ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Trần Thị Oanh

THUẬT TOÁN SELF-TRAINING VÀ CO-TRAINING ỨNG DỤNG TRONG PHÂN LỚP VĂN BẢN

KHOÁ LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUI

Ngành: Công nghệ thông tin

HÀ NỘI – 2006

ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Trần Thị Oanh

THUẬT TOÁN SELF-TRAINING VÀ CO-TRAINING ỨNG DỤNG TRONG PHÂN LỚP VĂN BẢN

KHOÁ LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUI

Ngành: Công nghệ thông tin

Cán bộ hướng dẫn: TS Hà Quang Thuỵ

Cán bộ đồng hướng dẫn: NCS Lê Anh Cường

HÀ NỘI – 2006

Lời cảm ơn

Trước tiên, tôi xin gửi lời cảm ơn chân thành và sự biết ơn sâu sắc tới Tiến sĩ Hà Quang Thuỵ (trường Đại học Công nghệ) và NCS Lê Anh Cường (Japan Advanced Institute of Science and Technology) đã tận tình hướng dẫn tôi trong suốt quá trình thực hiện khoá luận này.

Tôi xin bày tỏ lời cảm ơn sâu sắc đến các thầy cô giáo đã giảng dạy tôi trong suốt bốn năm học qua, đã cho tôi những kiến thức quí báu để tôi có thể vững bước trên con đường đi của mình.

Tôi xin gửi lời cảm ơn các anh chị trong nhóm seminar về khai phá dữ liệu: anh Nguyễn Việt Cường, anh Đặng Thanh Hải, chị Nguyễn Cẩm Tú, … đã nhiệt tình chỉ bảo trong quá trình tôi tham gia nghiên cứu khoa học và làm khoá luận.

Tôi xin gửi lời cảm ơn tới các bạn trong lớp K47CC, K47CA đã ủng hộ, khuyến khích tôi trong suốt quá trình học tập tại trường.

Và lời cuối cùng, tôi xin bày tỏ lòng chân thành và biết ơn vô hạn tới cha mẹ, và các anh chị tôi, những người luôn ở bên cạnh tôi những lúc tôi khó khăn nhất, giúp tôi vượt qua khó khăn trong học tập cũng như trong cuộc sống.

Hà Nội, ngày 24 tháng 05 năm 2006

Sinh viên

Trần Thị Oanh

TÓM TẮT NỘI DUNG

Hiện nay, tồn tại một số thuật toán học phân lớp văn bản thực hiện có kết quả rất tốt khi được xây dựng dựa trên một tập ví dụ học lớn. Tuy nhiên, trong thi hành thực tế thì điều kiện này hết sức khó khăn vì ví dụ học thường được gán nhãn bởi con người nên đòi hỏi rất nhiều thời gian và công sức. Trong khi đó, các dữ liệu chưa gán nhãn (unlabeled data) thì lại rất phong phú. Do vậy, việc xem xét các thuật toán học không cần nhiều dữ liệu gán nhãn, có khả năng tận dụng được nguồn rất phong phú các dữ liệu chưa gán nhãn nhận được sự quan tâm của nhiều nhà khoa học trên thế giới. Việc học này được đề cập đến với tên gọi là học bán giám sát.

Trong khóa luận này, chúng tôi khảo sát hai thuật toán học bán giám sát điển hình nhất, đó là self-training và co-training và đề xuất một số kỹ thuật làm trơn. Khóa luận cũng tiến hành ứng dụng các nghiên cứu nói trên vào bài toán phân lớp văn bản và cho kết quả rất khả quan .

MỤC LỤC

Bảng các ký hiệu và chữ viết tắt

EM: *E*xpectation-*M*aximization.

i.i.d : *i*ndependent and *i*dentically *d*istributed random variables.

PAC: *P*robably *A*pproximately *C*orrect.

SAE: Selected Added Examples.

TSVM: *T*ransductive *S*upport *V*ector *M*achine.

WSD: *W*ord *S*ense *D*isambiguation.

Danh mục hình vẽ

Danh mục các bảng biểu

MỞ ĐẦU

Hiện nay, tồn tại một số thuật toán học phân lớp văn bản thực hiện có kết quả rất tốt khi được xây dựng dựa trên một tập ví dụ học (dữ liệu được gán nhãn - *labeled data*) lớn. Tuy nhiên, trong thực tế thực thi điều kiện có được tập ví dụ lớn là hết sức khó khăn vì ví dụ học thường phải do con người gán nhãn cho nên đòi hỏi rất nhiều thời gian và công sức. Trong khi đó, các dữ liệu chưa gán nhãn (*unlabeled data*) thì lại rất phong phú. Đối với các bài toán học phân lớp dữ liệu văn bản, đặc biệt là phâp lớp trang Web, vấn đề nói trên trở nên phổ biến hơn. Do vậy, việc xem xét các thuật toán học không cần nhiều dữ liệu gán nhãn, có khả năng tận dụng được nguồn rất phong phú các dữ liệu chưa gán nhãn nhận được sự quan tâm của nhiều nhà khoa học trên thế giới. Việc học này được đề cập tới là việc học bán giám sát. Vào tháng 1-2006, Xiaojin Zhu đã cho một cái nhìn tổng quan về các thuật toán nói trên [23].

Học bán giám sát (semi-supervised learning) là việc học trên cả dữ liệu gán nhãn và dữ liệu chưa gán nhãn. Phương pháp sử dụng một số lượng lớn các dữ liệu chưa gán nhãn, và một lượng nhỏ dữ liệu được gán nhãn ban đầu (thường được gọi là *seed set*) để xây dựng một bộ phân lớp. Vì thông tin được bổ sung từ dữ liệu chưa gán nhãn, tiềm năng sẽ thu được một bộ phân lớp mới tốt hơn bộ phân lớp chỉ xây dựng trên dữ liệu gán nhãn. Có nhiều thuật toán học bán giám sát, điển hình như các thuật toán EM [20], TSVM (transductive support vector machine) [13], SGT (spectral graph transductive) [12]. Trong phạm vi khóa luận này, chúng tôi tập trung vào hai thuật toán thông dụng nhất là thuật toán self-training và co-training. Mục tiêu đặt ra cho khóa luận là khảo sát, phân tích kỹ lưỡng hai thuật toán này nhằm đề xuất một số kỹ thuật làm trơn chúng và ứng dụng chúng trong bài toán phân lớp trang Web.

Khóa luận được tổ chức thành bốn chương chính với nội dung cơ bản như sau:

- ***Chương 1*** trình bày tổng quan về phân lớp văn bản và học bán giám sát. Trước khi giới thiệu về phân lớp văn bản bán giám sát, khóa luận trình bày những nét cơ bản nhất về phân lớp văn bản có giám sát với thuật toán phân lớp điển hình là Naïve Bayes. Sau đó khóa luận giới thiệu về thuật toán học bán giám sát và đối sánh với thuật toán học giám sát.

- ***Chương 2*** trình bày hai thuật toán self-training và co-training. Phần đầu chương giới thiệu hai thuật toán học bán giám sát Self-training, Co-training và đánh giá chúng. Thông qua đó, khóa luận đề xuất một số kỹ thuật làm trơn và mô hình thi hành thuật toán self-training và co-training trên cơ sở thuật toán Naïve Bayes.

• Thực nghiệm phân lớp trang web được trình bày trong ***Chương 3***. Nội dung thực nghiệm các phương pháp Naïve Bayes được mô tả chi tiết cùng với một số nhận xét đánh về giá kết quả thực nghiệm.

• ***Phần Kết luận*** tổng hợp các kết quả đạt được của khóa luận và nêu một số phương hướng nghiên cứu tiếp theo.

Chương 1 TỔNG QUAN VỀ PHÂN LỚP VĂN BẢN VÀ HỌC BÁN GIÁM SÁT

1.1. Phân lớp văn bản

Phân lớp văn bản là việc gán một văn bản (tài liệu) được biểu diễn trong ngôn ngữ tự nhiên vào một hoặc nhiều lớp đã được xác định trước. Đầu tiên, người ta xây dựng một mô hình miêu tả một tập hợp ban đầu các văn bản (thường được đề cập như là việc học giám sát) dựa trên một tập các dữ liệu huấn luyện. Tập dữ liệu huấn luyện là tập các trang văn bản đã được gán nhãn lớp tương ứng cho chúng. Quá trình xây dựng tập dữ liệu huấn luyện này thường được thực hiện bằng con người. Sau đó, mô hình được sử dụng để phân lớp các trang văn bản chưa được gán nhãn.

Bộ phân lớp có thể được xây dựng bằng tay dựa vào các kỹ thuật ứng dụng tri thức (thường là xây dựng một tập các tri thức) hoặc có thể được xây dựng một cách tự động bằng các kỹ thuật học máy thông qua một tập các dữ liệu huấn luyện được định nghĩa trước phân lớp tương ứng. Trong hướng tiếp cận học máy, ta chú ý đến các vấn đề sau:

- *Biểu diễn văn bản*

  Thông thường, một văn bản được biểu diễn bằng một vector trọng số, độ dài của vector là số các từ khóa *(keyword / term)* xuất hiện trong ít nhất một mẫu dữ liệu huấn luyện. Biểu diễn trọng số có thể là nhị phân (từ khóa đó có hay không xuất hiện trong văn bản tương ứng) hoặc không nhị phân (từ khóa đó đóng góp tỷ trọng bao nhiêu cho ngữ nghĩa văn bản). Tồn tại một số phương pháp biểu diễn từ khoá điển hình như IDF, TF, TF-IDF,...

- *Loại bỏ các từ dừng và lấy từ gốc*

  Trước khi đánh trọng số cho các từ khoá cần tiến hành loại bỏ các từ dừng (*stop-word*). Từ điển Wikipedia định nghĩa: "Từ dừng là những từ xuất hiện thường xuyên nhưng lại không có ích trong đánh chỉ mục cũng như sử dụng trong các máy tìm kiếm hoặc các chỉ mục tìm kiếm khác". Thông thường, các trạng từ, giới từ, liên từ là các từ dừng. Trong tiếng Anh, người ta đã liệt kê ra danh sách các từ dừng nhưng với tiếng Việt thì chưa có một danh sách từ dừng

như vậy. Tuy nhiên, có thể liệt kê danh sách các từ dừng cho tiếng Việt mặc dù có thể là không đầy đủ (chẳng hạn, xem trong Phụ lục).

Việc lấy từ gốc và lưu lại các từ phát sinh từ mỗi từ gốc để nâng cao khả năng tìm kiếm được áp dụng cho các ngôn ngữ tự nhiên có chia từ, chẳng hạn như tiếng Anh.

- Tiêu chuẩn đánh giá:

Phân lớp văn bản được coi là không mang tính khách quan theo nghĩa dù con người hay bộ phân lớp tự động thực hiện việc phân lớp thì đều có thể xảy ra sai sót. Tính đa nghĩa của ngôn ngữ tự nhiên, sự phức tạp của bài toán phân lớp được coi là những nguyên nhân điển hình nhất của sai sót phân lớp. Hiệu quả của bộ phân lớp thường được đánh giá qua so sánh quyết định của bộ phân lớp đó với quyết định của con người khi tiến hành trên một tập kiểm thử (test set) các văn bản đã được gán nhãn lớp trước. Có ba độ đo điển hình được sử dụng để đánh giá độ hiệu quả của thuật toán phân lớp, đó là độ chính xác π (precision), độ hồi tưởng ρ (recall) và độ đo F1 được tính lần lượt theo công thức (1.1), (1.2), (1.3).

- $$precision = \frac{true\_positive}{(true\_positive)+(true\_negative)} \times 100 \qquad (1.1)$$

- $$recall = \frac{true\_positive}{(true\_positive)+(false\_positive)} \times 100 \qquad (1.2)$$

- $$F_1(recall, precision) = \frac{2 \times recall \times precision}{recall \times precision} \qquad (1.3)$$

Trong các công thức này, *positive* / *negative* liên quan tới các ví dụ còn *true* / *false* liên quan tới kết quả thực hiện của bộ phân lớp. Cụ thể, đại lượng *true_positive* để chỉ số lượng ví dụ *positive* mà bộ phân lớp cho là *đúng* thuộc lớp, đại lượng *true_nagative* để chỉ số lượng ví dụ *nagative* mà bộ phân lớp cũng cho là *đúng* thuộc lớp, còn đại lượng *false_positive* để chỉ số lượng ví dụ *positive* mà bộ phân lớp lại coi là không thuộc lớp.

Bài toán phân lớp văn bản có rất nhiều ứng dụng trong thực tế, điển hình là các ứng dụng lọc trên Internet.

1.2. Thuật toán phân lớp văn bản điển hình

Có rất nhiều thuật toán phân lớp có giám sát thực hiện phân lớp văn bản rất tốt như thuật toán k người láng giềng gần nhất (kNN), cây quyết định hay Naïve Bayes,… Ở đây, chúng tôi xin trình bày chi tiết thuật toán Naïve Bayes được sử dụng trong thực nghiệm của khoá luận.

1.2.1. Thuật toán Naive Bayes

Bộ phân lớp Naive Bayes thừa nhận một giả thiết mạnh (strong assumptions) là các đặc trưng (feature) là độc lập lẫn nhau. Thêm vào đó, bộ phân lớp xác suất lựa chọn một vài dạng giả định cho phân phối của mỗi đặc trưng trong một lớp. Những mô hình xác suất phổ biến nhất là mô hình đa thức (multinomial model), mô hình độc lập nhị phân (binary independence model) và một số mô hình khác:

- *Binary Independence Model (Multi-variate Bernoulli model).*
- ***Multinomial Model***
- *Poisson Naive Bayes Model*
- *Connection between Poisson and Multinomial Model*
- *Multinomial word model*
- *Negative binomial Naive Bayes Model*

Qua tìm hiểu các mô hình phân lớp Naive Bayes, chúng tôi quyết định sử dụng mô hình đa thức *(multinomial model)* vì nó đã được chứng minh là tốt nhất so với các mô hình còn lại trong nhiều trường hợp của phân lớp văn bản [3,15,22]. Mô hình đa thức biểu diễn văn bản bằng tập các lần xuất hiện của các từ. Mô hình không quan tâm đến trật tự của từ mà chỉ quan tâm đến số lần xuất hiện của các từ trong một văn bản.

Nội dung mô hình học phân lớp đa thức Naïve Bayes được mô tả như sau. Giả thiết rằng văn bản được tạo ra bởi một mô hình trộn (mixture model) với tham số θ. Mô hình trộn bao gồm các thành phần trộn $c_j \subset C = \{c_1, ..., c_{|C|}\}$. Mỗi một văn bản d_i được tạo ra bằng cách:

- Lựa chọn một thành phần dựa theo các ưu tiên của nó, $P(c_j;\theta)$

- Sau đó, mô hình trộn tạo ra văn bản dựa trên các tham số của nó, với phân phối $P(d_i|c_j;\theta)$.

Chúng ta mô tả likelihood của một văn bản là tổng xác suất của tất cả các thành phần trộn.

$$P(d_i|\theta) = \sum_{j=1}^{|C|} P(c_j|\theta)P(d_i|c_j;\theta) \quad (1.4)$$

Mỗi văn bản có một nhãn lớp, giả sử rằng có sự tương ứng một-một giữa nhãn lớp và thành phần của mô hình trộn, vì vậy, ta sẽ sử dụng c_j vừa để biểu diễn thành phần trộn thứ j vừa biểu diễn phân lớp thứ j. Trong mô hình đa thức, ta giả thiết rằng:

- Độ dài của văn bản là độc lập với phân lớp của nó.
- Giả thiết Naive Bayes: Xác suất sự xuất hiện của từ trong một văn bản là độc lập với ngữ cảnh và vị trí của từ trong văn bản đó.

Vì vậy, mỗi văn bản d_i được tạo ra từ phân phối đa thức của các từ với nhiều lần thử nghiệm độc lập với độ dài của văn bản. Ta định nghĩa N_{it} là số lần xuất hiện của từ w_t trong văn bản d_i, thì xác suất của văn bản d_i khi biết trước phân lớp đơn giản là phân phối đa thức như công thức (1.5):

$$P(d_i|c_j;\theta) = P(|d_i|)|d_i|!\prod_{t=1}^{|V|}\frac{P(w_t|c_j;\theta)^{N_{it}}}{N_{it}!} \quad (1.5)$$

Dựa vào công thức, ta tính toán tối ưu Bayes cho những đánh giá này từ tập dữ liệu huấn luyện. Ở đây, ước lượng cho xác xuất của từ w_t trong văn bản thuộc lớp c_j được tính theo công thức (1.6):

$$P(w_t/c_j) = \frac{1+\sum_{i=1}^{/D/} N_{it}P(c_j/d_i)}{/V/+\sum_{s=1}^{|V|}\sum_{i=1}^{/D/} N_{is}P(c_j/d_i)} \quad (1.6)$$

với $P(c_j|d_i) \in \{0, 1\}$ được xác định bởi nhãn lớp tương ứng của mỗi mẫu dữ liệu.

Xác suất ưu tiên của mỗi lớp được tính đơn giản dựa trên mỗi lớp thay vì trên các từ như công thức (1.7).

$$P(c_j) = \frac{\sum_{i=1}^{|D|} P(c_j | d_i)}{|D|} \tag{1.7}$$

Từ việc ước lượng các tham số trên dữ liệu huấn luyện theo các phương trình (1.5), (1.6), và (1.7) ta thực hiện phân lớp các văn bản kiểm thử và lựa chọn phân lớp với xác suất cao nhất theo qui tắc Bayes.

$$P(c_j | d_i) = \frac{P(c_j)P(d_i | c_j)}{P(d_i)} \tag{1.8}$$

Vì tính xác suất cho cùng một văn bản và do giả thiết Naive Bayes nên công thức (1.8) sẽ tương đương với công thức (1.9):

$$P(c_j | d_i) \alpha P(c_j)P(d_i | c_j)$$
$$= P(c_j) \prod_{k=1}^{|d_i|} P(w_{d_{i,k}} | c_j) \tag{1.9}$$

1.3. Tổng quan về học bán giám sát

Khi xử lý các bài toán phân lớp văn bản tự động ta thấy tồn tại một số lượng khổng lồ các dữ liệu văn bản trên WWW, thư điện tử, cơ sở dữ liệu tổng hợp, thư viện số, các bản ghi tình trạng của bệnh nhân, ... Các thuật toán học mang tính thống kê có thể được huấn luyện để phân lớp xấp xỉ các dữ liệu đó vào chủ đề tương ứng của nó. Một vài thuật toán học phân lớp văn bản đã được sử dụng để phân lớp các bài báo (Lewis & Gale, 1994; Joachims,1998), phân lớp trang web (Craven, DiPasquo, Freitag, McCallum, Mitchell, Nigam, & Slattery, 1998; Shavlik & Eliassi-Rad, 1998), tự động học thêm các sở thích về việc đọc của người dùng (Pazzani, Muramatsu, & Billsus, 1996; Lang, 1995), tự động sắp xếp thư điện tử (Lewis & Knowles, 1997; Sahami, Dumais, Heckerman, &Horvitz, 1998) (theo [20]).

Tuy nhiên, các thuật toán học này lại gặp phải khó khăn là: Để xây dựng được bộ phân lớp có độ tin cậy cao đòi hỏi phải có một số lượng lớn các mẫu dữ liệu huấn

luyện (chính là các văn bản đã được gán nhãn lớp tương ứng). Các dữ liệu huấn luyện này rất hiếm và đắt vì chúng thường được thực hiện bởi con người – một tiến trình tốn thời gian và công sức.

Ví dụ bài toán học để nhận biết được những bài báo, nhóm tin tức UseNet nào mà người dùng quan tâm. Khi đó hệ thống phải lọc, sắp xếp trước các bài báo và chỉ đưa ra các bài báo mà người dùng có thể quan tâm đến nhất – một bài toán đang thu hút được sự chú ý ngày nay. Theo [20], Lang đã phát hiện rằng, sau khi một người đọc và gán nhãn khoảng 1000 bài báo, một bộ phân lớp được huấn luyện qua chúng sẽ thu được độ chính xác khoảng 50% khi dự đoán chỉ 10% các bài báo có độ tin cậy cao nhất. Tuy nhiên, hầu hết người sử dụng hệ thống thực sẽ không có đủ kiên nhẫn để gán nhãn hàng nghìn bài báo – đặc biệt chỉ để thu được độ chính xác trên. Do đó vấn đề đặt ra là xây dựng một thuật toán đưa ra sự phân lớp chính xác mà chỉ cần một số lượng nhỏ dữ liệu học, tức chỉ với vài chục bài báo được gán thay vì hàng nghìn bài báo.

Nhu cầu về một lượng lớn các dữ liệu học và những khó khăn để thu được các dữ liệu đó đặt ra một câu hỏi quan trọng: Liệu có thể sử dụng được nguồn thông tin nào khác trong phân lớp văn bản mà có thể làm giảm sự cần thiết của dữ liệu gán nhãn? Đây chính là nguồn động lực thúc đẩy sự phát triển của các phương pháp học bán giám sát (semi-supervised learning).

Nhìn vào sự tồn tại của dữ liệu ta thấy, trong thực tế dữ liệu thường tồn tại ở dạng trung gian: Không phải tất cả dữ liệu đều được gán nhãn cũng như không phải tất cả chúng đều chưa được gán nhãn. Bán giám sát là một phương pháp học sử dụng thông tin từ cả hai nguồn dữ liệu này.

Động lực thúc đẩy học bán giám sát: sự hiệu quả của học bán giám sát

Đã có rất nhiều các nghiên cứu về học bán giám sát. Những kết quả thực nghiệm cũng như lý thuyết đã chỉ ra rằng sử dụng cách tiếp cận đánh giá khả năng giống nhau cực đại (Maximum Likelihood) có thể cải tiến độ chính xác phân lớp khi có thêm các dữ liệu chưa gán nhãn[20].

Tuy nhiên, cũng có những nghiên cứu chỉ ra rằng, dữ liệu chưa gán nhãn có thể cải tiến độ chính xác phân lớp hay không là phụ thuộc vào cấu trúc bài toán có phù hợp với giả thiết của mô hình hay không? Gần đây, Cozman [11] đã thực nghiệm trên dữ liệu giả hướng vào tìm hiểu giá trị của dữ liệu chưa gán nhãn. Ông chỉ ra rằng, độ

chính xác phân lớp có thể giảm đi khi thêm vào ngày càng nhiều dữ liệu chưa gán nhãn. Nguyên nhân của sự giảm này là do sự không phù hợp giữa giả thiết của mô hình và phân phối dữ liệu thực tế.

Theo [6], để việc học bán giám sát mang lại hiệu quả cần một điều kiện tiên quyết là: Phân phối các mẫu cần phát hiện phải phù hợp với bài toán phân lớp. Về mặt công thức, các tri thức thu được từ dữ liệu chưa gán nhãn $p(x)$ phải mang lại thông tin hữu ích cho suy luận $p(x|y)$. Olivier Chapelle [6] đã đề xuất một giả thiết làm trơn, đó là hàm nhãn lớp ở vùng có mật độ cao thì trơn hơn ở vùng có mật độ thấp. Giả thiết được phát biểu như sau:

Giả thiết bán giám sát: *Nếu hai điểm x_1, x_2 thuộc vùng có mật độ cao là gần nhau thì đầu ra tương ứng của chúng nên là y_1, y_2.*

Giả thiết này ngụ ý là nếu hai điểm được liên kết bởi một đường dẫn trên vùng mật độ cao thì đầu ra của chúng nên gần nhau.

Đối với bài toán phân lớp văn bản, ta hình dung như sau: Dữ liệu chưa gán nhãn sẽ cung cấp thông tin về phân phối xác suất đồng thời (*joint probability distribution*) của các từ khóa. Ví dụ với bài toán phân lớp trang web với hai lớp: trang chủ của một khoá học và không phải trang chủ của một khoá học. Ta coi trang chủ của một khoá học là hàm đích. Vì vậy, trang chủ của một khoá học sẽ là mẫu dương *(positive example)*, và các trang còn lại là các mẫu âm *(negative example)*. Giả sử chỉ sử dụng dữ liệu gán nhãn ban đầu ta xác định các văn bản có chứa từ "bài tập"*("homework")* thường thuộc lớp dương. Nếu sử dụng quan sát này để gán nhãn các dữ liệu chưa gán nhãn, chúng ta lại xác định được từ "bài giảng"*("lecture"*) xuất hiện thường xuyên trong các văn bản chưa gán nhãn mà được dự đoán là thuộc lớp dương. Sự xuất hiện của các từ "bài tập" và "bài giảng" trên một tập lớn các dữ liệu huấn luyện chưa gán nhãn có thể cung cấp thông tin hữu ích để xây dựng một bộ phân lớp chính xác hơn – xem xét cả "bài tập" và "bài giảng" như là các thể hiện của các mẫu dương.

Để có thể hiểu được bản chất của học bán giám sát, đầu tiên chúng ta cần hiểu thế nào là học giám sát (*supervised*) và học không giám sát (*unsupervised).*

1.3.1. Học giám sát và học không giám sát

Trong lý thuyết xác suất, một dãy các biến ngẫu nhiên được gọi là có độc lập cùng phân phối nếu chúng có cùng một phân phối và độc lập với nhau[25]. Các quan

sát trong một mẫu thường được giả thiết là độc lập cùng phân phối (i.i.d) nhằm làm đơn giản hoá tính toán toán học bên dưới của nhiều phương pháp thống kê. Trong nhiều ứng dụng thực, điều này thường không thực tế.

Học không giám sát: Cho trước một mẫu chỉ gồm các đối tượng *(objects)*, cần tìm kiếm cấu trúc đáng quan tâm (*interesting structures*) của dữ liệu, và nhóm các đối tượng giống nhau. Biểu diễn toán học của phương pháp này như sau:

Đặt $X = (x_1, x_2, ..., x_n)$ là tập hợp gồm n mẫu *(examples or points)*, $x_i \in$ X với mọi i$\in$[n]:= {1,2, ..., n}. Thông thường, ta giả thiết rằng các mẫu được tạo ra một cách độc lập và giống nhau (*i.i.d – independently and identically distributed*) từ một phân phối chung trên X . Mục đích của học không giám sát là tìm ra một cấu trúc thông minh(*interesting structure)* trên tập dữ liệu đó.

Học giám sát: Cho trước một mẫu bao gồm các cặp đối tượng - nhãn (x_i, y_i), cần tìm ra mối quan hệ dự đoán giữa các đối tượng và các nhãn. Mục đích là học một phép ánh xạ từ x tới y, khi cho trước một tập huấn luyện gồm các cặp (x_i, y_i), trong đó $y_i \in$ Y gọi là các nhãn hoặc đích của các mẫu x_i . Nếu nhãn là các số, $y = (y_i)_{i\in[n]}^T$ biểu diễn vector cột của các nhãn. Như đã nêu, một yêu cầu chuẩn là các cặp (x_i, y_i) tuân theo giả thiết i.i.d trải khắp trên $X \times Y$. Nhiệm vụ được định rõ là, ta có thể tính toán được một phép ánh xạ thông qua thi hành dự đoán của nó trên tập kiểm thử. Nếu các nhãn lớp là liên tục, nhiệm vụ phân lớp được gọi là hồi quy (regression). Có hai họ thuật toán giám sát: generative model và discriminative model

- *Generative model*:

Phương pháp này sẽ tạo ra một mô hình mật độ phụ thuộc vào lớp *(class-conditional density)* $p(x|y)$ bằng một vài thủ tục học không giám sát. Một mật độ sinh có thể được suy luận bằng cách sử dụng lý thuyết Bayes.

$$p(x|y) = \frac{p(x|y)\,p(y)}{\int_y p(x|y)\,p(y)\,dy} \qquad (1.10)$$

Gọi là mô hình sinh vì ta có thể tự tạo ra các mẫu dữ liệu.

- *Discriminative model:*

Phương pháp này sẽ thay vì đánh giá x_i được tạo ra như thế nào mà tập trung đánh giá $p(y|x)$. Một vài phương pháp *discirminative* hạn chế chúng để mô hình xem $p(y|x)$ lớn hơn hoặc nhỏ hơn 0.5, ví dụ như SVM. Trong thực hành, phương pháp này thường được đánh giá là hiệu quả hơn phương pháp sinh *(generative)*.

Từ đó, ***học bán giám sát*** có thể được xem là:

- Học giám sát cộng thêm dữ liệu chưa gán nhãn (Supervised learning + additional unlabeled data).
- Học không giám sát cộng thêm dữ liệu gán nhãn (Unsupervised learning + additional labeled data).

Học bán giám sát chính là cách học sử dụng thông tin chứa trong cả dữ liệu chưa gán nhãn và tập dữ liệu huấn luyện. Các thuật toán học bán giám sát có nhiệm vụ chính là mở rộng tập các dữ liệu gán nhãn ban đầu. Hiệu quả của thuật toán phụ thuộc vào chất lượng của các mẫu gán nhãn được thêm vào ở mỗi vòng lặp và được đánh giá dựa trên hai tiêu chí:

- Các mẫu được thêm vào phải được gán nhãn một cách chính xác.
- Các mẫu được thêm vào phải mang lại thông tin hữu ích cho bộ phân lớp (hoặc dữ liệu huấn luyện).

1.3.2. Phạm vi sử dụng học bán giám sát

Các phương pháp học bán giám sát sẽ rất hữu ích khi dữ liệu chưa gán nhãn nhiều hơn dữ liệu gán nhãn. Việc thu được dữ liệu gán nhãn là rẻ, nhưng để gán nhãn chúng thì tốn rất nhiều thời gian, công sức và tiền bạc. Đó là tình trạng của rất nhiều các lĩnh vực ứng dụng trong học máy như:

- Trong nhận dạng lời nói, ta sẽ dễ dàng ghi lại một lượng lớn các bài diễn thuyết, nhưng để gán nhãn chúng yêu cầu con người phải lắng nghe rồi đánh máy sao chép lại.
- Sự phong phú của hàng tỉ các trang web sẵn sàng cho xử lý tự động, nhưng để phân lớp chúng một cách tin cậy đòi hỏi con người phải đọc chúng.
- ...

Học bán giám sát là việc học trên cả dữ liệu đã và chưa được gán nhãn. Từ một số lượng lớn các dữ liệu chưa được gán nhãn, và một lượng nhỏ dữ liệu đã được gán nhãn ban đầu (thường gọi là *seed set*) để xây dựng một bộ phân lớp thậm chí là tốt hơn. Trong quá trình học như thế phương pháp sẽ tận dụng được những thông tin phong phú của dữ liệu chưa gán nhãn (*unlabeled data)*, mà chỉ yêu cầu một số lượng rất nhỏ các dữ liệu đã gán nhãn (*labeled data).*

Ý tưởng chung là vẫn thu được kết quả tốt như đối với việc học trên tập một tập dữ liệu lớn đã được gán nhãn.

Có một câu hỏi là: Liệu các phương pháp học bán giám sát có ích hay không? Chính xác hơn là, so sánh với học giám sát chỉ sử dụng dữ liệu gán nhãn, ta có thể hy vọng vào sự chính xác của dự đoán khi xét thêm các điểm không gán nhãn. Câu trả lời là "có" dưới những giả thiết phù hợp *(certain assumptions)* của từng mô hình[6].

1.4. Một số phương pháp học bán giám sát

Có rất nhiều phương pháp học bán giám sát nên trước khi quyết định lựa chọn phương pháp học cho một bài toán cụ thể cần phải xem xét các giả thiết của mô hình. Theo [23], chúng ta nên sử dụng phương pháp học mà giả thiết của nó phù hợp với cấu trúc của bài toán. Việc lựa chọn này có thể là khó khăn trong thực tế, tuy nhiên ta có thử các gợi ý sau: Nếu các lớp tạo ra dữ liệu *có tính phân cụm cao* thì EM với mô hình trộn sinh có thể là một sự lựa chọn tốt; nếu *các features có sự phân chia tự nhiên thành hai tập* thì co-training có thể phù hợp; nếu hai mẫu dữ liệu với *các feature tương tự nhau hướng tới thuộc về cùng một lớp* thì có thể sử dụng các phương pháp dựa trên đồ thị; nếu *các bộ phân lớp giám sát được xây dựng từ trước là phức tạp và khó sửa đổi* thì self-training sẽ là một lựa chọn ưu tiên.

Trước khi đi vào trình bày chi tiết hai phương pháp học self-training và co-training, chúng ta sẽ tìm hiểu một số phương pháp học bán giám sát điển hình gồm: Thuật toán cực đại kỳ vọng toán, thuật toán SVM truyền dẫn và thuật toán phân hoạch đồ thị quang phổ.

1.4.1. Thuật toán cực đại kỳ vọng toán

Thuật toán cực đại kỳ vọng (Expectation Maximization - EM) là một thuật toán tổng quát đánh giá sự cực đại khả năng*(ML – Maximum Likelihood)* mà dữ liệu là không hoàn chỉnh *(incomplete data)* hoặc hàm likelihood liên quan đến các biến

ẩn*(latent variables)*[5,20]. Ở đây, hai khái niệm "*incomplete data*" và "*latent variables*" có liên quan đến nhau: Khi tồn tại biến ẩn, thì dữ liệu là không hoàn chỉnh vì ta không thể quan sát được giá trị của biến ẩn; tương tự như vậy khi dữ liệu là không hoàn chỉnh, ta cũng có thể liên tưởng đến một vài biến ẩn với dữ liệu thiếu

Thuật toán EM gồm hai bước lặp: E-step và M-step. Khởi đầu, nó gán giá trị ngẫu nhiên cho tất cả các tham số của mô hình. Sau đó, tiến hành lặp hai bước lặp sau:

E-step (Expectation step): Trong bước lặp này, nó tính toán likelihood mong muốn cho dữ liệu dựa trên các thiết lập tham số và *incomplete data*.

M-step (Maximization step): Tính toán lại tất cả các tham số sử dụng tất cả các dữ liệu. Khi đó, ta sẽ có một tập các tham số mới.

Tiến trình tiếp tục cho đến khi likelihood hội tụ, ví dụ như đạt tới cực đại địa phương. EM sử dụng hướng tiếp cận leo đồi, nên chỉ đảm bảo đạt được cực đại địa phương. Khi tồn tại nhiều cực đại, việc đạt tới cực đại toàn cục hay không là phụ thuộc vào điểm bắt đầu leo đồi. Nếu ta bắt đầu từ một đồi đúng *(right hill)*, ta sẽ có khả năng tìm được cực đại toàn cục. Tuy nhiên, việc tìm được *right hill* thường là rất khó. Có hai chiến lược được đưa ra để giải quyết bài toán này: Một là, chúng ta thử nhiều giá trị khởi đầu khác nhau, sau đó lựa chọn giải pháp có giá trị likelihood hội tụ lớn nhất. Hai là, sử dụng mô hình đơn giản hơn để xác định giá trị khởi đầu cho các mô hình phức tạp. Ý tưởng là: một mô hình đơn giản hơn sẽ giúp tìm được vùng tồn tại cực đại toàn cục, và ta bắt đầu bằng một giá trị trong vùng đó để tìm kiếm tối ưu chính xác khi sử dụng mô hình phức tạp hơn.

Thuật toán EM rất đơn giản, ít nhất là về mặt khái niệm. Nó được sử dụng hiệu quả nếu dữ liệu có tính phân cụm cao.

1.4.2. Học SVM truyền dẫn[13]

Phần này trình bày nội dung cơ bản của học quy nạp (inductive learning) và học truyền dẫn (transductive learning).

- ***Học quy nạp***

  Ta xem xét hàm ***f*** ánh xạ từ đầu vào x tới đầu ra y: $y = f(x)$ với ($y \in \{-1,1\}$).

  Học inductive sẽ dựa vào các dữ liệu huấn luyện có dạng $\{(x_i, y_i): i = 1,2, ..., n\}$

để tìm hàm ***f***. Sau đó, ta sẽ sử dụng hàm ***f*** để dự đoán nhãn y_{n+1} cho các mẫu chưa gán nhãn x_{n+1}. Các vấn đề của phương pháp:

- Khó tập hợp các dữ liệu gán nhãn .
- Lấy các mẫu dữ liệu chưa gán nhãn thì dễ dàng.
- Các mẫu cần phân lớp là biết trước.
- Không quan tâm đến hàm phân lớp ***f***.

Do vậy cần ứng dụng học theo kiểu truyền dẫn.

- ***Học truyền dẫn***

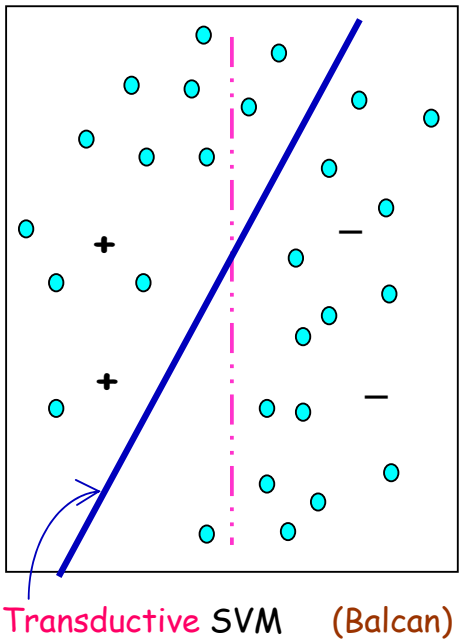
Học truyền dẫn được Vapnik đề cập từ năm 1998. Một bộ học được gọi là truyền dẫn nếu nó chỉ xử lý trên dữ liệu gán nhãn và dữ liệu chưa gán nhãn, và không thể xử lý dữ liệu mà nó chưa biết. Cho trước một tập các mẫu gán nhãn $\{(x_i, y_i): i = 1,2, ..., n\}$ và một tập các dữ liệu chưa gán nhãn $x_1', x_2', ..., x_m'$, mục đích của ta là tìm các nhãn $y_1', y_2', ..., y_m'$. Học truyền dẫn không cần thiết phải xây dựng hàm f, đầu ra của nó sẽ là một vector các nhãn lớp được xác định bằng việc chuyển thông tin từ dữ liệu gán nhãn sang dữ liệu chưa gán nhãn. Các phương pháp dựa trên đồ thị lúc đầu thường là truyền dẫn.

Phương pháp học TSVM:

Qui ước:

+, -: các mẫu âm, dương

o : các mẫu chưa gán nhãn

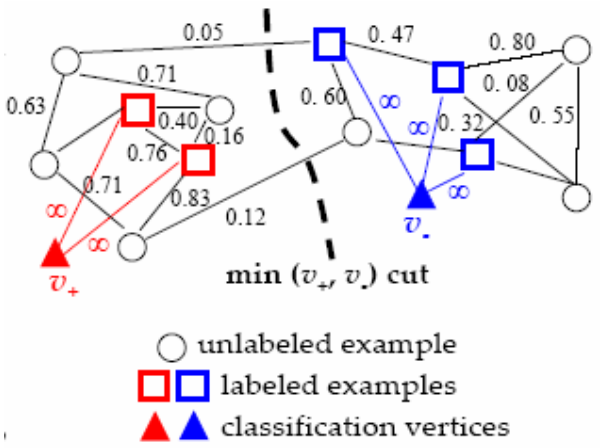


Hình 1. Siêu phẳng cực đại. Đường chấm chấm là kết quả của bộ phân lớp SVM quy nạp, đường liên tục chính là phân lớp SVM truyền dẫn

TSVM là một mở rộng của SVM chuẩn. Trong SVM chỉ có dữ liệu gán nhãn được sử dụng, mục đích là tìm siêu phẳng cực đại dựa trên các mẫu dữ liệu huấn luyện. Với TSVM, các điểm dữ liệu chưa gán nhãn cũng được sử dụng. Mục đích của TSVM là gán nhãn cho các điểm dữ liệu chưa gán nhãn để cho biên tuyến tính có lề phân cách là lớn nhất trên cả dữ liệu gán nhãn và dữ liệu chưa gán nhãn (xem hình 1).

1.4.3. Phân hoạch đồ thị quang phổ[12]

Phương pháp phân hoạch đồ thị quang phổ (Spectral Graph Partitioning) xây dựng một đồ thị có trọng số dựa trên các mẫu gán nhãn và các mẫu chưa gán nhãn. Trọng số của các cạnh tương ứng với một vài mối quan hệ giữa các mẫu như độ tương tự hoặc khoảng cách giữa các mẫu.



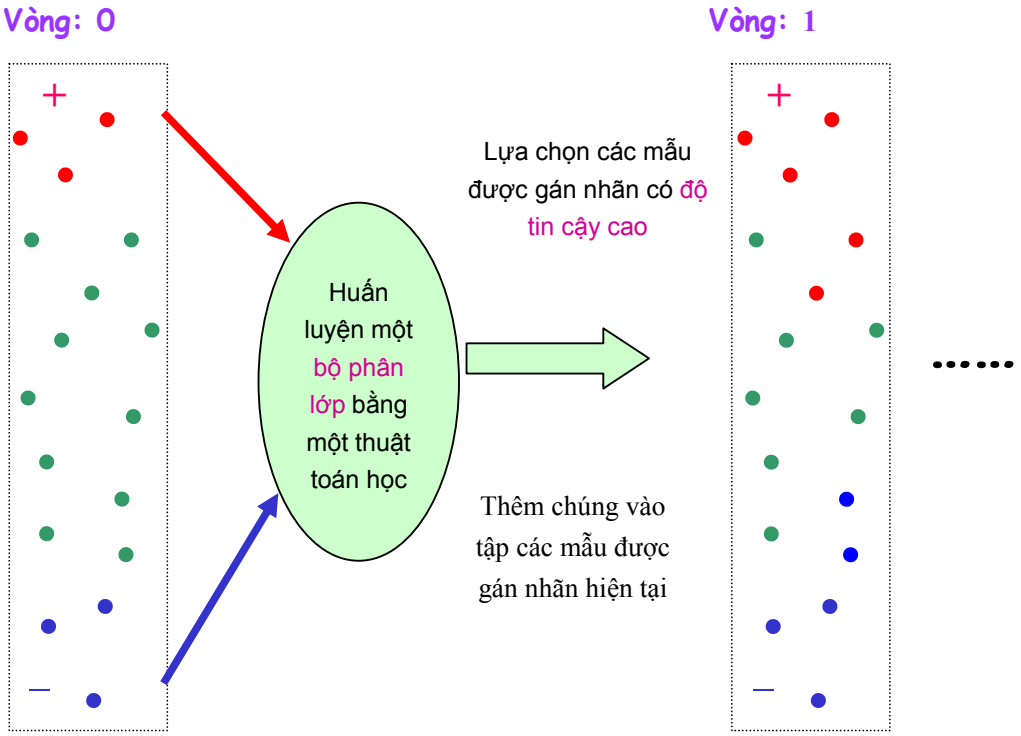
Hình 2. Đồ thị trọng số dựa trên các mẫu dữ liệu gán nhãn và dữ liệu chưa gán nhãn

Mục đích là tìm ra một nhát cắt cực tiểu (v_+, v_-) trên đồ thị (như hình 2). Sau đó, gán nhãn dương cho tất cả các mẫu chưa gán nhãn thuộc đồ thị con chứa v_+, và gán nhãn âm cho tất cả các mẫu chưa gán nhãn thuộc đồ thị con chứa v_-. Phương pháp này đưa ra một thuật toán có thời gian đa thức để tìm kiếm tối ưu toàn cục thực sự của nó.

CHƯƠNG 2 THUẬT TOÁN SELF-TRAINING VÀ CO-TRAINING

2.1. Thuật toán self-training

Có thể nói rằng, ý tưởng đầu tiên về sử dụng dữ liệu chưa gán nhãn trong phân lớp là thiết lập self-training. Ý tưởng về self-training xuất hiện từ những năm 1960. Đó là thuật toán bọc *(wrapper-algorithm)* sử dụng lặp nhiều lần một phương pháp học giám sát. Hình vẽ 3 biểu diễn một cái nhìn trực quan của thiết lập self-training.



Hình 3: Biểu diễn trực quan của thiết lập self-training

Self-training là kỹ thuật học bán giám sát được sử dụng rất phổ biến, với một bộ phân lớp (*classifier*) ban đầu được huấn luyện bằng một số lượng nhỏ các dữ liệu gán nhãn. Sau đó, sử dụng bộ phân lớp này để gán nhãn các dữ liệu chưa gán nhãn. Các dữ liệu được gán nhãn có độ tin cậy cao (vượt trên một ngưỡng nào đó) và nhãn tương ứng của chúng được đưa vào tập huấn luyện *(train set)*. Tiếp đó, bộ phân lớp được học

lại trên tập huấn luyện mới ấy và thủ tục lặp tiếp tục. Ở mỗi vòng lặp, bộ học sẽ chuyển một vài các mẫu có độ tin cậy cao nhất sang tập dữ liệu huấn luyện cùng với các dự đoán phân lớp của chúng. Tên gọi self-training xuất phát từ việc nó sử dụng dự đoán của chính nó để dạy chính nó. Sơ đồ thuật toán self-training được mô tả như hình 4.

Đặt

L : Tập các dữ liệu gán nhãn.

U : Tập các dữ liệu chưa gán nhãn

Lặp

- Huấn luyện bộ phân lớp h trên tập dữ liệu huấn luyện L.
- Sử dụng h để phân lớp dữ liệu trong tập U.
- Tìm tập con U' của U có độ tin cậy cao nhất.
- L + U' -> L
- U – U'-> U

Hình 4: Sơ đồ thuật toán self-training

Self-training đã được ứng dụng trong một vài nhiệm vụ xử lý ngôn ngữ tự nhiên: Riloff, Wiebe và Wilson (2003) [10] sử dụng self-training để xác định các danh từ có thuộc quan điểm cá nhân hay không... Self-training cũng được ứng dụng trong phân tích cú pháp và dịch máy.

2.2. Thuật toán co-training

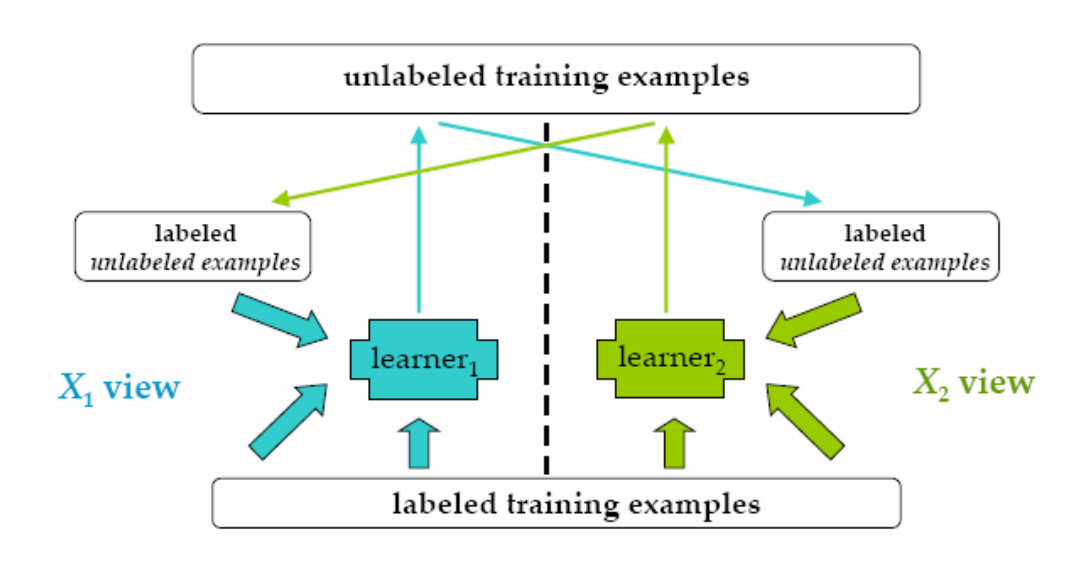
Thuật toán co-training dựa trên giả thiết rằng các *features* có thể được phân chia thành 2 tập con; Mỗi tập con phù hợp để huấn luyện một bộ phân lớp tốt. Hai tập con đó phải thoả *mãn tính chất độc lập điều kiện (conditional independent)* khi cho trước class. Thủ tục học được tiến hành như sau:

- Học 2 bộ phân lớp riêng rẽ bằng dữ liệu đã được gán nhãn trên hai tập thuộc tính con tương ứng.

- Mỗi bộ phân lớp sau đó lại phân lớp các dữ liệu *unlabel data*. Sau đó, chúng lựa chọn ra các *unlabeled data* + *nhãn dự đoán* của chúng (các examples có độ tin cậy cao) để dạy cho bộ phân lớp kia.
- Sau đó, mỗi bộ phân lớp được học lại *(re-train)* với các mẫu huấn luyện được cho bởi bộ phân lớp kia và tiến trình lặp bắt đầu.

Cái khó của co-training là ở chỗ: hai bộ phân lớp phải dự đoán trùng khớp trên dữ liệu chưa gán nhãn rộng lớn cũng như dữ liệu gán nhãn.

Những ý tưởng về sử dụng sự dư thừa *feature* đã được thi hành trong một vài nghiên cứu. Yarowsky đã sử dụng co-training để tìm nghĩa cho từ vựng, ví dụ quyết định xem từ "plant" trong một ngữ cảnh cho trước có nghĩa là một sinh vật sống hay là một xí nghiệp. Yarrowsky[8] tiến hành tìm nghĩa của từ bằng cách xây dựng một bộ phân lớp nghĩa *(sense classifier)* sử dụng ngữ cảnh địa phương của từ và một bộ phân lớp nghĩa dựa trên nghĩa của những lần xuất hiện khác trong cùng một văn bản; Riloff và Jones[9] phân lớp cụm danh từ chỉ vị trí địa lý bằng cách xem xét chính cụm danh từ đó và ngữ cảnh ngôn ngữ mà cụm danh từ đó xuất hiện; Collin và Singer[16] thực hiện phân lớp tên thực thể định danh sử dụng chính từ đó và ngữ cảnh mà từ đó xuất hiện. Sơ đồ co-training đã được sử dụng trong rất nhiều lĩnh vực như phân tích thống kê và xác định cụm danh từ. Hình vẽ 5 dưới đây cho chúng ta một cái nhìn trực quan của thiết lập co-training.



Hình 5: Sơ đồ biểu diễn trực quan thiết lập co-training

Blum và Mitchell [4] đã công thức hoá hai giả thiết của mô hình co-training và chứng minh tính đúng đắn của mô hình dựa trên thiết lập học giám sát theo mô hình PAC chuẩn. Cho trước một không gian các mẫu $X = X_1 \times X_2$, ở đây X_1 và X_2 tương ứng với hai khung nhìn *(views)* khác nhau của cùng một mẫu *(examples)*. Mỗi mẫu x vì vậy có thể được biểu diễn bởi một cặp (x_1, x_2). Chúng ta giả thiết rằng mỗi khung nhìn là phù hợp để phân lớp chính xác. Cụ thể, nếu $\mathcal{D}$ là một phân phối trên X, và C_1, C_2 là các lớp khái niệm *(concept classes)* được định nghĩa tương ứng trên X_1 và X_2; giả thiết rằng tất cả các nhãn trên các mẫu với xác suất lớn hơn không dưới phân phối $\mathcal{D}$ là trùng khớp với một hàm đích *(target function)* $f_1 \in C_1$, và cũng trùng khớp với hàm đích $f_2 \in C_2$. Nói cách khác, nếu f biểu diễn khái niệm đích kết hợp trên toàn bộ mẫu, thì với bất kỳ mẫu $x = x_1 \times x_2$ có nhãn l, ta có $f(x) = f_1(x_1) = f_2(x_2) = l$. Nghĩa là $\mathcal{D}$ gán xác suất bằng không mẫu (x_1, x_2) bất kỳ mà $f_1(x_1) \neq f_2(x_2)$.

- ***Giả thiết thứ nhất***: Tính tương thích *(compatibility)*

  Với một phân phối $\mathcal{D}$ cho trước trên X, ta nói rằng hàm đích $f = (f_1, f_2) \in C_1 \times C_2$ là tương thích *(compatible)* với $\mathcal{D}$ nếu thoả mãn điều kiện: $\mathcal{D}$ gán xác suất bằng không cho tập các mẫu (x_1, x_2) mà $f_1(x_1) \neq f_2(x_2)$. Nói cách khác, mức độ tương thích của một hàm đích $f = (f_1, f_2)$ với một phân phối $\mathcal{D}$ có thể được định nghĩa bằng một số $0 \leq p \leq 1$: $p = 1 - \Pr_D\left[(x_1, x_2) : f_1(x_1) \neq f_2(x_2)\right]$.

- ***Giả thiết thứ hai***: Độc lập điều kiện *(conditional independence assumption)*

  Ta nói rằng hàm đích f_1, f_2 và phân phối $\mathcal{D}$ thoả mãn giả thiết độc lập điều kiện nếu với bất kỳ một mẫu $(x_1, x_2) \in X$ với xác suất khác không thì,

$$\Pr_{(x_1,x_2)\in D}\left[\hat{x_1} = x_1 \middle| \hat{x_2} = x_2\right] = \Pr_{(x_1,x_2)\in D}\left[\hat{x_1} = x_1 \middle| f_2(\hat{x_2}) = f_2(x_2)\right]$$

  và tương tự,

$$\Pr_{(x_1,x_2)\in D}\left[x_2=\hat{x}_2 \,\middle|\, x_1=\hat{x}_1\right] = \Pr_{(x_1,x_2)\in D}\left[x_2=\hat{x}_2 \,\middle|\, f_1(x_1)=f_1\left(\hat{x}_1\right)\right]$$

Hai ông đã chỉ ra rằng, cho trước một giả thiết độc lập điều kiện trên phân phối D, nếu lớp đích có thể học được từ nhiễu phân lớp ngẫu nhiên theo mô hình PAC chuẩn, thì bất kỳ một bộ dự đoán yếu ban đầu nào cũng có thể được nâng lên một độ chính xác cao tuỳ ý mà chỉ sử dụng các mẫu chưa gán nhãn bằng thuật toán co-training. Hai ông cũng đã chứng minh tính đúng đắn của sơ đồ co-training bằng định lý sau:

Định lý (A.Blum & T. Mitchell).

Nếu C_2 có thể học được theo mô hình PAC với nhiễu phân lớp, và nếu giả thiết độc lập điều kiện thoả mãn, thì (C_1,C_2) có thể học được theo mô hình co-training chỉ từ dữ liệu chưa gán nhãn, khi cho trước một bộ dự đoán yếu nhưng hữu ích ban đầu $h(x_1)$.

Blum và Mitchell đã tiến hành thực nghiệm co-training trong phân lớp trang web theo sơ đồ trong hình 6 thể hiện rằng việc sử dụng dữ liệu chưa gán nhãn tạo ra một cải tiến quan trọng trong thực hành. Trong sơ đồ thiết lập trên, việc sử dụng U' sẽ tạo ra kết quả tốt hơn vì: Nó bắt buộc hai bộ phân lớp lựa chọn các mẫu có tính đại diện hơn cho phân phối $\mathcal{D}$ tạo ra tập U.

Cho trước:

- L là tập các mẫu huấn luyện đã gán nhãn.
- U là tập các mẫu chưa gán nhãn.

Tạo một tập U' gồm u mẫu được chọn ngẫu nhiên từ U

Lặp k vòng

- Sử dụng L huấn luyện bộ phân lớp h_1 trên phần x_1 của x.
- Sử dụng L huấn luyện bộ phân lớp h_2 trên
- phần x_2 của x.
- Cho h_1 gán nhãn p mẫu dương và n mẫu âm từ tập U'.
- Cho h_2 gán nhãn p mẫu dương và n mẫu âm từ tập U'.
- Thêm các mẫu tự gán nhãn này vào tập L.
- Chọn ngẫu nhiên $2p+2n$ mẫu từ tập U bổ sung vào tập U'.

Hình 6:Sơ đồ thiết lập co-training gốc cho vấn đề hai lớp

2.3. So sánh hai thuật toán

Bảng 1 đưa ra một số so sánh hai thiết lập self-training và co-training. Nói chung, sự khác nhau cơ bản giữa thuật toán self-training và co-training là ở chỗ: Self-training chỉ sử dụng một khung nhìn dữ liệu, trong khi đó co-training sử dụng hai khung nhìn dữ liệu. Self-training không yêu cầu sự phân chia của *features* thành hai khung nhìn độc lập như co-training. Nó chỉ cần một bộ phân lớp với một khung nhìn của dữ liệu.

Bảng 1. Bảng so sánh hai thiết lập self-training và co-training

Tiêu chí	Self-training	Co-training
Khung nhìn	1 khung nhìn	2 khung nhìn độc lập
Tình huống sử dụng	Khi bộ phân lớp cũ là khó chỉnh sửa	Thoả mãn thiết lập co-training
Ưu	Tận dụng nguồn dữ liệu chưa gán nhãn rất phong phú	
	Học tốt trong trường hợp các *features* không thể phân chia thành các views độc lập	Cho kết quả tốt nếu các giả thiết được thoả mãn Vì học trên 2 views dữ liệu nên chúng sẽ cung cấp nhiều thông tin hữu ích cho nhau hơn.
Nhược	- Khó khăn trong lựa chọn ngưỡng tin cậy của dự đoán (để làm giảm *noise* trong dự đoán). - Có thể có trường hợp có mẫu không được gán nhãn ➔ cần xác định số lần lặp để tránh lặp vô hạn.	
Khó khăn		Giả thiết độc lập điều kiện thường không đúng trong thực tế.

Co-training và self-training là hai thuật toán học bán giám sát có nhiệm vụ chính là mở rộng tập các mẫu gán nhãn ban đầu. Hiệu quả của thuật toán phụ thuộc vào chất lượng của các mẫu gán nhãn được thêm vào ở mỗi vòng lặp, được đo bởi hai tiêu chí:

- Độ chính xác của các mẫu được thêm vào đó.
- Thông tin hữu ích mà các mẫu mang lại cho bộ phân lớp.

Xem xét tiêu chí thứ nhất ta thấy, bộ phân lớp chứa càng nhiều thông tin thì độ tin cậy cho các dự đoán càng cao. Thuật toán co-training sử dụng hai khung nhìn khác nhau của một mẫu dữ liệu với giả thiết là mỗi khung nhìn là đầy đủ *(sufficient)* để dự đoán nhãn cho các mẫu dữ liệu mới. Tuy nhiên, giả thiết này là không thực tế bởi vì nhiều khi tập tất cả các *features* của một mẫu dữ liệu cũng chưa đủ để gán nhãn chúng một cách chính xác. Vì vậy, trong các ứng dụng thực, nếu xét theo tiêu chí này thì self-training thường có độ tin cậy cao hơn.

Với tiêu chí thứ hai, ta biết rằng thông tin mà mỗi mẫu dữ liệu gán nhãn mới đem lại thường là các *features* mới. Vì thuật toán co-training huấn luyện trên hai khung nhìn khác nhau nên nó sẽ hữu ích hơn trong việc cung cấp các thông tin mới cho nhau.

Việc lựa chọn các mẫu gán nhãn mới có độ tin cậy cao là một vấn đề hết sức quan trọng, vì nếu tiêu chí thứ nhất không được thoả mãn, các mẫu bị gán nhãn sai thì thông tin mới do chúng đem lại chẳng những không giúp ích được mà thậm chí còn làm giảm hiệu quả của thuật toán.

2.4. Các kỹ thuật làm trơn

Khi có một mẫu dữ liệu mới được thêm vào chúng ta phải xem xét 3 vấn đề:

1. Độ chính xác của mẫu đã gán nhãn được thêm vào từ dữ liệu chưa gán nhãn.Điều này cực kỳ quan trọng vì nếu chúng ta thêm vào một tập huấn luyện một lượng lớn các mẫu bị gán nhãn sai thì có thể làm cho bộ phân lớp trở nên tồi đi.
2. Tăng sự mất cân bằng dữ liệu huấn luyện: Nó sẽ hướng tới lựa chọn các mẫu của lớp thống trị với độ tin cậy cao và có thể những mẫu này được lựa chọn để thêm vào tập huấn luyện. Và như thế sẽ càng làm tăng tính mất cân bằng giữa các lớp.
3. Các thuật toán học bán giám sát sẽ dừng khi số vòng lặp đạt đến một giá trị định trước hoặc khi tập dữ liệu rỗng. Thông thường bộ phân lớp sau cùng sẽ được lựa chọn để xây dựng bộ phân lớp kết quả. Tuy nhiên không có bằng chứng chứng minh rằng bộ phân lớp này là tốt nhất. Câu hỏi đặt ra là làm

thế nào để lựa chọn được bộ phân lớp tốt nhất giữa các bộ phân lớp trung gian (generated classifier) hoặc có cách nào khác để lựa chọn bộ phân lớp cuối cùng là bộ phân lớp tốt nhất.

Xuất phát từ những đòi hỏi này, chúng tôi xin trình bày một số kỹ thuật làm trơn để nâng cao hiệu quả của thuật toán: Các kỹ thuật đảm bảo phân phối lớp và kết hợp các bộ phân lớp trung gian. Sau đây, chúng tôi sẽ trình bày chi tiết các kỹ thuật này.

2.4.1. Đảm bảo phân phối lớp

Việc đảm bảo phân phối lớp (class distribution) đã được đề xuất trong thuật toán co-training gốc (Blum and Mitchell, 1998) [2], bằng cách cố định số các mẫu gán nhãn được thêm vào ở mỗi vòng lặp. Tuy nhiên thuật toán này vẫn chưa giải quyết được vấn đề đó bởi vì vẫn có trường hợp tập các mẫu thêm vào nghiêng hẳn về một lớp nào đó.

Chúng tôi đề xuất một thủ tục để duy trì phân phối lớp của dữ liệu được gán nhãn. Thủ tục sẽ lựa chọn một tập con (subset) của một tập các mẫu được gán nhãn có độ tin cậy cao mà có thể duy trì được phân phối lớp ban đầu. Bởi vì các mẫu được thêm vào phải có đội tin cậy cao nên ta không thể luôn luôn thu được mẫu của tất cả các lớp ở mỗi vòng lặp. Vì vậy thủ tục phải đảm bảo các điều kiện sau.

- Duy trì phân phối lớp với sự chấp nhận lỗi.
- Có khả năng giải quyết trường hợp: Trong một tập các mẫu được gán nhãn

mới thì có thể có một lớp không có mẫu trong tập đó (trường hợp lớp rỗng).

Sơ đồ thủ tục được trình bày như ở hình 7. Đây là kỹ thuật nhằm làm tránh sự tăng nhảy vọt của một lớp sau mỗi vòng lặp.

Đầu vào:

S_N : Tập các mẫu gán nhãn mới;

S_{OL} : Tập các mẫu gán nhãn ban đầu;

D_{OC} : Phân phối lớp gốc; L : Tập các nhãn;

S_L : Tập các mẫu gán nhãn hiện tại;

Δ : Hằng số chấp nhận lỗi để duy trì phân phối lớp;

Đầu ra: Tập các mẫu gán nhãn mới được chọn từ S_N

Thuật toán:

Với mỗi $l \in L$, gọi $s_l(s_l^{'})$ là tập các mẫu trong $S_L(S_{OL})$ được gán nhãn l

While (*true*)

Tính phân phối lớp mới D_{NC} dựa vào tập các mẫu gán nhãn mới:

$$S = S_N \cup S_L$$

Với mỗi $l \in L$, gọi d_{ol} và d_{nl} là tỉ lệ của lớp l trong D_{OC} và D_{NC}

If (tồn tại một lớp $l \in L$ mà $d_{nl} - d_{ol} > \Delta$) then

Loại bỏ ngẫu nhiên (r+1) mẫu từ s_l sao cho

$$\frac{|s_l| + |s_l^{'}| - r}{|S| - r} = (d_{ol} + \Delta)$$

Else break;

***Hình7: SAE**: **S**elected**A**dded**E**xamples để lựa chọn các mẫu được gán nhãn mới mà vẫn đảm bảo được phân phối lớp ban đầu.*

2.4.2. Kết hợp bộ phân lớp

Một phương pháp để giải quyết vấn đề thứ 3 là liên kết các bộ phân lớp thành một bộ phân lớp duy nhất. Câu hỏi đặt ra là làm thế nào để tận dụng được lợi điểm của từng bộ phân lớp. Chúng ta nghiên cứu một vài chiến lược được sử dụng cho tìm nghĩa của từ (WSD – Word Sense Disambiguation) như đã trình bày trong [12].

Đặt $D = \{D_1,...,D_R\}$ là tập gồm R bộ phân lớp

$\Omega = \{\omega_1,...,\omega_c\}$ là tập các nhãn lớp khác nhau

Đầu ra của 1 bộ phân lớp được định nghĩa là một vector c-chiều

$$D_i(w) = [d_{i,1}(w),...,d_{i,c}(w)]$$

ở đây $d_{i,j}(w)$ là mức độ hỗ trợ mà bộ phân lớp D_i gán w vào lớp w_j.

Liên kết các bộ phân lớp nghĩa là tìm nhãn lớp cho w dựa trên đầu ra của R bộ phân lớp. Khi đó, đầu ra của bộ phân lớp cuối cùng sẽ là:

$$D(w) = [\mu_1(w),...,\mu_c(w)]$$

ta gán nhãn w vào lớp w_s nếu

$$\mu_s(w) \geq \mu_t(w), \forall t = 1,...,c$$

Ở đây, chúng ta sử dụng chiến lược phân lớp sau:

Median Rule: $k = \arg\max_j \left[\frac{1}{R}\sum_{i=1}^{R} d_{i,j}(w)\right]$

Max Rule: $k = \arg\max_j \left[\max_{i=1}^{R} d_{i,j}(w)\right]$

Min Rule: $k = \arg\max_j \left[\min_{i=1}^{R} d_{i,j}(w)\right]$

Majority Voting $k = \arg\max_j \sum_i \delta_{i,j}(w),$

trong đó:

$$\delta_{i,j}(w) = \begin{cases} 1, & \text{if } j = \underset{m}{\operatorname{argmax}}\, d_{i,m}(w) \\ 0, & \text{otherwise} \end{cases}$$

Những quy tắc này cung cấp các cách khác nhau để liên kết các bộ phân lớp riêng biệt: *median rule*: Lấy trung bình tất cả các bộ phân lớp, *max rule*: mức độ hỗ trợ *(support degree)* cho mỗi lớp được quyết định bởi mức độ hỗ trợ cao nhất trong các bộ phân lớp; *min rule*: Xác định mức độ hỗ trợ cho mỗi lớp bằng mức hỗ trợ thoả mãn tất cả các bộ phân lớp; *Majority Voting* lựa chọn lớp được hỗ trợ bởi nhiều bộ phân lớp nhất.

2.4.3. Thuật toán self-training và co-training với các kỹ thuật làm trơn

Sau đây, chúng tôi đề xuất một mô hình cải tiến thuật toán self-training và co-training bằng hai kỹ thuật làm trơn được trình bày ở trên.

Vì hai bộ phân lớp dựa trên hai views của dữ liệu có thể dự đoán nhãn khác nhau cho cùng một mẫu dữ liệu, nên ta sẽ sử dụng một trong số các chiến lược liên kết các bộ phân lớp. Việc kết hợp này hy vọng sẽ thu được kết quả tốt hơn từng bộ phân lớp. Ở đây ta có thể sử dụng các chiến lược như *max, min, median* ... như đã trình bày trong [12].

Thuật toán co-training được mô tả bằng sơ đồ trong hình 8.

Sự khác biệt khi thi hành thuật toán self-training và co-training chỉ là ở chỗ co-training sử dụng hai khung nhìn dữ liệu trong khi đó self-training sử dụng một khung nhìn dữ liệu. Thuật toán self-training có thể thu được từ sơ đồ thuật toán co-training trên đây bằng cách thay thế bước 2,3,4,5 bởi hai bước sau:

- Sử dụng L trên khung nhìn V và thuật toán H để huấn luyện bộ phân lớp h_1.
- Sử dụng h_1 gán nhãn các mẫu trong U' và lựa chọn các mẫu có độ tin cậy vượt trên ngưỡng θ.

Đầu vào:

L: Tập các mẫu huấn luyện đã gán nhãn;

U: Tập các mẫu chưa gán nhãn;

H: Thuật toán học giám sát cơ bản;

θ: Ngưỡng tin cậy để lựa chọn một mẫu mới;

C: Một chiến lược liên kết các bộ phân lớp;

Thuật toán:

Lặp k vòng lặp:

Begin

1. Tạo tập U' bằng cách lấy ngẫu nhiên u mẫu từ U để $|U'| = u$
2. Sử dụng L trên *view1* và thuật toán H để huấn luyện bộ phân lớp h_1
3. Sử dụng L trên *view2* và thuật toán H để huấn luyện bộ phân lớp h_2
4. Dùng h_1 gán nhãn các mẫu trong U' và chọn các mẫu được gán nhãn mới có độ tin cậy lớn hơn ngưỡng θ
5. Dùng h_2 gán nhãn các mẫu trong U' và chọn các mẫu được gán nhãn mới có độ tin cậy lớn hơn ngưỡng θ

   Gọi tập các mẫu gán nhãn mới vừa thu được là S_L

6. Gọi thủ tục SAE với đầu vào S_L, đầu ra của thủ tục này S_{NL} gồm tập các mẫu gán nhãn được chọn mới.
7. Thêm các mẫu tự gán nhãn S_{NL} này vào tập L
8. Loại bỏ khỏi U' tập các mẫu chưa gán nhãn tương ứng với các mẫu trong S_{NL}

Hình 8: Thuật toán co-training mới với thủ tục duy trì phân phối lớp và liên kết các bộ phân lớp.

Chương 3 THỰC NGHIỆM TRONG BÀI TOÁN PHÂN LỚP VĂN BẢN

3.1. Giới thiệu bài toán thực nghiệm

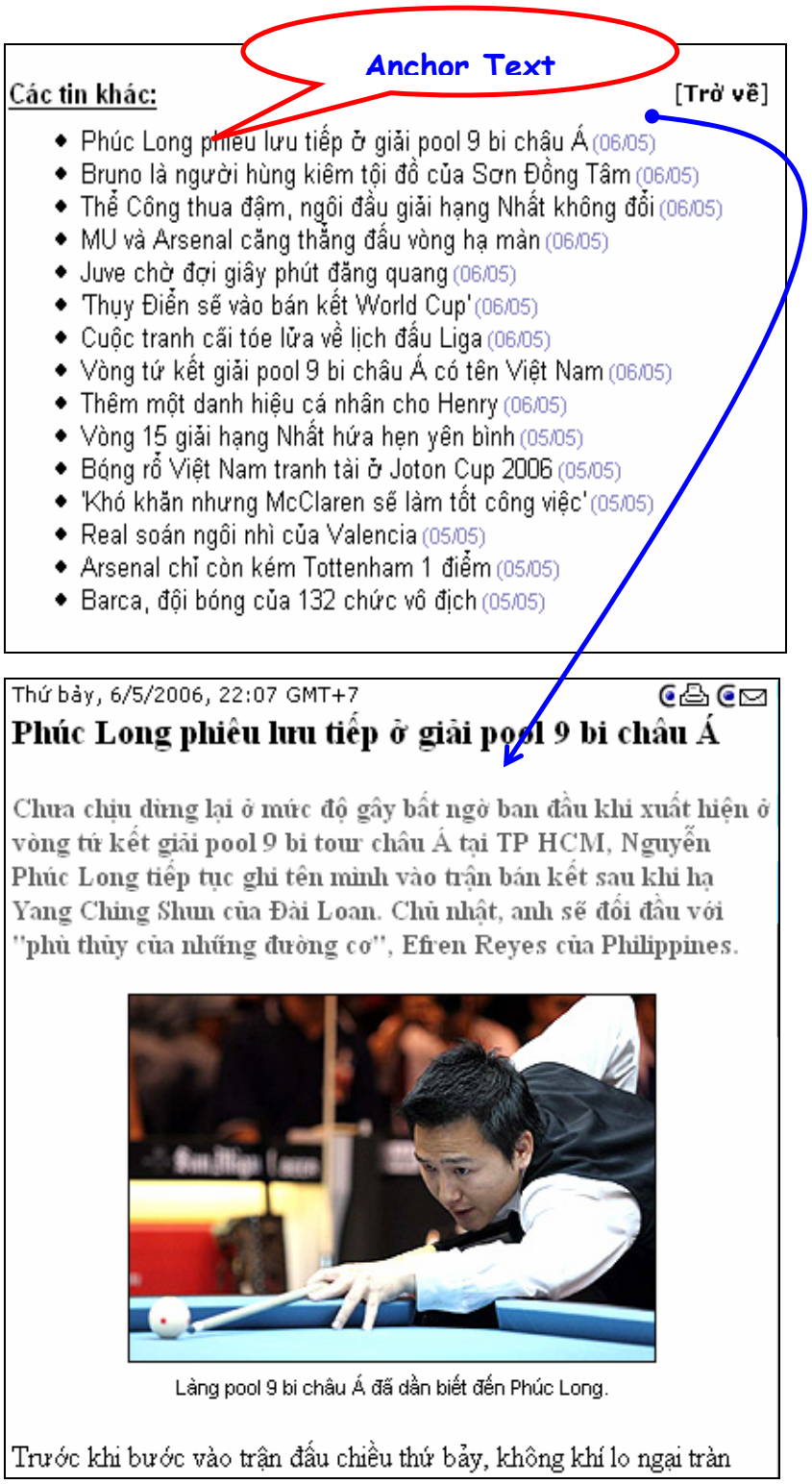
Phân lớp văn bản hiện nay là một chủ đề giành được nhiều sự quan tâm. Đây cũng chính là một trong những động lực thúc đẩy sự phát triển các phương pháp học bán giám sát. Trong thực tế, tồn tại một số lượng lớn các trang web chưa được gán nhãn, ta có thể dễ dàng thu được chỉ bằng một bộ web crawler.

Trong luận văn này chúng tôi tiến hành ứng dụng hai thuật toán học bán giám sát self-training và co-training trong bài toán phân lớp trang Web bởi các lý do sau:

- Việc ứng dụng trong phân lớp văn bản có một đặc điểm hấp dẫn: Mỗi mẫu có thể được mô tả sử dụng các "loại" thông tin khác nhau (different "kinds" of information). Loại thông tin đầu tiên là *text* xuất hiện trong chính trang web đó. Loại thông tin thứ hai là *anchor text* gắn với các *hyperlink* trỏ tới các trang web này từ các trang web khác.
- Chúng ta có thể giả thiết rằng đối với mỗi một trang P thì các từ trên trang P và các từ trong *hyperlinks* trỏ tới trang P đó là độc lập điều kiện khi cho trước phân lớp của P. Đây là điểm bắt đầu hợp lý vì trang web thường được xây dựng bởi người dùng chứ không phải là người tạo ra các liên kết *(links)*.

Với hai đặc điểm trên, ta có thể tiến hành học theo thuật toán co-training với hai khung nhìn là: view #1 (page-based)-các từ trên trang web; view #2 (hyperlink-based) -các từ xuất hiện trong các *hyperlinks* trỏ tới trang web đó

Một trang web có thể được phân lớp dựa trên các từ xuất hiện trong trang web hoặc các từ xuất hiện trong các siêu liên kết trỏ tới trang web đó. Do đó, ta huấn luyện hai bộ phân lớp tương ứng trên hai khung nhìn đó. Thêm vào đó, ta xác định một bộ phân lớp liên kết để trộn kết quả đầu ra của hai bộ phân lớp này. Hình 9 dưới đây cho một ví dụ về hai khung nhìn của một trang web.



Hình 9: Hai khung nhìn của một trang web

3.2. Các lớp văn bản

Hệ thống phân lớp nội dung Web của khoá luận được xây dựng dựa trên cây phân lớp tin tức của Báo điện tử VnExpress (http://vnexpress.net) của công ty truyền thông FPT. Chúng tôi lựa chọn các phân lớp sau từ cây phân lớp của VnExpress: Vi tính, Phương tiện, Sức khoẻ, Thể thao, Pháp luật, Văn hoá. Việc chúng tôi quyết định lựa chọn các phân lớp này là vì những phân lớp này có các đặc trưng có tính chuyên biệt cao. Bảng 2 mô tả nội dung liên quan đến từng lớp.

Bảng 2. Bảng mô tả các phân lớp

STT	Tên phân lớp	Vnexpress	Mô tả các nội dung liên quan
1	Công nghệ	Vi tính	Công nghệ thông tin và truyền thông
2	Pháp luật	Pháp luật	Các vụ án, vụ việc, các văn bản mới, ...
3	Phương tiện	Ôtô – Xe máy	Chủ yếu là giới thiệu các loại ôtô, xe máy mới
4	Sức khoẻ	Sức khoẻ	Sức khoẻ, giới tính, chăm sóc sắc đẹp, ...
5	Thể thao	Thể thao	Bóng đá, tennis, ...; các cầu thủ, trận đấu, ...
6	Văn hoá	Văn hoá	Âm nhạc, thời trang, điện ảnh, mỹ thuật, ...

3.3. Môi trường thực nghiệm

3.3.1. Môi trường phần cứng

Toàn bộ thực nghiệm được tiến hành trên cấu hình máy liệt kê ở bảng 3.

Bảng 3: Cấu hình máy tính

Thành phần	Chỉ số
CPU	PIV, 2.26GHz
RAM	384 MB
OS	Linux Fedora 2.6.11

3.3.2. Công cụ phần mềm.

Khoá luận sử dụng một số công cụ phần mềm hỗ trợ trong quá trình thực nghiệm như liệt kê trong bảng 4.

Bảng 4: Bảng công cụ phần mềm hỗ trợ

STT	Tên công cụ	Mô tả	Tác giả	Nguồn
1	HTML Parser	Bộ phân tích HTML	Jose Solorzano	http://jexpert.us
2	html2text.php	Công cụ lọc nhiễu theo từng trang web cụ thể cho toàn bộ các file .html.	Nguyễn Việt Cường – K46CA	http://203.113.130.205/~cuongnv/thesis/code/tools.tar.gz
3	text2telex.php	Công cụ chuyển văn bản bị mã hoá unicode tiếng Việt sang định dạng tiếng Việt kiểu telex cho toàn bộ các file mà html2text sinh ra.		

Ngoài ra, trong quá trình chuẩn bị dữ liệu, chúng tôi viết một số công cụ chạy trên nền Linux và Win với bộ biên dịch tổng hợp GNU GCC và bộ thông dịch PHP như liệt kê trong bảng 5.

Bảng 5: Bảng công cụ phần mềm xử lý dữ liệu

STT	Tên công cụ	Mô tả
1	reject_stop_word.php	Công cụ loại bỏ các từ dừng của một văn bản sau khi đã đưa về dạng telex
2	format_feature.php	Công cụ thống kê trong mỗi văn bản thì một từ xuất hiện bao nhiêu lần
3	text2telex.php	Công cụ chuyển văn bản bị mã hoá unicode tiếng Việt sang định dạng tiếng Việt kiểu telex cho toán bộ các file mà html2text sinh ra.
4	get_AnchorText.php	Công cụ dùng để lấy các AnchorText của một trang web

Việc cài đặt thuật toán, chúng tôi sử dụng một số lớp sau được liệt kê trong bảng 6:

Bảng 6: Bảng các lớp thực hiện học bán giám sát

STT	Tên lớp	Mô tả
1	BigNumber.h, BigNumber.cpp	Thực hiện các phép tính toán với số lớn có chiều dài tuỳ ý
2	KeyWord.h, KeyWord.cpp	Lưu trữ KeyWord của từng lớp theo dạng từ thứ i xuất hiện trong lớp j bao nhiêu lần
3	Lib.h, Lib.cpp	Một số hàm phục vụ cho các lớp
4	get_AnchorText.php	Công cụ dùng để lấy các AnchorText của một trang web
5	Random_Division.h, Random_Division.cpp	Phân chia ngẫu nhiên văn bản các lớp vào các tập test, train và tập chưa gán nhãn
6	Random_file.h, Random_file.cpp	Tạo ra một bể U' từ một tập các văn bản chưa gán nhãn
7	Processing_pool.h Processing_pool.cpp	Xử lý bể U' vừa tạo ra: gán nhãn lớp, lấy các mẫu tin cậy vào tập huấn luyện và thực hiện thủ tục SAE với các lớp có độ chênh lệch phân phối vượt quá tham số Δ
8	Test.h, Test.cpp	Từ thông tin KeyWord có được sau một số vòng lặp, thực hiện gán nhãn cho tập kiểm thử
9	Main.cpp	Chương trình chính thực hiện các thuật toán bootstrapping
10	Improve.h, Improve.cpp	Thực hiện các thủ tục cải tiến và kết hợp các bộ phân lớp.

3.4. Bộ dữ liệu thực nghiệm

Với mỗi phân lớp được lấy từ trang tin điện tử Vnexpress, chúng tôi lựa chọn mỗi lớp là 140 tin. Sau đó tiến hành phân chia tập dữ liệu đó như sau:

- Tập dữ liệu huấn luyện ban đầu: Mỗi lớp lấy 20 tin làm dữ liệu huấn luyện mô hình ban đầu.
- Tập dữ liệu kiểm tra: Mỗi lớp lấy 20 tin để làm dữ liệu kiểm tra.
- Còn lại 100 tin mỗi lớp đưa vào tập dữ liệu chưa gán nhãn rồi trộn đều. Việc lấy số lượng dữ liệu chưa gán nhãn bằng nhau cho mỗi lớp nhằm đảm bảo tính phân phối đồng đều và ngẫu nhiên (thoả mãn điều kiện i.i.d).

3.5. Quá trình tiến hành thực nghiệm

3.5.1. Xây dựng các đặc trưng

Sau khi thu được các trang web ở dạng html, chúng tôi tiến hành trích chọn các *anchor text* tương ứng cho từng trang web đó. Để việc xử lý các từ tiếng Việt được thuận tiện và dễ dàng, chúng ta sẽ biến đổi các từ tương ứng sang dạng chỉ gồm các ký hiệu trong bảng chữ cái và chữ số. Điều này được thực hiện bằng công cụ text2telex.php.

Các dữ liệu *text* và *anchor text* này sẽ được xử lý để loại bỏ các từ dừng để việc lựa chọn các đặc trưng cho từng lớp có tính chuyên biệt cao.

Các đặc trưng của văn bản quyết định phân lớp của văn bản đó. Trong phân lớp văn bản thì các đặc trưng của văn bản chính là các từ xuất hiện trong các văn bản đó. Việc xây dựng các đặc trưng dựa trên các mệnh đề mô tả thông tin ngữ cảnh. Trong khoá luận này chúng tôi sử dụng cấu trúc n-grams, với n = 1, 2, 3 vì thực tế với các giá trị trên của n là chúng ta có thể đủ để bao quát các thông tin ngữ cảnh đối với bài toán phân lớp văn bản tiếng Việt.

Chúng tôi tiến hành xây dựng các n-gram như sau:

- Đầu tiên, chúng ta tiến hành loại bỏ các từ dừng trong các văn bản: Đối với tiếng Việt do chưa có một danh sách các từ dừng chuẩn nên việc loại bỏ các từ dừng chỉ là tương đối theo một danh sách các từ dừng tiếng Việt do chúng tôi tự thiết kế.

- Sau đó, chúng ta tiến hành xây dựng n-gram: Xét ví dụ với mệnh đề thông tin ngữ cảnh là *"dự báo công nghệ thông tin Việt Nam năm 2005"* thì danh sách các n-gram là:

n-gram	Kết quả
1-gram	dự, báo, công, nghệ, thông, tin, Việt, Nam, năm, 2005
2-gram	dự_báo, báo_công, công_nghệ, nghệ_thông, thông_tin, tin_Việt, Việt_Nam, Nam_năm, năm_2005.
3-gram	dự_báo_công, báo_công_nghệ, công_nghệ_thông, nghệ_thông_tin, thông_tin_Việt, tin_Việt_Nam, Việt_Nam_năm, Nam_năm_2005.

Bảng 7: Danh sách các n-gram

Với các n-gram được sinh ra như trên (xem bảng 7), chúng tôi tiến hành xây dựng các mệnh đề thông tin ngữ cảnh như sau, ví dụ một mệnh đề chỉ ra văn bản thứ di có chứa cụm từ wt nào đó n lần:

[<di> chứa <wt>: n lần]

Do thuật toán học bán giám sát self-training và co-training là một tiến trình lặp nên việc thu được từng đặc trưng trong một văn bản mới là rất có ý nghĩa. Do vậy, chúng tôi quyết định lựa chọn tất cả các đặc trưng để tiến hành phân lớp mà không loại bỏ một đặc trưng nào cả.

3.5.2. Thiết lập tham số cho mô hình

Các tham số cho mô hình được thiết lập như sau:

|U'| = 150, số vòng lặp bằng 10, tham số chấp nhận lỗi $\Delta = 0.03$.

Số mẫu thêm vào sau mỗi vòng lặp: numOfAdded = 15.

Do hai bộ phân lớp trên hai *view* dữ liệu có thể dự đoán không trùng khớp cho cùng một mẫu dữ liệu (và thực tế thực nghiệm trên khung nhìn *anchor text* chúng tôi

thấy rằng bộ phân lớp anchor text) cho nên trong dự đoán mỗi lớp ta định nghĩa một bộ phân lớp liên kết.

$$P(c_j|x) = P(c_j|x_1)P(c_j|x_2) \quad (11)$$

3.6. Kết quả của các bộ phân lớp

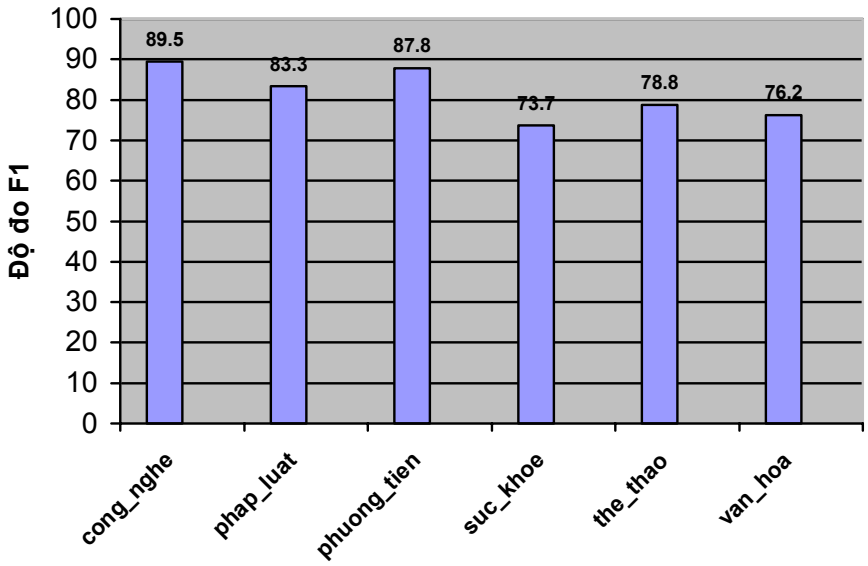
- Bộ phân lớp giám sát Naïve Bayes dựa trên nội dung của một tài liệu: Bảng 8 biểu diễn kết quả bộ phân lớp này với các độ đo: Độ chính xác, độ hồi tưởng, độ đo F1.

Bảng 8: Các độ đo của bộ phân lớp giám sát Naïve Bayes dưa trên content

	Độ chính xác	**Độ hồi tưởng**	**Độ đo F1**
cong_nghe	0.944	0.85	0.895
phap_luat	0.714	1	0.833
phuong_tien	0.857	0.9	0.878
suc_khoe	0.778	0.7	0.737
the_thao	1	0.65	0.788
van_hoa	0.727	0.8	0.762
Trung bình	**0.837**	**0.817**	**0.815**

Dựa vào kết quả ở bảng 8, ta thấy các độ đo của bộ phân lớp giám sát Naïve Bayes là khá cao. Độ đo F1 trong trường hợp cao nhất lên đến 89.5%. Do đó, ta hoàn toàn có thể tin cậy vào dự đoán của bộ phân lớp này để tiến hành các bước lặp self-training.

Từ bảng kết quả đó, chúng ta biểu diễn đồ thị độ đo F1 đối với từng lớp như hình 10.



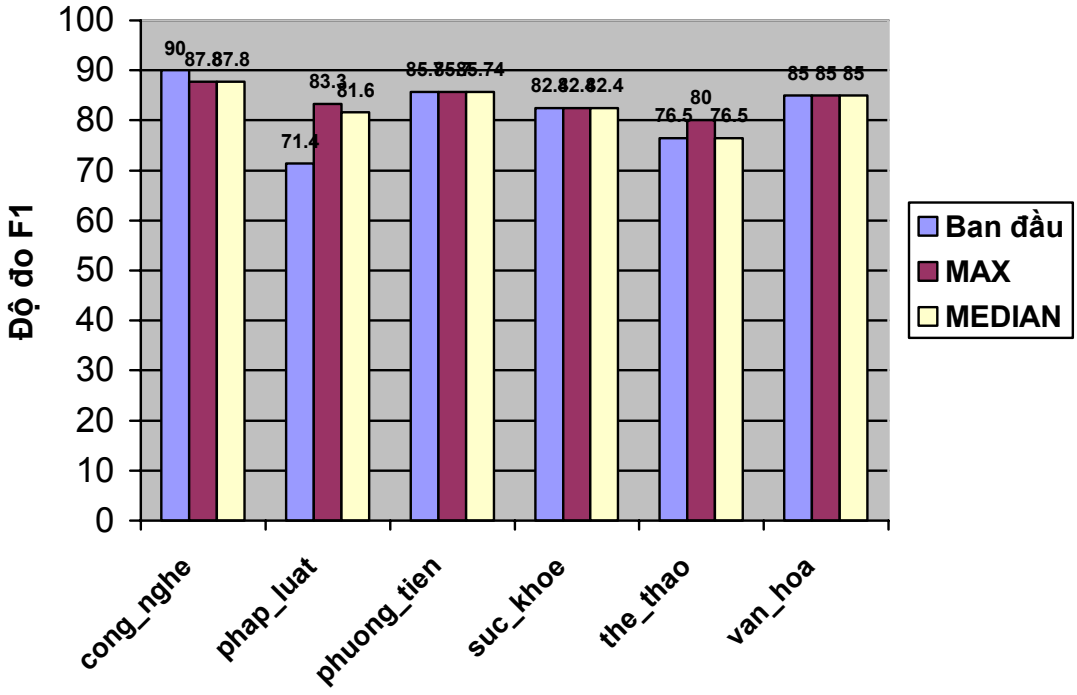
Hình 10: Đồ thị biểu diễn độ đo F1 của bộ phân lớp giám sát Naïve Bayes dựa trên content

- Bộ phân lớp bán giám sát self-training gốc và self-training cải tiến dựa trên nội dung của một văn bản: Các độ đo được liệt kê ở bảng 9.

Bảng 9: Các độ đo của self-training (ban đầu/cải tiến MAX/ cải tiến MEDIAN) dựa trên content.

	Ban đầu			**MAX**			**MEDIAN**		
	Precision	***Recall***	***F1***	***Precision***	***Recall***	***F1***	***Precision***	***Recall***	***F1***
cong_nghe	0.9	0.9	0.9	0.858	0.9	0.878	0.857	0.9	0.878
phap_luat	0.667	1	0.8	0.714	1	0.833	0.69	1	0.816
phuong_tien	0.818	0.9	0.857	0.818	0.9	0.857	0.818	0.9	0.857
suc_khoe	1	0.7	0.824	1	0.7	0.824	1	0.7	0.824
the_thao	0.929	0.65	0.765	0.933	0.7	0.8	0.929	0.65	0.765
van_hoa	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
Trung bình	**0.86**	**0.83**	**0.833**	**0.862**	**0.842**	**0.840**	**0.857**	**0.833**	**0.832**

Từ bảng các độ đo kết quả, ta biểu diễn đồ thị độ đo F1 trung bình của các bộ phân lớp bán giám sát self-training (ban đầu/ MAX/ MEDIAN) như hình vẽ 11.



Hình 11: Đồ thị biểu diễn độ đo F1 của bộ phân lớp bán giám sát self-training gốc và self-training cải tiến

3.7. Một số nhận xét kết quả đạt được

Từ kết quả thu được ở trên chúng tôi có một số nhận xét sau:

- Self-training đã nâng độ chính xác so với các thuật toán học giám sát thông thường: Độ đo F1 trung bình trong trường hợp học giám sát là 81.5%, trong khi đó độ đo F1 trung bình trong trường hợp học bán giám sát self-training ban đầu là 83.3%, self-training với qui tắc làm trơn MAX là 84%, self-training với qui tắc làm trơn MEDIAN là 83.2%

- Từ đó, chúng tôi nhận thấy việc áp dụng các qui tắc làm trơn được đề xuất trong khoá luận này thực sự đã đem lại hiệu quả trong trường hợp của bài toán phân lớp văn bản này.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Khoá luận đã nghiên cứu và tìm hiểu một số thuật toán học bán giám sát trong đó đặc biệt chú trọng xem xét, đánh giá hai thuật toán học bán giám sát self-training và co-training. Khóa luận đã đạt được một số kết quả như sau:

- Về lý thuyết, đã tìm hiểu được chứng minh tính đúng đắn của thiết lập co-training dựa trên một số giả thiết của Blum và Mitchel [2]. Nhằm cải tiến hướng mục tiêu thu nhận được dự đoán phân lớp chính xác hơn, khóa luận đã đề xuất việc duy trì phân phối lớp với sự chấp nhận lỗi Δ, kết hợp các bộ phân lớp trung gian cho thuật toán học self-training và co-training nhằm tận dụng lợi điểm của các bộ phân lớp trung gian được tạo ra.
- Về thực nghiệm, đã tiến hành và thử nghiệm các thực nghiệm về self-training trên bài toán phân lớp trang web tiếng Việt và thu được kết quả khả quan: độ đo F1 trung bình của thuật toán bán giám sát self-training so với trường hợp chỉ học giám sát trên mẫu dữ liệu huấn luyện ít ỏi tăng lên từ 81.5% lên 83%; độ đo F1 trung bình sau khi thực hiện các cải tiến duy trì phân phối lớp và áp dụng qui tắc kết hợp bộ phân lớp MAX tăng lên là 84%

Do còn nhiều hạn chế về thời gian và kiến thức, trong khoá luận này còn một số vấn đề phải tiếp tục hoàn thiện và phát triển trong thời gian tới:

- Xây dựng danh sách hoàn thiện các từ dừng tiếng Việt nhằm loại bỏ nhiễu trong quá trình dự đoán phân lớp.
- Tiếp tục tiến hành thử nghiệm co-training trên dữ liệu thực.
- Thực hiện thử nghiệm trên số lượng lớn hơn các trang web chưa gán nhãn.
- Thử nghiệm thêm ý tưởng với giả thiết có 2 thuật toán supervised learning A và B, ta tạo được 2 bộ phân lớp Ca và Cb - đoán nhận nhãn (class) cho một example e, thông thường thì ta dùng mình Ca, với xác xuất đoán nhận nhãn La > ngưỡng, chẳng hạn 0.9 thì ta chấp nhận nhãn đấy. Bây giờ ta dùng thêm Cb đoán nhận được nhãn Lb, nếu La = Lb thì ta mới chấp nhận đưa vào tập labeled data. Khi đó ta có thể giảm ngưỡng La là 0.8 chẳng hạn, ngưỡng cho Lb cũng 0.8 chẳng hạn mà kết quả chính xác hơn và nhận dạng được nhiều mẫu hơn.

Tài liệu tham khảo

Tiếng Việt

[1]. Nguyễn Việt Cường, Bài toán lọc và phân lớp nội dung Web tiếng Việt theo hướng tiếp cận entropy cực đại. *Khóa luận tốt nghiệp đại học 2005, Đại học Công nghệ - Đại học Quốc gia Hà Nội.*

[2]. Đặng Thanh Hải,Thuật toán phân lớp văn bản Web và thực nghiệm trong máy tìm kiếm VietSeek. *Khóa luận văn tốt nghiệp đại học 2004, Đại học Công nghệ - Đại học Quốc gia Hà Nội.*

Tiếng Anh

[3]. Andrew McCallum, Kamal Nigam, A Comparison of Event Model for Naive Bayes Text Classification, *Working Notes of the 1998 AAAI/ICML Workshop on Learning for Text Categorization, 1998*.

[4]. Avrim Blum and Tom Mitchell, Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT-98), 1998*.

[5]. A. P. Dempster, N. M. Laird, and D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):138, 1977.

[6]. Chapelle, O., Zien, A., & Sch¨olkopf, B. (Eds.), Semi supervised learning. *MIT Press, 2006.*

[7]. Cozman, F., Cohen, I., & Cirelo, M., Semi-supervised learning of mixture models. *ICML-03, 20th International Conference on Machine Learning, 2003*.

[8]. David Yarrowsky, Unsupervised Word Sense Disambiguation Rivaling Supervised Methods, In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, 189-196*.

[9]. E. Riloff and R. Jones, Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping.In *Proceedings of the 16th National Conference on Artificial Intelligence, 1999*.

[10]. Ellen Rillof, Janyce Wiebe, Theresa Wilson, Learning Subjective Nouns using Extraction Pattern Bootstrapping. *2003 Conference on Natural Language Learning (CoNLL-03), ACL SIGNLL, 2003.*

[11]. F. G. Cozman, and I. Cohen, "Unlabeled data can degrade classification performance of generative classifiers," *Int'l Florida Artificial Intell. Society Conf.*, 327-331, 2002.

[12]. Joachims, T. Transductive learning via spectral graph partitioning. In *Proceeding of. The Twentieth International Conference on Machine Learning (ICML2003), 290-297, 2003*.

[13]. Joachims T., Transductive Inference for Text Classification using Support Vector Machines. *International Conference on Machine Learning (ICML), 1999*

[14]. Le C. A., Huynh V. N., and Shimazu A., Combining Classifiers with Multi-Representation of Context in Word Sense Disambiguation. In *Proc. PAKDD*, 262–268, 2005.

[15]. McCallum, A. and Nigam K. "A Comparison of Event Models for Naive Bayes Text classification". *In AAAI/ICML-98 Workshop on Learning for Text Categorization, pp. 41-48. Technical Report WS-98-05. AAAI Press. 1998.*

[16]. Michael Collins and Yoram Singer, Unsupervised Model for Name Entity Recognition, In *EMNLP*.

[17]. Michael Thelen and Ellen Riloff, A bootstrapping method for Learning Semantic Lexicons using Extraction Pattern Contexts. *2002 Conf. on Empirical Methods in Natural Language Processing, Philadelphia, PA, July 2002, 214-221.*

[18]. Nigam, K., Ghani, R., Analyzing the effectiveness and applicability of cotraining. *In Proceedings of Ninth International Conference on Information and Knowledge Management (CIKM-2000), 86–93, 2000.*

[19]. Nigam, K., Ghani, R., Understanding the behavior of co-training. *In Proceedings of KDD-2000 Workshop on Text Mining,.2000*.

[20]. Nigam K., McCallum A., Thrun S., Mitchell T. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, *39(2/3):103-134*, *2000.*

[21]. Rosie Jones, Andrew McCallum, Kamal Nigam, Ellen Rillof, Bootstrapping for text learning Tasks, *IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications*, 1999.

[22]. Susana Eyheramendy, David D. Lewis, David Madigan, On the Naive Bayes Model for Text Classification, to appear in *Artificial Intelligence & Statistics 2003*.

[23]. Xiaojin Zhu, Semi-Supervised Learning Literature Survey. *Computer Sciences TR 1530, University of Wisconsin – Madison, February 22, 2006*.

[24]. http://en.wikipedia.org/wiki/