PHƯƠNG PHÁP PHÂN LỚP SỬ DỤNG MÁY VEC-TƠ HỖ TRỢ ỨNG DỤNG TRONG TIN SINH HỌC

Classification Using Support Vector Machines and Its Applications in Bioinformatics

Nguyễn Thị Thảo, Nguyễn Thị Huyền, Đoàn Thị Thu Hà
Trần Thị Thu Huyền, Nguyễn Thị Thủy

Khoa Công nghệ Thông tin, Trường Đại học Nông nghiệp Hà Nội
Đại chỉ email tác giả liên lạc: ntthao81@hua.edu.vn
Ngày gửi đăng: 29.08.2011; Ngày chấp nhận 20.10.2011

TÓM TẮT

Phương pháp phân lớp sử dụng máy vec-tơ hỗ trợ SVM (support vector machine) là một phương pháp nổi tiếng dựa trên việc cực đại hóa dải biên phân lớp (max margin classification) và việc lựa chọn các hàm nhân (kernel) phù hợp. Phương pháp này được sử dụng rộng rãi để giải quyết nhiều bài toán của tin sinh học do tính hiệu quả, độ chính xác cao, và khả năng xử lý đối với các bộ dữ liệu lớn. Trong bài viết này, chúng tôi giới thiệu những vấn đề cơ bản của kỹ thuật phân lớp sử dụng SVM, đồng thời giới thiệu một bộ công cụ phần mềm SVM cho bài toán phân lớp. Sau đó, trình bày một số thành công trong ứng dụng SVM cho một vài bài toán Tin sinh học, cụ thể là bài toán phát hiện vị trí cắt-nối (splice site detection) và bài toán phân lớp biểu hiện gene (gene expression classification).

Từ khóa: Biểu hiện gene, ghép mảnh, máy vec-tơ hỗ trợ, phân lớp/dự báo, SVM, tin sinh học

ABSTRACT

Support vector machines (SVMs) are well-known method for solving classification problems based on the idea of margin maximization and kernel functions. SVMs are widely used in Bioinformatics due to their high accuracy, efficiency and a great ability to deal with complex datasets. In this paper, basic principles of SVMs learning for classification and a well-known SVM toolbox for the task are briefly introduced. Then, we present some significant successes of using SVM for solving Bioinformatics problems based on results of applying SVM for the problem of splice site detection and gene expression classification.

Keywords: Bioinformatics, classification/prediction, gene expression, splice site support vector machine.

1. ĐẶT VẤN ĐỀ

Trong những thập kỷ gần đây, những nghiên cứu về gene và di truyền luôn được quan tâm sâu sắc. Những nghiên cứu này đã có những thành công nhất định, đồng thời cũng tạo ra một khối lượng lớn các dữ liệu đa dạng về gene sinh học. Các dữ liệu này nếu đơn giản chỉ để lưu giữ thì chỉ cần các hệ quản trị cơ sở dữ liệu. Tuy nhiên, để có thể khám phá và khai thác những thông tin quý giá tiềm tàng trong các dữ liệu này. Để hiểu về các hệ thống sinh học, thì ta phải cần đến các phương pháp tính toán phức tạp với các giải thuật tính toán chính xác và hiệu quả.

Rất nhiều vấn đề quan trọng trong sinh học tính toán (Computational Biology) liên quan đến bài toán phân lớp (classification)

hay dự báo (prediction), như: dự báo vị trí cắt-nối (splice site prediction) để tìm kiếm gene, dự báo cấu trúc gene, chức năng của gene, sự tương tác, và vai trò của gene trong một số loại bệnh tật v.v. Một trong những kỹ thuật tính toán nổi tiếng cho bài toán phân lớp/dự báo cho độ chính xác cao và được sử dụng rộng rãi trong cộng đồng nghiên cứu tin sinh học trong những năm gần đây là kỹ thuật phân lớp sử dụng máy vec-tơ hỗ trợ SVM (support vector machine). Trong bài viết này, chúng tôi sẽ giới thiệu những vấn đề cơ bản của lý thuyết học máy (machine learning) cho bài toán phân lớp sử dụng SVM, đồng thời giới thiệu bộ công cụ phần mềm LibSVM trên nền Matlab cho bài toán phân lớp. Sau đó chúng tôi sẽ tìm hiểu, tổng hợp và giới thiệu về một số thành công trong ứng dụng SVM giải quyết một số bài toán tin sinh học, cụ thể là bài toán phát hiện vị trí cắt-nối (splice site detection) và bài toán phân lớp biểu hiện gene (gene expression classification).

2. PHƯƠNG PHÁP NGHIÊN CỨU

Đây là một bài phân tích, tổng hợp nhằm tìm hiểu và giới thiệu công cụ phần mềm máy tính, gồm giải thuật và công cụ phần mềm, ứng dụng trong sinh học tính toán. Các tài liệu thứ cấp được sử dụng để nghiên cứu về cơ sở lý thuyết của phương pháp phân lớp máy vec-tơ hỗ trợ SVM. Với bộ công cụ phần mềm SVM, dựa trên các tài liệu gốc về cài đặt và hướng dẫn sử dụng; các thử nghiệm được làm trực tiếp với công cụ phần mềm trên nền hệ điều hành Windows với các bộ dữ liệu và các tham số được thiết lập khác nhau. Các nghiên cứu ứng dụng SVM cho các bài toán tin sinh học được nghiên cứu, tổng hợp từ nhiều bài viết, các nghiên cứu và thí nghiệm từ nhiều nguồn khác nhau.

3. MÁY VEC-TƠ HỖ TRỢ SVM

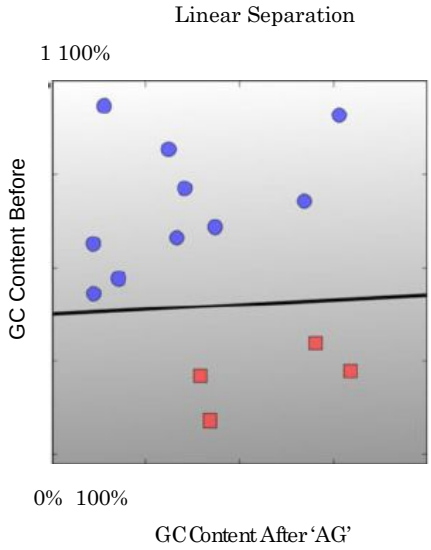
3.1. Bài toán phân lớp

Phân lớp (classification) là một tiến trình xử lý nhằm xếp các mẫu dữ liệu hay các đối tượng vào một trong các lớp đã được định nghĩa trước. Các mẫu dữ liệu hay các đối tượng được xếp vào các lớp dựa vào giá trị của các thuộc tính (attributes) cho một mẫu dữ liệu hay đối tượng. Sau khi đã xếp tất cả các đối tượng đã biết trước vào các lớp tương ứng thì mỗi lớp được đặc trưng bởi tập các thuộc tính của các đối tượng chứa trong lớp đó.

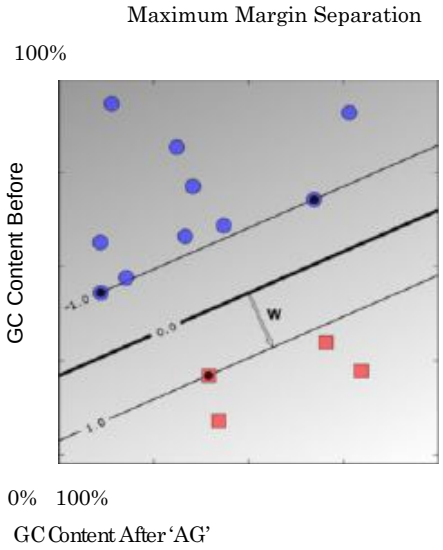
Quá trình phân lớp còn được gọi là quá trình gán nhãn cho các tập dữ liệu. Nhiệm vụ của bài toán phân lớp dữ liệu là cần xây dựng mô hình (bộ) phân lớp để khi có một dữ liệu mới vào thì mô hình phân lớp sẽ cho biết dữ liệu đó thuộc lớp nào. Có nhiều cách để biểu diễn một mô hình phân lớp và có rất nhiều thuật toán giải quyết nó. Các thuật toán phân lớp tiêu biểu bao gồm như mạng neural, cây quyết định, suy luận quy nạp, mạng Beyesian, Support Vector Machine (SVM),... Trong các kỹ thuật đó, SVM được coi là công cụ mạnh, phổ biến và đặc biệt thích hợp cho phân lớp dữ liệu lớn và nhiều chiều.

3.2. SVM cho bài toán phân lớp tuyến tính

Hình thức đơn giản của việc phân lớp là phân lớp nhị phân: phân biệt giữa các đối tượng thuộc về một trong hai lớp: dương (+1) hoặc âm (-1). SVMs sử dụng hai khái niệm để giải quyết vấn đề này: 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 1). 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 ta có thể phân tách khá xa hai tập dữ liệu (Hình 1 và 2). Đây là ý tưởng về sự phân chia biên rộng.



Hình 1. Một đường thẳng tuyến tính phân chia 2 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. Độ rộng biên lớn nhất được tính toán bởi một SVMs 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. Ở đây, có ba support vectors trên các cạnh của vùng biên (f(x) = -1 hoặc f (x)=1).

Trong phần này, ý tưởng về phân lớp tuyến tính sử dụng SVM được giới thiệu. Các dữ liệu bao gồm các đối tượng có nhãn là một trong hai nhãn. Để thuận tiện, giả định rằng các nhãn +1 (dương) và -1 (âm). 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.

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

$$f(x) = \langle w, x\rangle + b \qquad 1)$$

Hàm *f(x)* là hàm của 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 2 chiều các điểm ứng với phương trình *<w, x> = 0* tương ứng với một đường qua gốc tọa độ, trong không gian 3 chiều thì nó là một mặt phẳng qua gốc tọa độ. Biến *b* sẽ dịch chuyển mặt phẳng đi một lượng so với mặt phẳng qua gốc tọa 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 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)) 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.

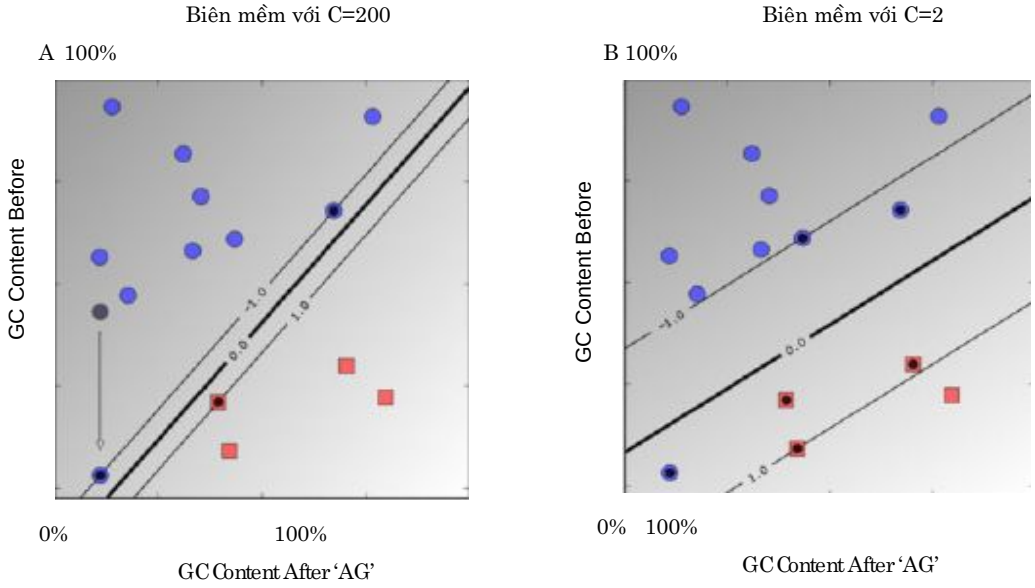
Với bất kỳ một tập dữ liệu khả tách tuyến tính có tồn tại một mặt phẳng phân lớp tất cả các điểm dữ liệu. Có nhiều mặt phẳng như vậy nhưng phải lựa chọn mặt phẳng nào để đảm bảo thời gian huấn luyện ngắn và phân lớp một cách chính xác. Thực tế quan sát cũng như lý thuyết học thống kê (Vapnik, 1999) cho thấy rằng phân lớp siêu phẳng sẽ làm việc tốt hơn nếu siêu phẳng tách biệt chính xác với một biên độ lớn. Ở đây, biên của một phân lớp tuyến tính được định nghĩa là khoảng cách gần nhất để quyết định ranh giới, như thể hiện trong hình 2. Có thể điều chỉnh *b* để siêu phẳng phân tách các điểm tương ứng. Hơn nữa nếu cho phương trình (1) các giá trị ± 1, thì biên độ sẽ là *1 / ||w||* (trong đó *||w||* là độ dài của vec tơ *w*) còn được gọi là chuẩn, được tính là $\sqrt{< w, w >}$.

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 các dữ liệu dạng này (Hình 2). Để tính toán *w* và *b* tương ứng với các biên cực đại, ta phải giải quyết bài toán tối ưu sau đây:

$$\underset{w,b}{\text{minimize}} \ \frac{1}{2}\|w\|^2, \text{ với ràng buộc: } y_i(\langle w, x_i\rangle + b) \geq 1, \ i = 1,.....,n, \qquad 2)$$

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 $y_i(<w,x_i>+b) \geq 1$, với *w* càng nhỏ càng tốt. Bài toán tối ưu hóa này có thể được giải bằng cách sử dụng các công cụ tiêu chuẩn từ tối ưu hóa lồi (Boyd và Vandenberghe, 2004).



Hình 3. 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 bên trái, 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 phép bỏ qua điểm gần ranh giới, và làm tăng biên. Ranh giới quyết

định giữa các điểm dương và các điểm âm được thể hiện bằng dòng đậm.
Các dòng nhạt hơn là biên độ (giá trị bằng -1 hoặc +1).

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 (2) với y_i ($\langle w, x_i \rangle + b) \geq 1 - \xi_i$, $i = 1,..., n$, trong đó $\xi_i \geq 0$ là các biến phụ không âm. $C\sum_{i=1}^{n}\xi_i$ được thêm vào hàm tối ưu hóa:

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

ràng buộc: $y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i$, 3)

$\xi_i \geq 0$

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 (Cortes và Vapnik, 1995).

Ảnh hưởng của sự lựa chọn C được minh họa trong hình 3. Với một giá trị C lớn (minh họa hình 3A), hai điểm gần siêu phẳng nhất bị ảnh hưởng lớn hơn các điểm dữ liệu khác. Khi C giảm (Hình 3B), những điểm chuyển động bên trong lề, và hướng của siêu phẳng được thay đổi, dẫn đến một biên lớn hơn cho dữ liệu. Lưu ý rằng giá trị của C không có ý nghĩa trực tiếp, và có một công thức của SVMs trong đó sử dụng một tham số trực quan hơn $0 < \nu \leq 1$. Tham số ν kiểm soát các véctơ hỗ trợ, và lỗi biên (Schölkopf và Smola, 2002), và (Shawe và Cristianini, 2004).

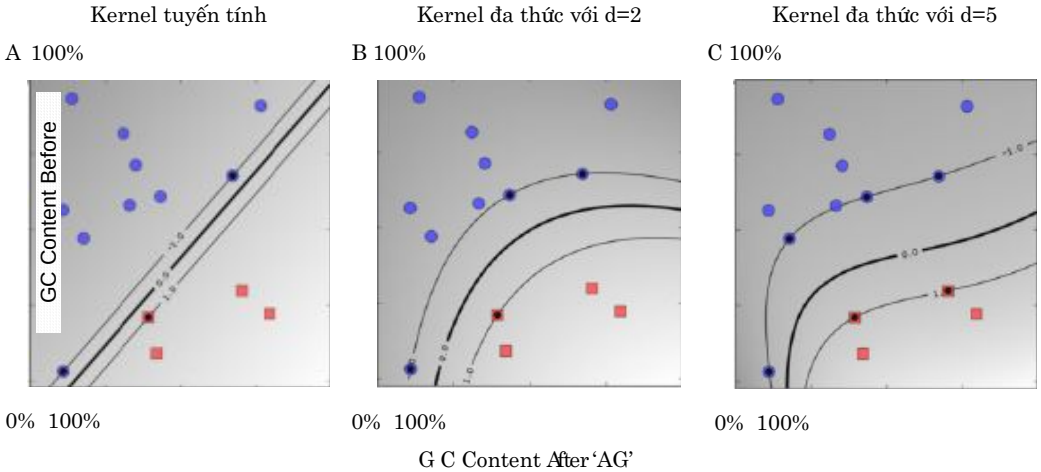
3.3. SVM cho phân lớp phi tuyến

Trong nhiều ứng dụng, một bộ phân lớp phi tuyến có độ chính xác cao hơn. Tuy nhiên, phân lớp tuyến tính có một lợi thế đó là các thuật toán đơn giản (Bishop, 2007; Hastie & cs 2001). Điều này đặt ra câu hỏi có cách phân lớp tuyến tính nào có thể mở rộng cho phi tuyến không? Hơn nữa, chúng ta có thể xử lý dữ liệu có thể không được biểu diễn trong không gian vectơ, như trong lĩnh vực sinh học.

Có một cách đơn giản chuyển phân lớp tuyến tính sang phi tuyến hoặc sử dụng cho phân lớp dữ liệu không biểu diễn dưới dạng vectơ. Đó là ánh xạ dữ liệu cho một không gian vector nào đó, mà chúng ta sẽ đề cập đến như là không gian đặc trưng, bằng cách sử dụng hàm ϕ. Hàm đó là:

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

4)



Hình 4. Mức độ tác động của kernel đa thức. Kernel đa thức dẫn đến một sự phân tách tuyến tính (A). Kernel đa thức cho phép một ranh giới quyết định linh hoạt hơn (B - C).

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 cả các tích của các cặp (liên quan đến kernel đa thức). Kết quả là một một bộ phân loại có dạng hàm phân tách bậc hai. Cách tiếp cận tính toán trực tiếp các đặc trưng phi tuyến này khó mở rộng cho số lượng đầu vào lớn.

Chiều của không gian đặc trưng liên quan kích thước của không gian đầu vào. Nếu chúng ta sử dụng đơn thức bậc *d* cao hơn 2, số chiều sẽ lũy thừa theo *d*, kết quả là tăng sử dụng bộ nhớ và thời gian cần thiết để tính toán các hàm phân tách. Nếu dữ liệu nhiều chiều, chẳng hạn như trong trường hợp dữ liệu biểu hiện gen, thì rất phức tạp. Phương pháp kernel tránh điều phức tạp này bằng cách ánh xạ dữ liệu tới không gian đặc trưng nhiều chiều.

Chúng ta đã thấy ở trên là 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 thuật giải tuyến tính. Hàm phân tách trở thành:

$$f(x) = \sum_{i=1}^{n} y_i \alpha_i \langle \phi(x_i), \phi(x) \rangle + b \qquad (5)$$

Việc biểu diễn dưới dạng biến 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 (4)) khi thay thế x_i với $\phi(x_i)$.

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

$$k(x, x') = \langle \phi(x), \phi(x') \rangle \qquad (6)$$

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 các độ đo tương tự (hàm kernel) có thể được tính một cách hiệu quả.

Kernel cho các dữ liệu thực

Dữ liệu thực là dữ liệu mà các mẫu là các vector có số chiều xác định. Đây là dạng dữ liệu phổ biến trong tin sinh học và nhiều lĩnh vực khác. Một vài ví dụ về áp dụng SVM xử lý dữ liệu thực bao gồm dự đoán trạng thái của bệnh từ dữ liệu vi mảng (Guyon I & cs, 2002), và dự đoán chức năng protein từ một tập các tính năng bao gồm thành phần acid amin và các thuộc tính khác nhau của các axit amin trong protein (Cai & cs., 2003).

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:

$$k_{d,k}^{polynomial}(x, x') = (\langle x, x' \rangle) + k)^d \qquad (7)$$

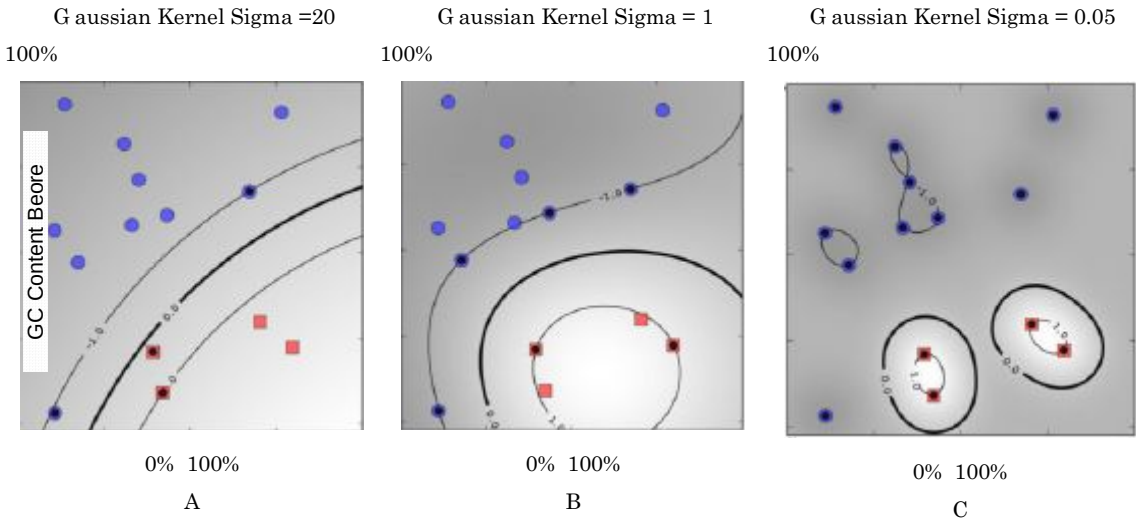
κ là thường được chọn là 0 (đồng nhất) hoặc 1 (không đồng nhất). Không gian đặc trưng cho các hàm kernel không đồng nhất bao gồm tất cả các đơn thức bậc nhỏ hơn *d* (Schölkopf và Smola, 2002). Nhưng, thời gian tính toán của nó là tuyến tính với số chiều của không gian đầu vào. Kernel với *d = 1* và $\kappa = 0$, 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 4). 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 4 ở đa thức bậc 2 đã đủ linh hoạt để phân biệt giữa hai lớp với một biên tốt. Đa thức bậc 5 định lượng một ranh giới quyết định tương tự, với độ cong lớn hơn.

Quá trình chuẩn hóa có thể giúp cải thiện hiệu suất và ổn định *d*.

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

$$k_{\sigma}^{Gaussian}(x, x') = \exp(-\frac{1}{\sigma} \| x - x' \|^2 \quad (8)$$



Hình 5. Ảnh hưởng của số chiều Gaussian kernel (σ) cho một giá trị cố định của các hằng số biên mềm. Đối với giá trị của σ (A) lớn quyết định ranh giới là gần như tuyến tính. Khi giảm σ tính linh hoạt của ranh giới quyết định tăng (B). Giá trị σ nhỏ dẫn đến học quá (overfitting) (C)

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 (hình 4-5). 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 một ví dụ minh họa, các kết quả trên một mẫu lớn hơn nhiều các tập dữ liệu hai chiều xác định vị trí cắt-nối được hiển thị trong bảng 1. Việc sử dụng của 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 thu được giảm.

3.4. Bộ công cụ phân lớp LibSVM trên nền Matlab

LibSVM là một trong số nhiều thư viện hỗ trợ cho SVM (Chih-Chung Chang & cs, 2011). LibSVM được viết bằng C++ và Java, và có thể chạy được trên nhiều hệ điều hành khác nhau như: Windows, Unix…

Bảng 1. SVM tính chính xác vào các nhiệm vụ xác nhận vị trí cắt -nối sử dụng đa thức và Gaussian kernel với d và độ rộng **σ** khác nhau

Kernel	auROC
Linear	88,2%
Polynomial d = 3	91,4%
Polynomial d = 7	90,4%
Gausian σ = 100	87,9%
Gausian σ = 1	88,6%
Gausian σ = 0,1	77,3%

LibSVM là một thư viện đơn giản dễ sử dụng, và hiệu quả cho SVM để phân lớp (C-SVC, nu-SVC), hồi quy (epsilon-SVR, nu-SVR), ước lượng phân phối (one-class SVM) và hỗ trợ phân lớp đa lớp (multi-class classification). Với mục đích là cho phép người sử dụng có thể dễ dàng sử dụng SVM vào các ứng dụng cụ thể của họ.

Có thể kể đến những nghiên cứu thành công trong một số lĩnh vực đã sử dụng LibSVM như:

- Thị giác máy tính
- Xử lý ngôn ngữ tự nhiên
- Tin sinh học

Các tính năng chính của LibSVM bao gồm:

- Cho phép người dùng lựa chọn các công thức SVM khác nhau
- Thực hiện phân lớp đa lớp hiệu quả
- Xác nhận chéo để lựa chọn mô hình
- Ước lượng xác suất
- Lựa chọn các hàm nhân khác nhau: tuyến tính, đa thức ...
- SVM trọng số cho dữ liệu không cân bằng
- Lựa chọn mô hình tự động

Quá trình sử dụng LibSVM:

Để sử dụng LibSVM, cần chuẩn bị dữ liệu cho quá trình huấn luyện và thử nghiệm. Dữ liệu dùng để huấn luyện và thử nghiệm được lưu trong các tập tin sao cho mỗi hàng trong tập tin là một mẫu với các thông tin được trình bày theo dạng:

<label> <index1>:<value1> <index2> : <value2> ...

Trong đó:

<label> là một giá trị xác định nhãn của lớp, với bài toán phân lớp nó là một số nguyên, đối với hồi quy nó là một số thực bất kỳ.

Mỗi cặp <index1>:<value1> tương ứng một đặc trưng, giá trị <index> là một số nguyên bắt đầu từ 1 và <value> là một số thực.

LibSVM có một số lệnh cho phép đọc dữ liệu từ tập tin và chuẩn hóa dữ liệu vào như: libsvmread, svm_scale ...

Sau khi chuẩn bị dữ liệu, quá trình sử dụng LibSVM bao gồm 2 bước:

Bước 1: Huấn luyện (training):

Sử dụng một tập hợp dữ liệu để huấn luyện:

svm-train [options] training_file [model_file]

Trong đó:

Option: tham số này cho phép người dùng lựa chọn các công thức SVM khác nhau, các lớp hàm nhân khác nhau cùng với các thuộc tính cho hàm nhân.

training_file: tập tin chứa dữ liệu dùng để huấn luyện

model_file: tập tin chứa mô hình huấn luyện. Mô hình huấn luyện là một cấu trúc có thể bao gồm các tham số:

- Số lượng các lớp
- Tổng số véctơ hỗ trợ (support vector)
- Các tham số w, -b trong phương trình wx-b
- Nhãn cho mỗi lớp
- Số lượng véctơ cho mỗi lớp ...

Bước 2: Thử nghiệm mô hình (testing):

Sử dụng mô hình (ở bước 1) để dự đoán thông tin của một tập dữ liệu.

svm-predict [options] test_file model_file output_file

Trong đó:

options: -b 0 hoặc -b 1 để dự đoán ước lượng xác suất

test_file: tập tin chứa dữ liệu thử nghiệm

model_file: mô hình được tạo ra bởi svm-train

output_file: tập tin chứa kết quả của quá trình thực nghiệm bao gồm:

- Độ chính xác; độ chính xác vector (phân lớp), hệ số tương quan bình phương (hồi quy).

- Ma trận chứa các giá trị quyết định hoặc xác suất ước tính.

- Nhãn dự đoán cho mỗi đặc trưng

4. SỬ DỤNG SVM TRONG MỘT SỐ BÀI TOÁN PHÂN LỚP

4.1 Bài toán phát hiện vị trí cắt -nối

Như đã biết, các vị trí cắt-nối trên DNA là ranh giới của exon (mã cho những phần protein) và intron (không mang thông tin mã hóa). Xác định chính xác vị trí cắt-nối giúp dễ dàng xác định chính xác vị trí gen trên DNA. Từ khi phần lớn DNA được phân tích gen, vấn đề xác định chính xác vị trí cắt-nối lại càng quan trọng hơn. SVMs là thuật toán xuất sắc để giải quyết vấn đề phân lớp. SVM đã được áp dụng thành công trong nhiều lĩnh vực trong đó có vấn đề tin sinh học (Jaakkola và Haussler, 1999). SVM đơn giản nhất sử dụng cho phân lớp nhị phân: phân biệt các vị trí cho (vị trí ranh giới exon-intron) và vị trí nhận (vị trí ranh giới intron-exon) từ các vị trí nghi vấn.

Thực hiện huấn luyện và đánh giá SVM trên các tập dữ liệu mới được tạo ra đã cải tiến hiệu suất trong việc dự đoán các vị trí cắt-nối trong tập dữ liệu so với các nghiên cứu trước đó. Đã có nhiều nhóm nghiên cứu sử dụng SVM và có kết quả rất tốt.

Nhóm tác giả Sören Sonnenburg (2007) đã dự đoán chính xác vị trí cắt-nối sử dụng SVM kết hợp với kernel trọng số để xác định vị trí vị trí cắt-nối trong gene của Caenorhabditis elegans, ruồi giấm Drosophila melanogaster, A rabidopsis thaliana, Danio rerio, và Homo sapiens. Nhóm nghiên cứu chỉ ra các vị trí cắt-nối rất chính xác trong các bộ gen và phương pháp này tốt hơn rất nhiều phương pháp khác như: chuỗi Markov, GeneSplicer và SpliceMachine. Nhóm không những xác định các vị trí cắt-nối trong gene mà còn cung cấp công cụ dự báo độc lập sử dụng kết hợp với bộ tìm gen.

Nhóm nghiên cứu Gunnar Ratsch và Soren Sonnenburg đã có nghiên cứu về "dự đoán chính xác các vị cắt-nối của aenorhabditis elegans sử dụng SVM". Nhóm đã thiết kế và thử nghiệm hệ thống tìm vị trí cắt-nối trong đó kết hợp dự đoán SVM với thông tin thống kê bổ sung về vị trí cắt-nối. Sử dụng hệ thống này có thể để dự đoán chính xác cấu trúc exon-intron của một gen. Hệ thống này đã được thử nghiệm thành công trên tập gen mới được tạo ra và so sánh với GenScan. Hệ thống dự báo vị trí cắt-nối mà nhóm sử dụng chính xác hơn 92%, trong khi GenScan chỉ đạt được độ chính xác 77,5%.

4.2 Bài toán phân lớp biểu hiện gene

Sự ra đời của công nghệ vi mảng DNA đã mang lại cho các nhà phân tích một khối lượng lớn dữ liệu về sự biểu hiện gen. Một số bộ dữ liệu được công khai trên Internet và đã có rất nhiều nhóm các nhà nghiên cứu đã thực hiện các nghiên cứu khác nhau trên bộ dữ liệu này. Một trong số đó có thể kể đến nhóm Isabelle G uyon (2000). Nhóm đã nghiên cứu và chứng minh rằng bằng cách

áp dụng SVMs có thể lựa chọn được một tập con các gen từ vi mảng DNA để xây dựng bộ phân loại với độ tin cậy cao.

Kết quả nghiên cứu của nhóm được thử nghiệm trên hai bộ dữ liệu gen lấy ra từ vi mảng DNA của một số bênh nhân.

Trên bộ dữ liệu thứ nhất thu được từ những bệnh nhân ung thư với hai biến thể khác nhau của bệnh bạch cầu (ALL và AML). Dữ liệu được chia thành hai tập con: Một tập huấn luyện, được sử dụng để lựa chọn gen và điều chỉnh trọng số của phân loại, và một tập khác được sử dụng để ước lượng hiệu suất của hệ thống thu được. Tập huấn luyện bao gồm 38 mẫu (27 ALL và 11 AML) lấy từ các mẫu tủy xương, tập thử nghiệm có 34 mẫu (20 ALL và 14 AML) bao gồm 24 mẫu lấy từ tủy xương và 10 mẫu xét nghiệm mẫu máu. Mỗi mẫu có 7129 đặc trưng.

Thực hiện so sánh SVMs và phương pháp cơ bản trong các trường hợp bộ phân loại được huấn luyện trên tập con các gen được lựa chọn theo phương pháp SVM RFE và phương pháp cơ bản kết quả cho thấy với cả hai phương pháp phân loại, lựa chọn gen với SVM RFE luôn cho hiệu suất tốt hơn phương pháp cơ bản.

Trên bộ dữ liệu thứ hai thu được từ những mô đại tràng ung thư hoặc bình thường. Thực hiện so sánh các kỹ thuật lựa chọn gen khác nhau với cùng một bộ phân loại (SVM tuyến tính). Bộ dữ liệu sử dụng được lấy từ DNA micro-array, sau khi tiền xử lý cho ra một bảng của 62 mẫu x 2000 giá trị biểu hiện gen. 62 mẫu bao gồm 22 mẫu bình thường và 40 mẫu ung thư. Thực hiện phân chia ngẫu nhiên 62 mẫu thành 2 tập: 31 mẫu dùng để huấn luyện và 31 mẫu dùng để thử nghiệm. Kết quả thử nghiệm cho thấy phương pháp SVM (bộ phân loại SVM được huấn luyện trên tập gen được lựa chọn theo phương pháp SVM RFE) là tốt hơn đáng kể so với phương pháp cơ bản.

Như vậy, nhóm đã chứng minh bằng thực nghiệm rằng các gen được lựa chọn bằng các kỹ thuật SVM thực hiện phân loại tốt hơn trong việc phân loại ung thư. Phương pháp của nhóm nghiên cứu này đạt độ chính xác là 98%, trong khi phương pháp cơ bản độ chính xác là chỉ có 86%. Ngoài ra, SVM thực hiện tốt hơn với số lượng gene ít hơn.

5. KẾT LUẬN

Với khả năng vượt trội của SVM về tính hiệu quả, độ chính xác, khả năng xử lý các bộ dữ liệu một cách linh hoạt, việc sử dụng máy vec-tơ hỗ trợ SVM đã và đang là sự lựa chọn tối ưu nhất trong việc giải quyết các bài toán phân loại/dự báo trong một số các các ngành khoa học. Trong bài viết này, chúng tôi đã giới thiệu phương pháp phân lớp sử dụng máy vec-tơ hỗ trợ SVM cho bài toán phân loại nói chung và một số bài toán trong Tin sinh học, cụ thể là bài toán phát hiện vị trí cắt-nối (splice site detection) và bài toán phân loại biểu hiện gene (gene expression classification). Ngoài ra, đã giới thiệu bộ công cụ phân loại LibSVM trên nền Matlab với mục đích là cho phép người sử dụng dễ dàng sử dụng SVM vào các ứng dụng cụ thể. Với những lợi thế sẵn có của SVM, việc ứng dụng và cái tiến thuật toán phân loại sử dụng máy vec-tơ hỗ trợ SVM vào bài toán phân lớp trong Tin sinh học (điển hình là các bài toán liên quan đến gene và di truyền) là một lĩnh vực mà nhóm chúng tôi quan tâm và tập trung nghiên cứu trong thời gian tới vì những ứng dụng thiết thực của nó trong thực tiễn.

TÀI LIỆU THAM KHẢO

Bishop C (2007). Pattern Recognition and Machine Learning. Springer.

Boyd S, Vandenberghe L (2004). Convex Optimization. Cambridge University Press.

Brown MPS, Grundy WN, Lin D, Cristianini N, Sugnet C, Furey TS,Ares JM, Haussler D (2000). Knowledge-based analysis of microarray gene expression data using support vector machines. PNAS, 97:262-267.

Chang C. and C.-J. Lin. LIBSVM (2011). A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1{27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

Cai C, Han L, Ji Z, Chen X, Chen Y (2003). SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence. Nucl Acids Res 31:3692-3697. doi:10.1093/nar/gkg600. URLhttp://nar.oxfordjournals.org/cgi/content/abstract/31/13/3692. http://nar.oxfordjournals.org/cgi/reprint/31/13/3692.pdf.

Cortes C, Vapnik V (1995). Support vector networks. Machine Learning 20:273-297.

Guyon I, Weston J, Barnhill S, Vapnik V (2002). Gene selection for cancer classification using support vector machines. Mach Learn 46:489-422.

Hastie T, Tibshirani R, Friedman J (2001). The Elements of Statistical Learning. Springer.

Isabelle Guyon, Jason Weston, Stephen Barnhill, M.D and Vladimir Vapnik (2000). Gene Selection for Cancer Classification using Support Vector Machines. Submitted to Machine Learning

Jaakkola T, Haussler D (1999). Exploiting Generative Models in Discriminative Classifiers. In Advances in Neural Information Processing Systems Volume 11. Edited by: Kearns M, Solla S, Cohn D. Cambridge, MA, MIT Press; 487-493.

Schölkopf B, Smola A (2002). Learning with Kernels. Cambridge, MA: MIT Press.

Shawe-Taylor J, Cristianini N (2004). Kernel Methods for Pattern Analysis. Cambridge, UK: Cambridge UP.

Sören Sonnenburg, Gabriele Schweikert, Petra Philips, Jonas Behr and Gunnar Rätsch (2007) Accurate splice site prediction using support vector machines. BMC Bioinformatics 8(S-10):

Tsuda K, Kawanabe M, Rätsch G, Sonnenburg S, Müller K (2002). A New Discriminative Kernel from Probabilistic Models. Advances in Neural information processings systems, 14:977

Vapnik V (1999). The Nature of Statistical Learning Theory. Springer, 2nd edition.

Zien A, Rätsch G, Mika S, Schölkopf B, Lengauer T, Müller KR (2000). Engineering Support Vector Machine Kernels That Recognize Translation Initiation Sites. BioInformatics, 16(9):799-807.