúp quản lí khoa học các file html và văn bản thu được so với cách 1 nhưng chưa phải là tối ưu nhất?

- Để giải quyết câu hỏi từ cách 1 và cách 2 với lượng thông tin lớn, và việc quản lí dữ liệu thu được một cách khoa học và chính xác, việc thu thập dữ liệu sẽ dựa phương pháp *thu thập dữ liệu tự động* bằng cách:

- Tạo 1 service giải quyết bài toán thu thập dữ liệu rồi đẩy lên server.
- Bên trong service tạo ra các thread, mỗi thread tìm kiếm dựa trên một nguồn mà mình mặc định sẵn dựa trên công cụ Google Custom Search. Các thread này sau khi nhận được các link url trả về sau khi search, sẽ làm nhiệm vụ tự động phân tích từ URL thông qua các phương thức và bộ thư viện JSOUP để lấy ra nội dung cần thiết, bên cạnh đó, lọc bỏ các tag html để tạo thành một văn bản thuần, và cuối cùng, tự động lưu vào CSDL.

Với phương pháp trên, việc thu thập dữ liệu đã trở nên đơn giản hơn, tự động và rút ngắn thời gian cũng như đem lại kết quả chính xác với mỗi từ khóa mà người dùng tìm kiếm. Phương pháp thu thập dữ liệu tự động này thường áp dụng trong phạm vi các trang web có cung cấp search engine, những web dạng chuẩn html.

Dựa trên mục tiêu cũng như yêu cầu và phương pháp giải quyết vấn đề phân loại quan điểm người dùng về sản phẩm công nghệ đã được đề cập ở chương 1 và chương 2, nhận thấy rằng *thu thập dữ liệu tự độn*g là giải pháp phù hợp nhất với bài toán mà đề tài đang nghiên cứu.

	Result_id	Url	Content
		Click to select all grid cells tinhte.vn/categories/android.150/	Android | Tinhte.vn Android Android Android Khu vực dành cho các Thiết bị (Điện thoại, M
2	128	https://www.tinhte.vn/forums/android-tro-choi.467/	[Android] Trò chơi | Tinhte.vn Android Android Android [Android] Trò chơi Cài Xmoc
3	129	https://www.tinhte.vn/forums/android-phan-mem.216/	[Android] Phần mềm | Tinhte.vn Android Android Android [Android] Phần mềm Giới thiệu và thảo lu
4	130	https://www.tinhte.vn/forums/android-tin-tuc-danh-gia.151/	[Android] Tin tức - Đánh giá | Tinhte.vn Android Android Android [Android] Tin tức - Đ&aacu
5	131	https://www.tinhte.vn/forums/android-nang-cap-firmware.280/	[Android] Nâng cấp Firmware | Tinhte.vn Android Android Android [Android] Nâng cấp Firmware
6	132	https://www.tinhte.vn/forums/android-thu-thuat.290/	[Android] Thủ thuật | Tinhte.vn Android Android Android [Android] Hỏi đáp - CSKN [Android] Thủ t
7	133	https://www.tinhte.vn/tags/%E1%BB%A9ng+d%E1%BB%A5ng+android/	ứng dụng android | Tinhte.vn Android Android Tìm kiếm chủ đề đã tag với ứng dụng androi
8	134	https://www.tinhte.vn/forums/android-hoi-dap-cskn.217/	[Android] Hỏi đáp - CSKN | Tinhte.vn Android Android Android [Android] Hỏi đáp - CSKN G
9	135	https://www.tinhte.vn/tags/android+app/	android app | Tinhte.vn Android Android Tìm kiếm chủ đề đã tag với android app android ap
10	136	https://www.tinhte.vn/tags/android+l/	android l | Tinhte.vn Android Android Tìm kiếm chủ đề đã tag với android l android l Trang t
11	137	https://www.tinhte.vn/threads/football-manager-handheld-2014-v5-0-4-game-quan...	Android Android Android [Android] Trò chơi [Android] Trò chơi Không phải l&a
12	138	https://www.tinhte.vn/threads/football-manager-handheld-2012-game-android-qua...	QUẢN LÝ - Football Manager Handheld 2012: game android quản lý đội bóng cực
13	139	https://www.tinhte.vn/threads/quan-ly-bong-da-football-manager-handheld-2013-v...	Android Android Android [Android] Trò chơi [Android] Trò chơi 14 quốc gia với giải đấu c&c
14	140	https://www.tinhte.vn/threads/football-manager-handheld-2013-v4-0-game-quan-l...	Android Android Android [Android] Trò chơi [Android] Trò chơi Yêu cầu phiên b
15	141	https://www.tinhte.vn/threads/android-lan-san-sang-may-choi-game-handheld.164...	ANDROID lần sân sang máy chơi game Handheld! | Tinhte.vn Android Android Android [An
16	142	https://www.tinhte.vn/threads/game-football-manager-handheld-2014-va-cach-ha...	Android Android mấy bác ios chơi dc r ko biết android chơi đc chưa @@

Hình 3.1: Kết quả thu được sau bước thu thập dữ liệu tự động

3.2.3. Tiền xử lý dữ liệu

Tuy nhiên, có thể nhận thấy tại dữ liệu mà mỗi link URL trả về đươc lưu trong cơ sở dữ liệu vẫn chứa một lượng lớn thông tin mà chắc chắn không mang quan điểm như các tag html * ,
, <p>,...* các từ ngữ không mang quan điểm như *"nếu", "thì", "là", "mà"...*, các kí tự đặc biệt như *"," / ";" / "!" / ">" / "\" / "?" "enter" / "..."*....Nếu như dữ liệu gồm tất cả những kí tự dư thừa và từ ngữ không mang quan điểm trên thì lượng dữ liệu mà hệ thống phải quản lí là rất lớn. Điều này gây giảm hiệu năng, tốc độ xử lý của hệ thống cũng như ảnh hưởng xấu đến kết quả phân loại. Do đó, việc làm sạch dữ liệu là điều hết sức quan trọng trong giai đoạn tiền xử lí dữ liệu.

Input: tập dữ liệu chưa được làm sạch lưu trữ trong CSDL dưới định dạng file .txt

Output: mỗi file .txt được làm sạch tuyệt đối (chỉ chứa các từ ngữ mang quan điểm và không chứa dấu câu).

Đối với dữ liệu thu được của hệ thống phân loại quan điểm trong đồ án này, tiền xử lí dữ liệu được thực hiện qua các bước sau:

- *Loại bỏ tag html:*

Sử dụng thư viện Jsoup với tính năng Lấy văn bản từ URL (Load Document from Url) và loại bỏ các tagHtml bằng cách replace các kí tự đặc biệt bằng kí tự trống.

- *Tách dữ liệu thành câu*:

Xử lý với đơn vị nhỏ nhất là 1 file .txt. Trong file này, sau khi đã loại bỏ tag html, dữ liệu ở dưới dạng văn bản theo cấu trúc đoạn - từ - câu.

Hệ thống sẽ được xây dựng một bộ tách dữ liệu, khi gặp bất kì dấu câu như tổ hợp dấu ". " và các kí tự đặc biệt như "," / ";" / "!" / ">" / "\" / "?" "enter" / "…" thì tự động sẽ tách đoạn các từ ngữ nằm giữa các dấu câu này thành 1 đoạn.

- *Tách dữ liệu thành từ*

Tuy nhiên, dữ liệu thu được sau bước trên chưa phải là tối ưu nhất vì vẫn chưa những từ ngữ không mang định hướng quan điểm như "có", "với", "cái", "tuy nhiên", "còn", "như", "bằng"…

Nên phương pháp để loại bỏ các từ ngữ này là sử dụng bộ đặc trưng đã được xây dựng, so sánh dữ liệu với bộ các từ mang đặc trưng này, chỉ những từ nào so khớp thì được giữ lại, còn tất cả những từ khác sẽ được loại bỏ. Khi đó, 1 file .txt được rút gọn tối đa mà vẫn thể hiện đầy đủ quan điểm đánh giá của người dùng. Về bộ đặc trưng mà đồ án sử dụng sẽ được trình bày ở mục 3.3.

- *Chuyển đổi dữ liệu dạng text sang vector:*

Tại bước này, hệ thống sẽ kết hợp dữ liệu sau khi tách từ ở dạng text và các từ đặc trưng đã được xây dựng để tạo thành các vector đúng định dạng chuẩn của công cụ SVM light (công cụ SVM [light] sẽ được trình bày ở mục 3.4).

```
-1 6:0.625 19:0.5 39:0.5 71:0.5 108:0.5 167:0.75 255:0.5 282:0.75 310:0.75
-1 2:0.375 58:0.75 120:0.625 166:0.875 196:0.875 209:0.625 257:0.75 326:0.
-1 39:0.5 71:0.5 108:0.5 234:0.5 282:0.75 477:0.75 484:0.5 559:0.5 569:0.6
-1 41:0.625 162:0.5 255:0.5 257:0.75 433:0.636 435:0.556 484:0.5 569:0.675
-1 39:0.5 71:0.5 108:0.5 234:0.5 278:0.5 282:0.75 477:0.75 484:0.5 559:0.5
```

Hình 3.2: Định dạng chuẩn cấu trúc vector của SVM[light]

	Nội dung dữ liệu
Dữ liệu ban đầu	[Androi]| [Android] <p>@cmt: Pin chạy tương đối bền. Ngoài ra, điện thoại còn nhiều chức năng khác như camera sau xoay 180 độ, hai sim hai sóng, mở khóa bằng nhận diện khuôn mặt…Tuy nhiên, màn hình có với độ phân giải thấp so với cái giá quá đắt! :p
Loại bỏ các kí tự đặc biệt và các thẻ tag html	Pin chạy tương đối bền. Ngoài ra, điện thoại còn nhiều chức năng khác như camera sau xoay 180 độ, hai sim hai sóng, mở khóa bằng nhận diện khuôn mặt…Tuy nhiên, màn hình có với độ phân giải thấp so với cái giá quá đắt
Tách dữ liệu thành câu	Pin chạy tương đối bền Ngoài ra điện thoại còn nhiều chức năng khác như camera sau xoay 180 độ hai sim hai sóng mở khóa bằng nhận diện khuôn mặt Tuy nhiên màn hình có độ phân giải thấp so với cái giá quá đắt
Tách dữ liệu thành từ	tương đối bền nhiều chức năng thấp đắt

Bảng 3.1: Kết quả sau các bước xử lý dữ liệu

3.3. Xây dựng bộ từ đặc trưng

Dựa theo bảng 3.1 đã trình bày ở trên có thể thấy trong một đoạn text thể hiện quan điểm vẫn có những từ ngữ không hoặc thể hiện rất ít quan điểm người dùng, thậm chí, trong một câu dài, số lượng từ ngữ mang quan điểm rất ít. Nếu dữ liệu dùng để phân lớp chứa cả những từ dưa thừa này sẽ làm việc phân lớp bị chậm vì có nhiều từ cần phải xử lý, đồng thời làm giảm độ chính xác của thuật toán phân lớp. Do vậy việc xây dựng bộ các từ ngữ mang quan điểm đánh giá về sản phẩm công nghệ sẽ quyết định đến chất lượng của bộ huấn luyện cũng như việc xây dựng mô hình phân lớp sau này. Hơn nữa, việc phân loại quan điểm ở đồ án này là đánh giá ý kiến người dùng bằng ngôn ngữ tự nhiên tiếng Việt. Vì đơn vị nhỏ nhất của tiếng Việt là tiếng, trong khi để đánh giá quan điểm phải đánh giá từ ngữ mang nghĩa và không phải từ ngữ nào cũng có một tiếng. Nói cách khác, một tiếng có thể có nghĩa nhưng cũng có

thể không đem lại nghĩa gì. Bên cạnh đó, tiếng Việt lại có rất nhiều từ ghép, đặc biệt là tính từ - là loại từ sẽ thể hiện quan điểm rõ ràng nhất, sau đó đến động từ và một số danh từ. Như vậy, điều này sẽ gây khó khăn trong quá trình xác định từ mang quan điểm cũng như việc đánh giá ý kiến người dùng sẽ mang nghĩa tích cực hay tiêu cực nếu bước tách từ không thực hiện tốt.

3.3.1. Giới thiệu VietSentiWordNet

Đồ án xây dựng bộ các từ đặc trưng thể hiện quan điểm dựa trên bộ từ điển tiếng dành cho tiếngViệt – VietSentiWordNet mở rộng của Lưu Công Tố [7] dựa trên miền dữ liệu tin tức của nhóm tác giả Vũ Xuân Sơn và cộng sự [8].

Trong bộ từ điển VietSentiWrodNet, mỗi từ ngữ mang quan điểm (có dạng tính từ, động từ và động từ) đều có các thuộc tính:

- *POS*: từ loại của từ
- *ID*: mã đại diện của từ
- *PosScore*: trong số tích cực của từ
- *NegScore*: trọng số tiêu cực của từ
- *SynsetTerms*: là từ được nhắc đến với một ID duy nhất
- *Gloss:* chú thích, dịch nghĩa của từ

Ví dụ: từ "*vui vẻ*" trong VietSentiWrodNet có Pos là "*a*" kí hiệu của từ "*adj*" tức là tính từ, ID = "*00362467*", PosScore = "*0.75*", NegScore = "*0*", SynsetTerms là "*vui_vẻ*", Gloss là "*một tinh thần tốt, thể hiện tâm trạng rất vui; "tính vui vẻ của cô ấy"; "một lời chào vui vẻ"; "một căn phòng vui vẻ""*".

3.3.2. Bộ từ đặc trưng

Đối tượng của đồ án hướng tới là các sản phẩm công nghệ nên các từ ngữ mang quan điểm cũng phải chọn lọc sao cho phù hợp với đối tượng mà đồ án nghiên cứu. Bộ đặc trưng riêng được xây dựng lại dựa vào bộ từ điển VietSentiWordNet, với số lượng từ mang quan điểm ít hơn, cấu trúc cũng được rút gọn cho phù hợp hơn với bài toán phân loại quan điểm người dùng.

Mỗi từ/ cụm từ được chọn đều là từ ngữ mang quan điểm hướng về việc đánh giá các sản phẩm công nghệ như "bền", "đẹp", "giá quá cao", "chính hãng"… Mỗi từ/ cụm từ này đều có các thuộc tính: ID – mã của từ, PosScore – chỉ số tích cực, NegScore – chỉ số tiêu cực, SynsetTerms - từ được nhắc đến.

Khi thực hiện bước tách từ, bộ dữ liệu sau khi tách câu sẽ được so sánh với bộ đặc trưng trên và chỉ giữ lại những từ nào có mặt trong bộ đặc trưng, chính là giữ lại những từ ngữ mang quan điểm đánh giá. Sau bước tách từ, dữ liệu mà hệ thống thu thập đã được rút rất nhiều cả về số lượng lẫn chất lượng.

3.4. Phân loại quan điểm

Phương pháp được lựa chọn để giải quyết bài toán phân loại quan điểm người dùng về sản phẩm công nghệ trong đồ án này là phương pháp máy vector hỗ trợ

(Support Vector Machine - SVM). Một trong những công cụ được xây dựng dựa trên thuật toán SVM là *SVM light*, công cụ hỗ trợ này sẽ được sử dụng để phân lớp quan điểm mà đồ án đã đề cập.

Bộ công cụ phân lớp svm-light viết trên C được phát triển bởi Joachims Thorste [10] với các đặc điểm chính sau:

a, Tính năng chính:

- Tối ưu hóa thuật toán
- Giải quyết nhanh các vấn đề phân loại và hồi quy đối với các kết quả đầu ra đa biến.
- Hỗ trợ các phương pháp nhận dạng mẫu

b, Các thành phần chính:

- SVMTlearn : huấn luyện mô hình
- SVMTagger : gán nhãn cho phân lớp
- SVMTeval : đánh giá kết quả
- SVMClassicfy: kiểm thử kết quả

c,Cài đặt:

- Download từ nguồn:

http://download.joachims.org/svm_light/current/svm_light.tar.gz

- Sử dụng command line để cài đặt và sử dụng svm light với *svm-learn* và *svm-classify*.

d, Sử dụng:

Dữ liệu cần chuẩn bị:

Bộ từ điển Vietsentiwordnet, bộ từ đặc trưng.

Bộ trainning đã được gán nhãn.

Bộ test riêng biệt chưa gán nhãn.

Quá trình huấn luyện

Câu lệnh: svm-learn [-option] train_file model_file

Trong đó:

train_file: bao gồm dữ liệu huấn luyện. Tên file của train_file có thể là bất kỳ. Phần mở rộng của file phải do người sử dụng đặt nhưng phải giới hạn với 3 ký tự.

model_file: chứa model được xây dựng dựa trên dữ liệu huấn luyện của SVM.

- Những *[-option]* được sử dụng khi sử dụng câu lệnh svm_learn:

Cấu hình chung:

- - *?*: Hiển thị tất cả các mục, giúp người sử dụng có thể tìm hiểu và sử dụng công cụ SVM^{light} một cách thông thạo và có hiệu quả.
- *v[0...3]*: Việc học chi tiết của công cụ với các vector có 4 mức bắt đầu từ mức 0 cho tới mức 3. Mặc định không chọn gì công cụ sẽ hiểu là ở mức 1.

Các mục học:

- *z {c,r,p}*: Việc học vector sẽ chọn lọc giữa việc phân loại (c), hồi quy (r) và việc ưu tiên xếp hạng vector theo (p).

- *c float :* trao đổi giữa việc học vector và biên độ lỗi. Mặc định: $[avg.x^2]^{-1}$.Với *avg* là biên độ lỗi trung bình của các vector.
- *w[o..]* : chiều rộng của Ɛ cho việc hồi quy. Mặc định: 0.1
- *j float* : hệ số cố định cho việc học đối với các vector tích cực dựa trên các vector tiêu cực. Mặc đinh: 1.
- *b[0,1]* : sử dụng các tính toán đối xứng giữa các vector là $x.w + b_0$ thay cho việc tính toán không đối xứng giữa các vector là $x.w_0$. Mặc định: 1
- *i[0,1]* : Xóa bỏ những vector học không nhất quán và tiến hành học lại. Mặc định: 0.

Các mục đánh giá hiệu năng sai số khi học:

- *x[0,1]*: So sánh những ước tính của từng vector còn lại. Mặc định : 0
- *o[0..2]*: Tính toán hàm ước lượng đối với mỗi vector. Mặc định : 1.0.
- *k[0..100]*: Mở rộng độ sâu tính toán. Mặc định : 0

Mục chuyển dữ liệu khi học

- *p[0..1]*: Phần nhỏ những vector chưa có nhãn được xếp và mặt tích cực dựa trên tỉ số tích cực và tiêu cực nằm trong bộ huấn luyện.

Các hạt nhân chính dùng để học:

- *t int* : định nghĩa các hàm dùng để học bộ dữ liệu với 0 là hàm tuyến tính (mặc định); 1 là hàm đa thực $(sa.b + c)^d$; 2 là hàm tính giá trị radial; 3 là đường sigmoid với $\tan^{-1}(sa.b + c)$; 4 là người dùng tự định nghĩa hàm để lọc bộ dữ liệu trong file kernel.h
- *d int* : là tham số đầu vào với hàm đa thức
- *g float* : tham số đầu vào γ.
- *s float* : tham số đầu vào s
- *r float* : tham số đầu vào r
- *u string* : tham số đầu vào do người sử dụng định nghĩa.

Các mục tối ưu hóa khi học

- *q [2...]*: Kích thước lớn nhất của việc phân loại. Mặc đinh: 10.
- *n [2..q]*: số các biến được đưa vào dữ liệu học.
- *m [5..]*: kích thước của pha đánh giá vector với độ đo là MB. Mặc định: 40mb. Kích thước càng lớn việc đánh giá sẽ nhanh hơn.
- *e float*: cấp phát những lỗi khi học vector theo tiêu chuẩn đầu cuối với công thức eps = [y.[w.x + b] -1]. Mặc định: 0.001.
- *h [5..]*: số lần lặp lại 1 vector cần phải được tối ưu trước khi thu hẹp. Mặc đinh: 100.
- *f [0,1]*: thực hiện kiểm tra tối ưu cuối cùng cho các biến bị loại bỏ bằng cách thu hẹp. Mặc dù thử nghiệm này thường là tích cực, không có đảm bảo rằng tối ưu đã được tìm thấy nếu kiểm tra được bỏ qua. Mặc định: 1.

- *y string*: nếu các tùy chọn được đưa ra, đọc bản α từ dữ liệu cho trước và sử dụng chúng để bắt đầu gán nhãn.
- *#int*: chấm dứt việc tối ưu hóa các vector sau số lần lặp lại. Mặc đinh: 100000.

Các đầu ra khi học:

- *l char*: file sinh ra để viết nhãn dự đoán của các ví dụ không có nhãn sau khi học.
- *a char*: viết tất cả các vector vào file sau khi học.

- Quá trình phân lớp

  Câu lệnh: svm_classify [options] example_file model_file output_file

  Trong đó:

  example_file: file chưa có nhãn, chỉ có các thuộc tính

  model_file: file được sinh ra trong quá trình huấn luyện

  output_file: giống file *example_file* nhưng đã được gán nhãn

- Những *[-option]* được sử dụng khi sử dụng câu lệnh svm_classify:

v[0...3]: Việc học chi tiết của công cụ với các vector có 4 mức bắt đầu từ mức 0 cho tới mức 3. Mặc định không chọn gì công cụ sẽ hiểu là ở mức 1.

f[0,1]: thực hiện kiểm tra tối ưu cuối cùng cho các biến bị loại bỏ bằng cách thu hẹp. Mặc dù thử nghiệm này thường là tích cực, không có đảm bảo rằng tối ưu đã được tìm thấy nếu kiểm tra được bỏ qua. Mặc định là 1.

e, Ưu và nhược điểm của SVM^{light}

- Điểm tích cực và đã làm được của SVM^{light}:
  - Tìm được khoảng cách biên lớn nhất giữa 2 miền dữa liệu. Chia dữ liệu thành 2 miền dữ liệu là miền tích cực và tiêu cực.
  - Giúp người dùng dễ dàng phân tách được các dữ liệu tích cực, tiêu cực thay vì phải làm bằng thủ công.
  - Công cụ SVM^{light} đạt độ chính xác tương đối cao.
- Điểm hạn chế của SVM^{light}:
  - Công cụ SVM^{light} còn hạn chế với các dữ liệu đầu vào có cùng kích cỡ. Với các dữ liệu đầu vào có kích cỡ vector khác nhau thì độ chính xác khi phân loại chưa cao.

Sau khi tạo dữ liệu đầu vào đúng, thực hiện quá trình phân lớp dữ liệu bằng các câu lệnh đã trình bày ở trên, công cụ SVM^{light} sẽ tiến hành phân lớp dữ liệu, mỗi vector cần phân lớp trong *example_file* sẽ được gán một giá trị mới. Dựa vào giá trị đó, ta có thể hiểu được vector đó được phân loại tích cực hay tiêu cực.

CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

4.1. Phương pháp đánh giá

4.1.1. Phương pháp kiểm tra chéo Cross Validation

Ước lượng độ chính xác của bộ phân lớp là quan trọng ở chỗ nó cho phép dự đoán được độ chính xác của kết quả phân lớp những dự liệu trong tương lai. Đồ án này sử dụng kỹ thuật *k-fold cross validation* để thực hiện việc đánh giá hệ thống phân lớp.

Trong phương pháp *k-fold cross validation,* tập dữ liệu ban đầu được chia ngẫu nhiên thành k tập con không giao nhau (gọi là fold) có kích thước xấp xỉ nhau S_1, S_2, …, S_k.

Quá trình học và kiểm tra thực hiện *k* lần. Mỗi lần (trong số k lần) lặp, một tập con được sử dụng làm tập kiểm thử và *(k-1)* tập con còn lại được dùng làm tập huấn luyện. Các lựa chọn thông thường của k là 10 hoặc 5. Tại lần lặp thứ *i,* S_i là tập dữ liệu kiểm tra, các tập còn lại hợp thành tập dữ liệu đào tạo. Độ chính xác là toàn bộ số phân lớp đúng chia cho tổng số mẫu của tập dữ liệu ban đầu.

Đối với bài toán đánh giá hệ thống phân loại quan điểm người dùng, có hai chỉ số cần quan tâm đó là: precision (độ chính xác) và recall (độ bao phủ).

Với công thức tính độ chính xác như sau:

- tP (true_positive): số lượng đánh giá tích cực được gán nhãn đúng
- fP (false_positive): số lượng đánh giá tích cực bị gán nhãn sai
- tN (true_negative): số lượng đánh giá tiêu cực được gán nhãn đúng
- fN (false_negative): số lượng đánh giá tiêu cực bị gán nhãn sai
- Precision = $\frac{tP}{(tP+fP)}$
- Recall = $\frac{tP}{(tP+fN)}$

4.1.2. Bộ thư viện hỗ trợ LibSVM

Một trong những công cụ được xây dựng dựa trên kỹ thuật *k-fold cross validation* là bộ thư viện libSVM [11].

- *Định dạng file:*

  Định dạng của file dữ liệu huấn luyện và file test là:
  <label><index1>:<value1><index2>:<value2> ... trong đó:
  Trong đó:
  <*label*>: là giá trị đích của tập huấn luyện. Đối với việc phân lớp là một số nguyên xác định một lớp.
  <*index*>: là một số nguyên bắt đầu từ 1 (trong bài toán đồ án đề cập là id của từ đặc trưng).

<value>: là một số thực. Các nhãn trong file dữ liệu test chỉ được sử dụng để tính toán độ chính xác hoặc lỗi (trong bài toán đồ án đề cập là giá trị dùng để đánh giá từ đặc trưng: PosScore hoặc NegScore).

- *Cách sử dụng:*

Trước khi phân lớp dữ liệu test, cần xây dựng một mô hình SVM ('svm_model') sử dụng dữ liệu huấn luyện. Một mô hình cũng có thể được lưu trong một file cho việc sử dụng sau này. Mỗi một mô hình SVM phải sẵn sàng, có thể dùng nó để phân lớp dữ liệu mới.

- Sử dụng *svm-train*:

Cú pháp: svm-train [options] training_set_file [model_file]

Trong đó: *model_file* là file mô hình được sinh bởi svm-train

- Sử dụng *svm-predict*:

Cú pháp: svm-predict [options] test_file model_file output_file

Trong đó: *test_file* là file dữ liệu test mà ta muốn đánh giá. svm-predict sẽ đưa ra trong file output_file.

4.2. Dữ liệu đầu vào

Mô hình được xây dựng thực hiện chức năng phân loại quan điểm người dùng về sản phẩm công nghệ với ngôn ngữ tiếng Việt.

Dữ liệu để thực nghiệm là các comment (nhận xét) của người sử dụng các sản phẩm công nghệ từ website: https://www.tinhte.vn/ và http://www.handheld.vn/

		Đánh giá tích cực	Đánh giá tiêu cực
Tổng số tài liệu	1000	500	500
Tập dữ liệu huấn luyện	900	450	450
Tập dữ liệu kiểm thử	100	50	50

Bảng 4.1: Phân chia dữ liệu

4.3. Quá trình đánh giá

Dữ liệu ban đầu gồm 1000 mẫu dữ liệu (1000 comments của người sử dụng các sản phẩm công nghệ), trong đó có 500 mẫu nhận xét tích cực và 500 mẫu nhận xét tiêu cực được chia làm 10 phần bằng nhau (k=10) với kí hiệu: S_1, S_2, …, S_{10}.

Trung bình, mỗi tập S_i có chứ 100 mẫu, trong đó có 50 mẫu nhận xét tích cực và 50 mẫu nhận xét tiêu cực.

Áp dụng công thức đã trình bày ở mục 4.1, từ quá trình sử dụng libSVM ta thu được kết quả như sau:

- *Trường hợp sử dụng bộ đặc trưng 400 từ:*

- *i=1:* Precision = $\frac{tP}{(tP+fP)} = \frac{21}{(21+3)} = \frac{21}{24}$

Recall = $\frac{tP}{(tP+fN)} = \frac{21}{(21+29)} = \frac{21}{50}$

- *i=1:* Precision = $\frac{tP}{(tP+fP)} = \frac{21}{(21+3)} = \frac{21}{24}$

  Recall = $\frac{tP}{(tP+fN)} = \frac{21}{(21+29)} = \frac{21}{50}$
- *i=1:* Precision = $\frac{tP}{(tP+fP)} = \frac{28}{(28+4)} = \frac{28}{32}$

  Recall = $\frac{tP}{(tP+fN)} = \frac{28}{(28+22)} = \frac{28}{50}$
- *i=2:* Precision = $\frac{tP}{(tP+fP)} = \frac{25}{(25+9)} = \frac{25}{36}$

  Recall = $\frac{tP}{(tP+fN)} = \frac{25}{(25+25)} = \frac{25}{50}$
- *i=3:* Precision = $\frac{tP}{(tP+fP)} = \frac{25}{(25+7)} = \frac{25}{32}$

  Recall = $\frac{tP}{(tP+fN)} = \frac{25}{(25+25)} = \frac{25}{50}$
- *i=4:* Precision = $\frac{tP}{(tP+fP)} = \frac{27}{(27+18)} = \frac{27}{45}$

  Recall = $\frac{tP}{(tP+fN)} = \frac{27}{(27+23)} = \frac{27}{50}$
- *i=5:* Precision = $\frac{tP}{(tP+fP)} = \frac{28}{(28+6)} = \frac{28}{34}$

  Recall = $\frac{tP}{(tP+fN)} = \frac{28}{(28+22)} = \frac{28}{50}$
- *i=6:* Precision = $\frac{tP}{(tP+fP)} = \frac{29}{(29+12)} = \frac{29}{41}$

  Recall = $\frac{tP}{(tP+fN)} = \frac{29}{(29+21)} = \frac{29}{50}$
- *i=7:* Precision = $\frac{tP}{(tP+fP)} = \frac{32}{(32+7)} = \frac{32}{39}$

  Recall = $\frac{tP}{(tP+fN)} = \frac{32}{(32+18)} = \frac{32}{50}$
- *i=8:* Precision = $\frac{tP}{(tP+fP)} = \frac{22}{(22+7)} = \frac{22}{29}$

  Recall = $\frac{tP}{(tP+fN)} = \frac{22}{(22+28)} = \frac{22}{50}$
- *i=9:* Precision = $\frac{tP}{(tP+fP)} = \frac{29}{(29+25)} = \frac{29}{54}$

  Recall = $\frac{tP}{(tP+fN)} = \frac{29}{(29+21)} = \frac{29}{50}$
- *i=10:* Precision = $\frac{tP}{(tP+fP)} = \frac{24}{(24+5)} = \frac{24}{29}$

  Recall = $\frac{tP}{(tP+fN)} = \frac{24}{(24+26)} = \frac{24}{50}$

Dưới đây là bảng tổng hợp kết quả dựa vào các công thức trên:

	Tập dữ liệu kiểm tra	Tập dữ liệu huấn luyện	Precision (%)	Recall (%)
i = 1	S_1	S_2, …, S_{10}	87.5	56
i = 2	S_2	S_1,S_3, …, S_{10}	69.4444	50
i = 3	S_3	S_1,S_2,S_4, …, S_{10}	78.125	50
i = 4	S_4	S_1,S_2,S_3,S_5, …, S_{10}	60	54
i = 5	S_5	S_1, …, S_4, S_6, …, S_{10}	82.3529	56
i = 6	S_6	S_1, …, S_5, S_7, …, S_{10}	70.7317	58
i = 7	S_7	S_1, …, S_6, S_8, …, S_{10}	82.0513	64
i = 8	S_8	S_1, …, S_7, S_9, S_{10}	75.8621	44
i = 9	S_9	S_1, …, S_8, S_{10}	64.4444	58
i = 10	S_{10}	S_1, …, S_9	82.7586	48
TB			**74.68**	**53**

Bảng 4.2: Kết quả đánh giá của tập dữ liệu với bộ đặc trưng 400 từ

- *Trường hợp sử dụng bộ đặc trưng 300 từ:*

  Áp dụng công thức tương tự như trường hợp trên để tính kết quả trong bảng 4.3.

	Tập dữ liệu kiểm tra	Tập dữ liệu huấn luyện	Precision (%)	Recall (%)
i = 1	S_1	S_2, …, S_{10}	87.5	42
i = 2	S_2	S_1,S_3, …, S_{10}	65.3846	34
i = 3	S_3	S_1,S_2,S_4, …, S_{10}	60	36
i = 4	S_4	S_1,S_2,S_3,S_5, …, S_{10}	62.5	40
i = 5	S_5	S_1, …, S_4, S_6, …, S_{10}	66.667	44
i = 6	S_6	S_1, …, S_5, S_7, …, S_{10}	69.444	50
i = 7	S_7	S_1, …, S_6, S_8, …, S_{10}	70.9677	44
i = 8	S_8	S_1, …, S_7, S_9, S_{10}	66.6667	20
i = 9	S_9	S_1, …, S_8, S_{10}	78.125	50
i = 10	S_{10}	S_1, …, S_9	66.6677	44
TB			**69**	**40.4**

Bảng 4.3: Kết quả đánh giá của tập dữ liệu với bộ đặc trưng 300 từ

- *Trường hợp sử dụng bộ đặc trưng 200 từ*

	Tập dữ liệu kiểm tra	Tập dữ liệu huấn luyện	Precision (%)	Recall (%)
i = 1	S_1	S_2, …, S_{10}	68.4211	52
i = 2	S_2	S_1,S_3, …, S_{10}	60.6061	40
i = 3	S_3	S_1,S_2,S_4, …, S_{10}	68.1818	30
i = 4	S_4	S_1,S_2,S_3,S_5, …, S_{10}	68.8696	28
i = 5	S_5	S_1, …, S_4, S_6, …, S_{10}	58.0645	36
i = 6	S_6	S_1, …, S_5, S_7, …, S_{10}	68.75	44
i = 7	S_7	S_1, …, S_6, S_8, …, S_{10}	61.5385	15.6863
i = 8	S_8	S_1, …, S_7, S_9, S_{10}	60	54
i = 9	S_9	S_1, …, S_8, S_{10}	54.1166	26
i = 10	S_{10}	S_1, …, S_9	59.0903	26
TB			**63**	**36.81**

Bảng 4.4: Kết quả đánh giá của tập dữ liệu với bộ đặc trưng 200 từ

4.4. Kết quả

Như vậy sau 10 lần kiểm tra với bộ đặ trưng 400 từ, tập dữ liệu kiểm tra và tập dữ liệu huấn luyện là riêng biệt với nhau, độ chính xác trung bình của hệ thống đạt kết quả ở mức 74,68%.

Tuy nhiên, bên cạnh đó, chúng ta thực hiện việc kiểm thử với số lượng từ đặc trưng khác nhau và thu được kết quả:

	Độ chính xác trung bình (%)	Độ bao phủ trung bình (%)
400	74.68	53
300	69	40.4
200	63	36.81

Bảng 4.5: Kết quả đánh giá của hệ thống với mỗi bộ từ đặc trưng khác nhau

Dựa vào bảng 4.5 có thể nhận thấy, việc thay đổi số lượng từ đặc trưng có ảnh hưởng không hề nhỏ đến kết quả của hệ thống. Vì vậy, việc lựa chọn bộ đặc trưng thích hợp cũng quyết định đến kết quả mà mô hình phân lớp đem lại.

KẾT LUẬN

Cho đến nay, bài toán phân loại quan điểm nói riêng và khai phá quan điểm nói chung vẫn luôn là mối quan tâm hàng đầu của các nhà khoa học. Đã có nhiều phương pháp được lựa chọn để giải quyết bài toán, mỗi phương pháp có ưu, nhược điểm riêng phù hợp với từng trường hợp cụ thể trong thực nghiệm. Nếu như ban đầu, việc phân loại quan điểm được áp dụng phổ biến đối với ngôn ngữ tiếng Anh, Ấn, Pháp… thì cho đến hiện tại, phạm vi ngôn ngữ đã được mở rông hơn rất nhiều, kể cả với những ngôn ngữ đòi hỏi phải có những phương pháp xử lý phân lớp riêng biệt và phức tạp hơn, trong đó có tiếng Việt. Nhiều nghiên cứu của các nhà khoa học đã giải quyết vấn đề này và đạt hiệu quả tương đối cao.

Mô hình xây dựng trong đồ án này đã phân loại được quan điểm người dùng về sản phẩm công nghệ dựa trên phương pháp học máy Support Vector Machine (SVM). Không thể phủ nhận SVM là một trong những phương pháp sử dụng trong phân lớp tài liệu được đánh giá cao bởi hiệu quả mà phương pháp này đem lại. Cùng với sự phát triển không ngừng của khoa học máy tính hương pháp học máy có giám sát đang trở thành một xu thế phổ biến trong hệ trợ giúp quyết định hay các hệ thống thông minh.

Trong đồ án này có hai công cụ hỗ trợ được sử dụng dựa trên phương pháp học máy vector hỗ trợ là *SVM* ^{light} và bộ thư viện *libSVM*. Với đặc điểm là gọn nhẹ, yêu cầu cấu hình thấp, dễ sử dụng nên hai công cụ này đã hỗ trợ rất nhiều cho việc giải quyết bài toán phân lớp quan điểm. Hệ thống cũng giải quyết được vấn đề thu thập dữ liệu bằng cách lấy tự động thông qua việc sử dụng công cụ hỗ trợ và API của Google Custom Search. Do đó việc lấy dữ liệu trở nên linh hoạt, tiết kiệm thời gian cũng như đem lại hiệu quả cao hơn so với các cách truyền thống thông thường. Bên cạnh đó, hệ thống được thiết kế trên công nghệ java nên có thể ứng dụng trong nhiều hệ thống khác nhau.

Tuy nhiên, hệ thống vẫn còn nhiều hạn chế như độ chính xác chưa cao. Điều này có thể nhìn thấy từ bảng kết quả đánh giá. Bởi ngay từ chính nội dung ban đầu hệ thống thu thập được gồm các bình luận, đánh giá ý kiến của các cá nhận trên website. Có thể hiểu dữ liệu mà hệ thống cần xử lý ở đây chính là ngôn ngữ tự nhiên bằng tiếng Việt. Khó khăn lớn nhất là cấu trúc từ trong tiếng Việt và việc đánh giá ngôn ngữ tự nhiên là không theo một qui luật hay thuật toán nào mà phụ thuộc hoàn toàn vào hoàn cảnh xuất hiện câu bình luận, kiến thức ngôn ngữ cũng như cảm quan của người đánh giá. Một yếu tố khác cũng ảnh hưởng đến chất ượng hệ thống là bộ các từ đặc trưng. Các từ đặc trưng nên là những từ thể hiện rõ quan điểm và sát với đối tượng của đề tài, nhưng khi số lượng từ trong bộ đặc trưng quá nhiều hay quá ít sẽ gây "nhiễu" cho quá trình huấn luyện. Nên việc lựa chọn bộ đặc trưng "phù hợp" là một yếu tố quan trọng không kém quyết định kết quả hệ thống. Với đặc thù như vậy, việc xây dựng bộ dữ huấn luyện thường do các chuyên gia có kinh nghiệm đảm nhiệm, mất rất nhiều thời gian, tốn nhiều tiền bạc và số lượng dữ liệu lại ít so với dữ liệu thực trong thực tế.

Qua việc nghiên cứu và xây dựng thệ thống hoàn chỉnh đã cho thấy tầm quan trọng của phân loại quan điểm đối với các cá nhân, cộng đồng và dựa trên những mặt hạn chế của hệ thống để tiếp tục nghiên cứu, tìm giải pháp khắc phúc các vấn đề còn tồn đọng gồm có việc gán nhãn lớp cho dữ liệu kiểu ngôn ngữ tự nhiên và xử lý linh hoạt hơn khi phải làm việc với lượng dữ liệu lớn, đặc biệt là bộ các từ đặc trưng "phù hợp.

DANH MỤC TÀI LIỆU THAM KHẢO

Tiếng Anh:

[1]. Aixin Sun, Ee-Peng, Wee-Keong Ng. Sun (2002). Web classification using support vector machine. Proceeding of the 4th International Workshop on Web Information and Data Management, McLean, Virginia, USA, 2002 (ACM Press).

[3]. Ian H. Witten, Eibe Frank, Mark A. Hall (2011). Data Mining Practical Machine Learning Tools and Techniques Third Edition, Burlington, USA.

[4]. Minqing Hu and Bing Liu. "Mining and summarizing customer reviews". Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2004, full paper), Seattle, Washington, USA, Aug 22-25, 2004.

[5]. Pang, B., Lee, L., and Vaithyanathan, S., 2002. Thumbs up? Sentiment Classification Using Machine Learning Techniques. In Proc.of EMNLP 2002.

[6]. Turney, P. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. ACL' 02.

Tiếng Việt:

[7]. Lưu Công Tố (2011). "Mở rộng VietSentiWordNet dựa trên mô hình học bán giám sát SVMlight và áp dụng vào bài toán khai phá quan điểm", Khóa luận tốt nghiệp đại học, Trường Đại học Công nghệ.

[8]. Vũ Xuân Sơn, Trần Trung Hiếu, Lê Thu Hà, Đào Thủy Ngân. Xây dựng từ điển VietSentiWordNet ứng dụng khai phá quan điểm trên tin tức. Công trình tham gia giải thưởng "Sinh viên nghiên cứu khoa học" ,Đại học Công nghệ, (năm 2011).

[10] Trần Thị Thu Huyền, Nguyễn Thị Thảo, Nguyễn Thị Huyền, Đoàn Thị Thu Hà , Nguyễn Thị Thủy. Phương pháp phân lớp sử dụng máy vec – tơ hỗ trợ ứng dụng trong tin sinh học. Tạp chí Khoa học và Phát triển, tập 9, số 6: 1021 - 1031 Trường đại học Nông nghiệp Hà Nội. (năm 2011)

Danh mục các Website tham khảo

[2]. http://en.wikipedia.org/wiki/Support_vector_machine

[10]. http://svmlight.joachims.org/

[11]. http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/eval/