Kỷ yếu Hội nghị Khoa học Quốc gia lần thứ IX "Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR'9)"; Cần Thơ, ngày 4-5/8/2016
DOI: 10.15625/vap.2016.00025

GIẢI THUẬT *t*SVM CHO PHÂN LỚP PHI TUYẾN TẬP DỮ LIỆU LỚN

Đỗ Thanh Nghị, Phạm Nguyên Khang, Trần Nguyễn Minh Thư, Nguyễn Hữu Hòa
Khoa CNTT-TT, Trường Đại học Cần Thơ
Khu 2, Đường 3/2, Xuân Khánh, Ninh Kiều, TP. Cần Thơ
dtnghi@cit.ctu.edu.vn

TÓM TẮT— *Trong bài viết này, chúng tôi trình bày giải thuật tSVM cho phân lớp phi tuyến tập dữ liệu lớn. Giải thuật tSVM sử dụng máy học cây quyết định để phân hoạch nhanh tập dữ liệu lớn thành k phân vùng được gọi là nút lá. Chỉ những nút lá có nhãn (lớp) của các phần tử thuần nhất (giống nhau) được giải thuật tSVM gán nhãn tương ứng như giải thuật cây quyết định dùng để phân lớp. Với mỗi nút lá có nhãn các phần tử không thuần nhất, giải thuật tSVM huấn luyện một mô hình SVM phi tuyến dùng để phân lớp dữ liệu cục bộ của nút lá. Việc huấn luyện các mô hình SVM trên từng nút lá có nhãn không thuần nhất hoàn toàn độc lập với nhau, vì thế có thể được thực hiện song song trên các máy tính multi-core. Kết quả thực nghiệm trên các tập dữ liệu của UCI và 3 tập dữ liệu nhận dạng ký tự viết tay và tập dữ liệu phân lớp ảnh cho thấy giải thuật tSVM cho kết quả phân lớp nhanh, chính xác khi so sánh với giải thuật SVM chuẩn như LibSVM.*

Từ khóa— *Máy học véc-tơ hỗ trợ (SVM), mô hình máy học cục bộ, phân lớp phi tuyến tập dữ liệu lớn.*

I. GIỚI THIỆU

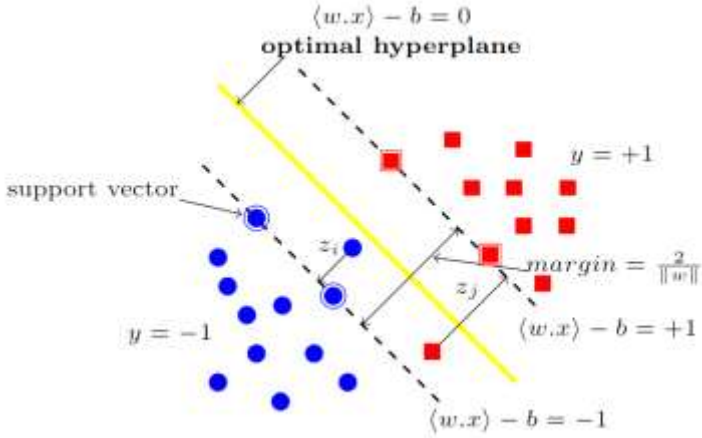
Giải thuật cây quyết định [Breiman et al., 1984], [Quinlan, 1993] và máy học véc-tơ hỗ trợ (Support Vector Machines - SVM [Vapnik, 1995]) được cộng đồng khám phá tri thức và khai thác dữ liệu bình chọn là hai trong top 10 giải thuật khai thác dữ liệu phổ biến và hiệu quả [Wu & Kumar, 2009]. Ưu điểm của mô hình cây quyết định là giải thuật huấn luyện đơn giản, nhanh, xử lý được cả dữ liệu rời rạc và liên tục, luật quyết định rút trích từ mô hình cây quyết định dễ hiểu với chuyên gia về dữ liệu. Trong khi đó, giải thuật máy học véc-tơ hỗ trợ sử dụng các hàm hạt nhân (kernel function), cung cấp các mô hình có độ chính xác rất cao cho các vấn đề phân lớp và hồi quy phi tuyến trong thực tế. Các ứng dụng thành công của SVM đã được công bố trong nhiều lĩnh vực khác nhau như nhận dạng mặt người, phân lớp văn bản và tin-sinh học [Guyon, 1999]. Mặc dù có được những ưu điểm kể trên, giải thuật huấn luyện một mô hình SVM có độ phức tạp cao so với giải thuật huấn luyện mô hình cây quyết định do phải giải bài toán quy hoạch toàn phương (quadratic programming). Độ phức tạp tối thiểu của giải thuật huấn luyện mô hình SVM là bậc 2 của số lượng phần tử dữ liệu [Platt, 1999]. Do đó, cần thiết phải có những cải tiến để giải thuật học SVM có thể xử lý được các tập dữ liệu với kích thước lớn.

Trong bài viết, chúng tôi đề xuất giải thuật mới *t*SVM là giải thuật lai của giải thuật cây quyết định và máy học SVM để tận dụng được ưu điểm của cả hai lớp giải thuật học này. Giải thuật *t*SVM có thể huấn luyện mô hình chính xác như giải thuật SVM gốc nhưng có độ phức tạp thấp hơn. Để đạt được cả hai mục tiêu này, giải thuật *t*SVM sử dụng máy học cây quyết định để phân hoạch nhanh tập dữ liệu lớn thành *k* phân vùng được gọi là nút lá. Các nút lá có nhãn (lớp) của các phần tử thuần nhất (giống nhau) được gán nhãn giống như giải thuật cây quyết định, sử dụng khi phân lớp. Với các nút lá có nhãn các phần tử không thuần nhất, giải thuật *t*SVM huấn luyện song song các mô hình SVM phi tuyến cục bộ, mỗi mô hình dùng để phân lớp dữ liệu cục bộ dữ liệu của nút lá. Kết quả thực nghiệm trên các tập dữ liệu của UCI [Lichman, 2013] và 3 tập dữ liệu nhận dạng ký tự viết tay [Lecun et al., 1989], [Lecun et al., 1998], [van der Maaten, 2009] và tập dữ liệu phân lớp ảnh [Geusebroek et al., 2005] cho thấy giải thuật *t*SVM cho kết quả phân lớp nhanh, chính xác khi so sánh với giải thuật SVM chuẩn như LibSVM [Chang & Lin, 2011].

Phần còn lại của bài viết được tổ chức như sau. Chúng tôi sẽ trình bày tóm tắt giải thuật máy học véc-tơ hỗ trợ trong phần 2. Giải thuật *t*SVM được trình bày trong phần 3. Kết quả thực nghiệm sẽ được trình bày trong phần 4. Các nghiên cứu liên quan được thảo luận trong phần 5 trước khi kết luận và hướng phát triển được trình bày trong phần 6.

II. MÁY HỌC VÉC-TƠ HỖ TRỢ

Xét ví dụ phân lớp nhị phân tuyến tính đơn giản được mô tả như hình 1, với m phần tử $x_1, x_2, ..., x_m$ trong không gian n chiều (thuộc tính) với nhãn (lớp) của các phần tử tương ứng là $y_1, y_2, ..., y_m$ có giá trị *1* (lớp dương) hoặc giá trị *-1* (lớp âm). Giải thuật máy học SVM [Vapnik, 1995] tìm siêu phẳng tối ưu (xác định bởi véc-tơ pháp tuyến w và độ lệch của siêu phẳng với gốc toạ độ b) để tách dữ liệu ra 2 lớp. Máy học SVM tìm siêu phẳng cách xa 2 lớp nhất (siêu phẳng tối ưu) dựa trên 2 siêu phẳng hỗ trợ song song của 2 lớp. Siêu phẳng hỗ trợ của lớp *+1* ($w.x - b = +1$) là siêu phẳng mà các phần tử x_p thuộc lớp $y_p = +1$ nằm về phía bên phải của nó, tức là: $w.x_p - b \geq +1$. Tương tự, siêu phẳng hỗ trợ của lớp *-1* ($w.x - b = -1$) là siêu phẳng mà các phần tử x_n thuộc lớp $y_n = -1$ nằm về phía bên trái siêu phẳng hỗ trợ lớp *-1*, tức là: $w.x_n - b \leq -1$. Những phần tử nằm ngược phía với siêu phẳng hỗ trợ được coi như lỗi. Khoảng cách lỗi được biểu diễn bởi $z_i \geq 0$ (với x_i nằm đúng phía của siêu phẳng hỗ trợ của nó thì khoảng cách lỗi tương ứng $z_i = 0$, còn ngược lại thì $z_i > 0$ là khoảng cách từ điểm x_i đến siêu phẳng hỗ trợ tương ứng của nó). Khoảng cách giữa 2 siêu phẳng hỗ trợ được gọi là lề $= 2/||w||$, trong đó $||w||$ là độ lớn (2-norm) của pháp véc-tơ w. Siêu phẳng tối ưu (nằm giữa 2 siêu phẳng hỗ trợ) cần tìm phải thỏa 2 tiêu chí là cực đại hóa lề (lề càng lớn, mô hình phân lớp càng an toàn) và cực tiểu hóa lỗi. Vấn đề tìm siêu phẳng tối ưu của giải thuật SVM dẫn đến việc giải bài toán quy hoạch toàn phương (1):



Hình 1. Phân lớp tuyến tính với máy học SVM

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j K\langle x_i, x_j\rangle - \sum_{i=1}^{m}\alpha_i$$

với ràng buộc: (1)

$$\begin{cases} \sum_{i=1}^{m} y_i a_i = 0 \\ 0 \le a_i \le C \quad \forall\, i = 1,2,...,m \end{cases}$$

trong đó C là hằng số dương dùng để điều chỉnh độ lớn của lề và tổng khoảng cách lỗi; $K\langle x_i, x_j\rangle$ là hàm nhân tuyến tính $K\langle x_i, x_j\rangle = \langle x_i \times x_j\rangle$.

Giải bài toán quy hoạch toàn phương (1) thu được *#SV* phần tử x_i tương ứng với $\alpha_i > 0$, được gọi là các véc-tơ hỗ trợ. Chỉ cần *#SV* véc-tơ hỗ trợ này ta có thể dựng lại được siêu phẳng phân lớp. Mô hình SVM thực hiện phân lớp phần tử mới x bằng (2):

$$predict(x) = sign\left(\sum_{i=1}^{\#SV} y_i \alpha_i K\langle x, x_i\rangle - b\right) \quad (2)$$

Máy học SVM có thể sử dụng các hàm nhân khác nhau để giải quyết lớp các bài toán phân lớp phi tuyến [Cristianini & Shawe-Taylor, 2000]. Để xử lý các vấn đề phân lớp phi tuyến, không cần bất kỳ thay đổi nào hơn từ giải thuật mà chỉ cần thay thế hàm nhân tuyến tính trong (1) và (2) bằng các hàm nhân khác. Có 2 hàm nhân phi tuyến phổ biến là:

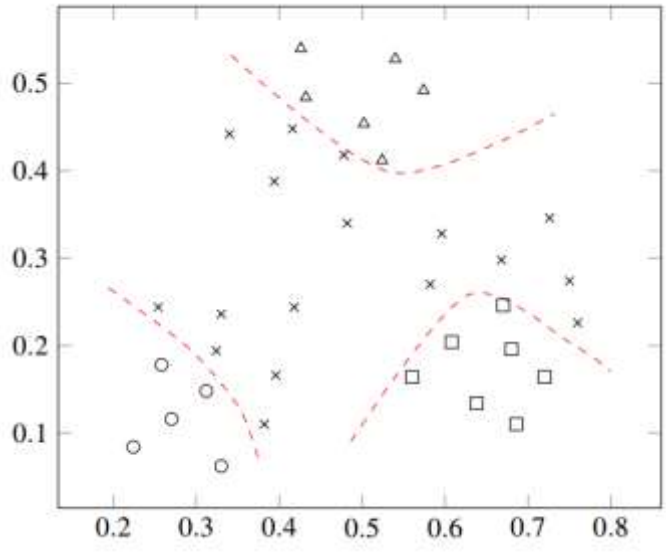
- Hàm đa thức bậc d: $K\langle x_i, x_j\rangle = \left(\langle x_i \times x_j\rangle + 1\right)^d$
- Hàm cơ sở bán kính (Radial Basic Function – RBF): $K\langle x_i, x_j\rangle = e^{-g\|x_i - x_j\|^2}$

Mô hình máy học SVM cho kết quả cao, ổn định, chịu đựng nhiễu tốt và phù hợp với các bài toán phân lớp, hồi quy. Nhiều ứng thành công của SVM đã được công bố trong nhiều lĩnh vực như nhận dạng ảnh, phân loại văn bản và sinh-tin học [Guyon, 1999].

Nghiên cứu trong [Platt, 1998] chỉ ra rằng các giải thuật huấn luyện được đề xuất trong [Boser et al., 1992], [Chang & Lin, 2011], [Osuna et al., 1997], [Platt, 1998] có độ phức tạp tính toán lời giải bài toán quy hoạch toàn phương (1) tối thiểu là $O(m^2)$ trong đó m là số lượng phần tử được dùng để huấn luyện. Điều này làm cho giải thuật SVM không phù hợp với dữ liệu lớn.

III. GIẢI THUẬT *t*SVM

Hình 2 chỉ ra mô hình SVM phi tuyến toàn cục được huấn luyện bởi LibSVM [Chang & Lin, 2011], sử dụng hàm nhân RBF với tham số $\gamma = 10$ và hằng số $C = 10^6$ (dùng để dung hòa độ rộng lề và cực tiểu lỗi).



Hình 2. Mô hình SVM toàn cục (hàm nhân RBF với tham số $\gamma = 10$ và hằng số $C = 10^6$)

A. Huấn luyện mô hình tSVM

Thay vì phải huấn luyện mô hình SVM toàn cục như đã thực hiện bởi các giải thuật SVM chuẩn có độ phức tạp tối thiểu bậc 2 với *m* phần tử của tập huấn luyện, chúng tôi đề xuất giải thuật *t*SVM, xây dựng cây quyết định sử dụng các luật gán nhãn SVM cục bộ cho các nút lá có nhãn của các phần tử không thuần nhất (không cùng lớp).

Quá trình huấn luyện mô hình phân lớp của giải thuật *t*SVM được thực hiện qua 2 bước chính:

*t*SVM sử dụng giải thuật máy học cây quyết định (C4.5 [Quinlan, 1993]) để phân hoạch tập dữ liệu có *m* phần tử của tập huấn luyện thành *k* phân vùng (gọi là nút lá). Quá trình phân hoạch của giải thuật cây quyết định có thể sử dụng điều kiện dừng sớm nếu phân vùng có chứa số phần tử nhỏ hơn giá trị ngưỡng *minobj* thì không thực hiện phân hoạch nữa mà xem phân vùng đó là nút lá.

Với các nút lá có chứa các phần tử có nhãn thuần nhất (cùng lớp) thì giải thuật gán nhãn cho nút lá được dùng khi phân lớp. Các nút lá có chứa các phần tử có nhãn không thuần nhất, giải thuật huấn luyện song song các mô hình SVM phi tuyến, mỗi mô hình để phân lớp cục bộ dữ liệu cho từng nút lá có nhãn không thuần nhất.

Hình 3 trình bày mô hình phân lớp thu được từ giải thuật *t*SVM trên cùng tập dữ liệu sử dụng để huấn luyện mô hình SVM toàn cục đã thực hiện trong hình 2. Trong ví dụ này, *t*SVM phân hoạch tập huấn luyện thành 5 nút lá sử dụng điều kiện dừng sớm là $minobj = 7$. Nút lá D_2 có chứa các phần tử cùng nhãn là hình vuông (□) được gán nhãn là hình vuông (□). Nút lá D_4 có chứa các phần tử cùng nhãn là hình chéo (x) nên được gán nhãn là hình chéo (x). Các nút lá D_1, D_3, D_5 đều chứa các phần tử có nhãn không thuần nhất, nên *t*SVM huấn luyện các mô hình SVM phi tuyến, $lSVM_1$, $lSVM_3$, $lSVM_5$ (θ chính là tham số hàm nhân RBF $\gamma = 10$ và hằng số $C = 10^6$), mỗi mô hình $lSVM_i$ phân lớp dữ liệu cục bộ của một nút lá D_i không thuần nhất.

B. Phân lớp phần tử mới x bằng mô hình tSVM

Mô hình *t*SVM thực hiện phân lớp phần tử mới *x* bằng cách đẩy *x* theo đường dẫn từ nút gốc đến nút lá. Nếu *x* đến nút lá có chứa các phần tử có nhãn thuần nhất thì nhãn của *x* là nhãn của nút lá. Nếu *x* đến nút lá có chứa các phần tử có nhãn không thuần nhất thì nhãn của *x* được dự đoán dựa vào mô hình SVM phi tuyến được huấn luyện để phân lớp cục bộ các dữ liệu huấn luyện trong nút lá đó.

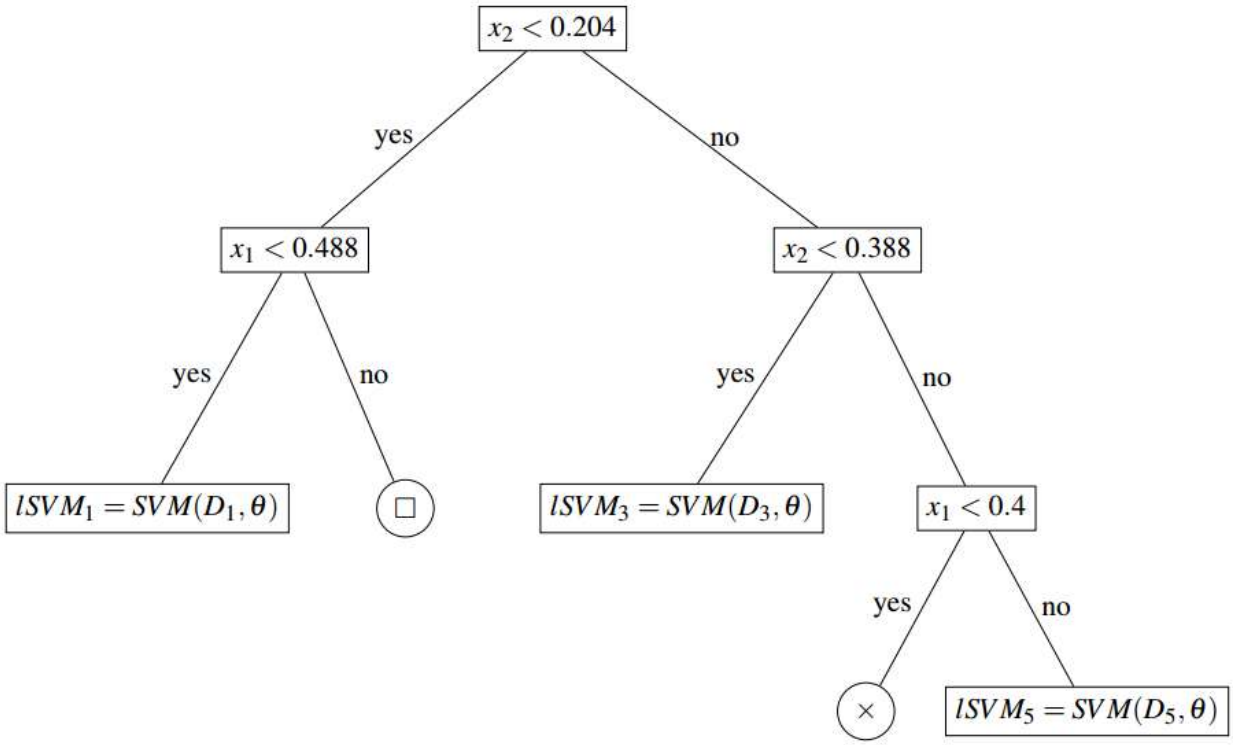
C. Phân tích hiệu quả của giải thuật tSVM

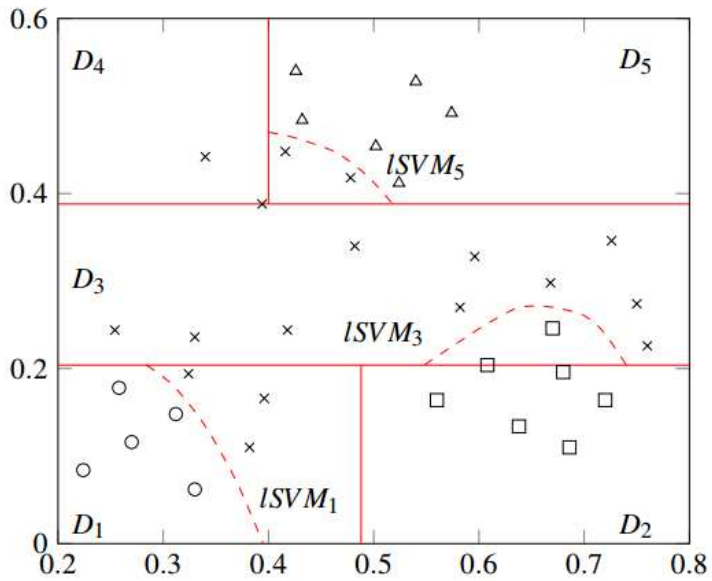
Giả sử tập dữ liệu huấn luyện có *m* phần tử được *t*SVM phân hoạch thành *k* nút lá có kích thước bằng nhau là *m/k* phần tử (hay nói cách khác *minobj = m/k*).

Huấn luyện một mô hình SVM cục bộ cho từng nút lá có chứa *m/k* phần tử có độ phức tạp là $O\left(\frac{m^2}{k^2}\right)$.

Chính vì vậy, độ phức tạp của quá trình huấn luyện song song *k* mô hình SVM cục bộ cho *k* nút lá của giải thuật *t*SVM trên máy tính có P-core là: $O\left(\frac{m^2}{Pk}\right) = O\left(\frac{m}{P}\min obj\right)$.

So với huấn luyện mô hình SVM toàn cục của các giải thuật SVM chuẩn, *t*SVM có độ phức tạp thấp hơn $Pk = P\frac{m}{\min obj}$ lần [1].





Hình 3. Mô hình *t*SVM sử dụng điều kiện dừng sớm *minobj=7* (hàm nhân RBF với tham số $\gamma = 10$ và hằng số $C = 10^6$)

*t*SVM sử dụng tham số điều kiện dừng sớm quá trình phân hoạch của cây quyết định *minobj* để dung hòa giữa khả năng tổng quát hóa (độ chính xác khi dự đoán nhãn của phần tử mới) và chi phí tính toán của giải thuật huấn luyện mô hình.

Nếu *minobj* được đặt quá nhỏ, khi so với mô hình SVM toàn cục, *t*SVM giảm thời gian huấn luyện rất lớn, thậm chí các nút lá hầu hết đều chứa các phần tử có nhãn thuần nhất (cây quyết định thông thường), tính tổng quát mô hình thấp, *t*SVM cho độ chính xác thấp khi phân lớp.

[1] Chú ý rằng độ phức tạp *t*SVM ở đây chưa bao gồm độ phức tạp của quá trình phân hoạch tập dữ liệu huấn luyện của cây quyết định, tuy nhiên do độ phức tạp của quá trình phân hoạch rất nhỏ so với độ phức tạp của việc giải bài toán quy hoạch toàn phương của giải thuật SVM.

Nếu *minobj* được đặt quá lớn, khi so với mô hình SVM toàn cục, *t*SVM giảm thời gian huấn luyện không nhiều, tuy nhiên mô hình *t*SVM có tính tổng quát cao, *t*SVM cho độ chính xác cao khi phân lớp. Thậm chí nếu đặt *minobj* = *m* thì *t*SVM chính là mô hình SVM toàn cục.

Điều này cho thấy được tham số *minobj* trong *t*SVM cần được đặt đủ lớn (từ 200 đến 1000 [Bottou & Vapnik, 1992] tùy theo từng tập dữ liệu) để dung hòa được độ chính xác khi phân lớp và giảm được độ phức tạp khi huấn luyện.

IV. KẾT QUẢ THỰC NGHIỆM

A. Cài đặt chương trình

Chúng tôi tiến hành đánh giá hiệu quả của giải thuật đề xuất *t*SVM cho bài toán phân lớp. Chúng tôi đã cài đặt giải thuật *t*SVM bằng ngôn ngữ C/C++ sử dụng chương trình C4.5 [Quinlan, 1993], thư viện SVM chuẩn, LibSVM [Chang & Lin, 2011], thư viện OpenMP (giao diện lập trình song song C/C++ trên máy tính đa nhân sử dụng bộ nhớ chia sẻ). Chúng tôi thực hiện so sánh hiệu quả phân lớp của giải thuật *t*SVM và LibSVM, dựa trên hai tiêu chí: độ chính xác phân lớp và thời gian huấn luyện.

Tất cả các thí nghiệm được chạy trên máy tính cá nhân, cài hệ điều hành Linux Fedora 20, bộ vi xử lý Intel® Core i7-4790, 3.6 GHz, 4 nhân và bộ nhớ RAM 32 GB.

B. Chuẩn bị tập dữ liệu

Thí nghiệm được thực hiện trên 4 tập dữ liệu UCI [Lichman, 2013] và 3 bộ dữ liệu ký tự viết tay chuẩn hai bộ cũ: USPS [Lecun et al., 1989], MNIST [Lecun et al., 1998], một bộ dữ liệu ký tự viết tay mới [van der Maaten, 2009] và tập dữ liệu phân lớp ảnh [Geusebroek et al., 2005]. Bảng 1 trình bày mô tả của các tập dữ liệu thực nghiệm. Nghi thức kiểm tra đánh giá được chỉ ra trong cột cuối của bảng. Dữ liệu đã được chia thành hai tập: huấn luyện (Trn) và kiểm tra (Tst). Chúng tôi sử dụng tập huấn luyện để huấn luyện các mô hình SVM. Sau đó, sử dụng các mô hình phân lớp thu được để phân lớp dữ liệu trong tập kiểm tra.

Bảng 1. Mô tả các tập dữ liệu thực nghiệm

ID	Dataset	Số phần tử	Số thuộc tính	Số lớp	Nghi thức kiểm tra
1	Opt. Rec. of Handwritten Digits	5620	64	10	3832 Trn - 1797 Tst
2	Letter	20000	16	26	13334 Trn - 6666 Tst
3	Isolet	7797	617	26	6238 Trn - 1559 Tst
4	USPS Handwritten Digit	9298	256	10	7291 Trn - 2007 Tst
5	A New Benchmark for HCR	40133	3136	36	36000 Trn - 4133 Tst
6	MNIST	70000	784	10	60000 Trn - 10000 Tst
7	ALOI	108000	128	1000	72000 Trn - 36000 Tst
8	Forest Cover Types	581012	54	7	400000 Trn - 181012 Tst

C. Điều chỉnh tham số

Chúng tôi đề xuất sử dụng hàm nhân RBF trong cả *t*SVM và LibSVM vì tính tổng quát và tính hiệu quả của nó [Chang & Lin, 2011]. Chúng tôi cũng điều chỉnh siêu tham số γ của hàm nhân RBF và hằng số C (tham số dung hòa lỗi và độ rộng của lề SVM) để có được kết quả cao nhất. Hơn nữa giải thuật *t*SVM của chúng tôi có sử dụng thêm một tham số điều kiện dừng sớm quá trình phân hoạch của cây quyết định *minobj* được đặt bằng *1000* phần tử (nhằm tạo ra một sự dung hòa giữa độ chính xác của mô hình phân lớp và chi phí tính toán). Bảng 2 trình bày các siêu tham số được sử dụng cho *t*SVM và LibSVM.

Bảng 2. Các tham số của *t*SVM và LibSVM

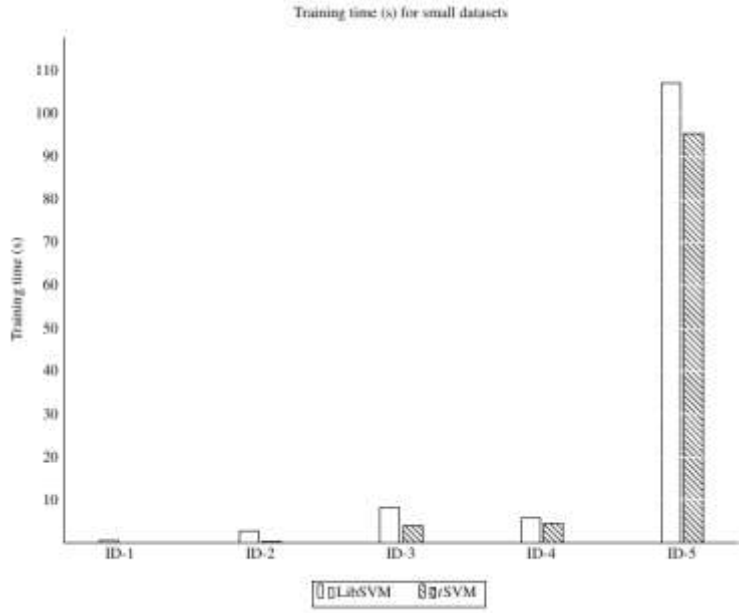
ID	Dataset	γ	C	Minobj
1	Opt. Rec. of Handwritten Digits	0.0001	100000	1000
2	Letter	0.0001	100000	1000
3	Isolet	0.0001	100000	1000
4	USPS Handwritten Digit	0.0001	100000	1000
5	A New Benchmark for HCR	0.001	100000	1000
6	MNIST	0.05	100000	1000
7	ALOI	0.01	100000	1000
8	Forest Cover Types	0.0001	100000	1000

D. Kết quả phân lớp

Kết quả phân lớp của LibSVM và *t*SVM trên 8 tập dữ liệu được cho trong bảng 3 và các hình 4, 5 và 6. Như mong đợi, giải thuật *t*SVM của chúng tôi có thời gian huấn luyện ngắn hơn nhiều so với giải thuật LibSVM. Về tiêu chí độ chính xác phân lớp, giải thuật của chúng tôi cho kết quả có thể so sánh được với giải thuật LibSVM.

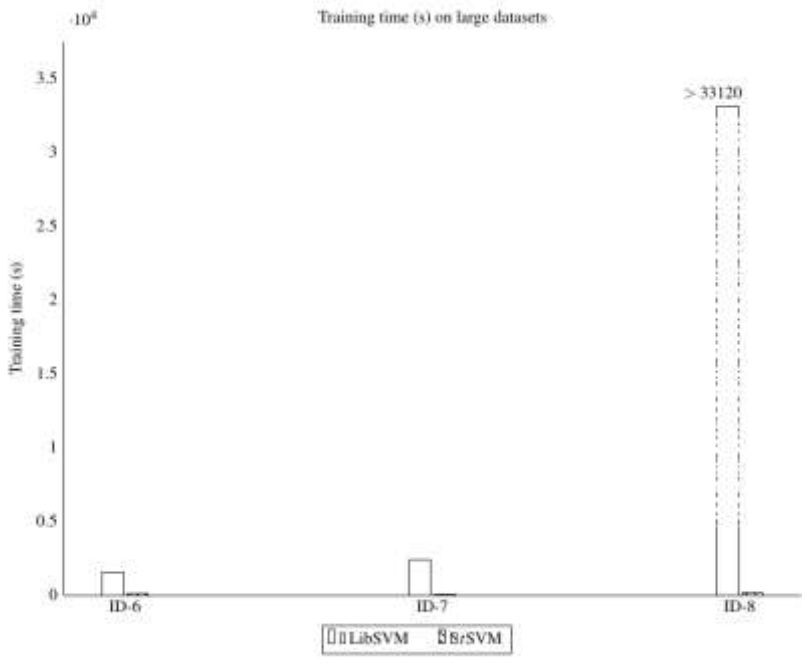
Bảng 3. So sánh hiệu quả của các phương pháp theo độ chính xác (%) và thời gian huấn luyện (giây)

ID	Dataset	Độ chính xác (%)		Thời gian huấn luyện (giây)	
		LibSVM	*t*SVM	LibSVM	*t*SVM
1	Opt. Rec. of Handwritten Digits	98.33	96.99	0.58	0.12
2	Letter	97.40	95.65	2.87	0.42
3	Isolet	96.47	95.38	8.37	3.98
4	USPS Handwritten Digit	96.86	95.02	5.88	4.62
5	A New Benchmark for HCR	95.14	92.72	107.07	95.37
6	MNIST	98.37	98.24	1531.06	124.48
7	ALOI	95.16	93.17	2400	30
8	Forest Cover Types	NA	96.73	NA	179.84



Hình 4. So sánh thời gian huấn luyện của LibSVM và *t*SVM trên 5 tập dữ liệu nhỏ

Với 5 tập dữ liệu nhỏ đầu tiên, cải tiến về mặt thời gian của *t*SVM là không đáng kể. Tuy nhiên với các tập dữ liệu lớn, *t*SVM tăng tốc đáng kể quá trình huấn luyện.

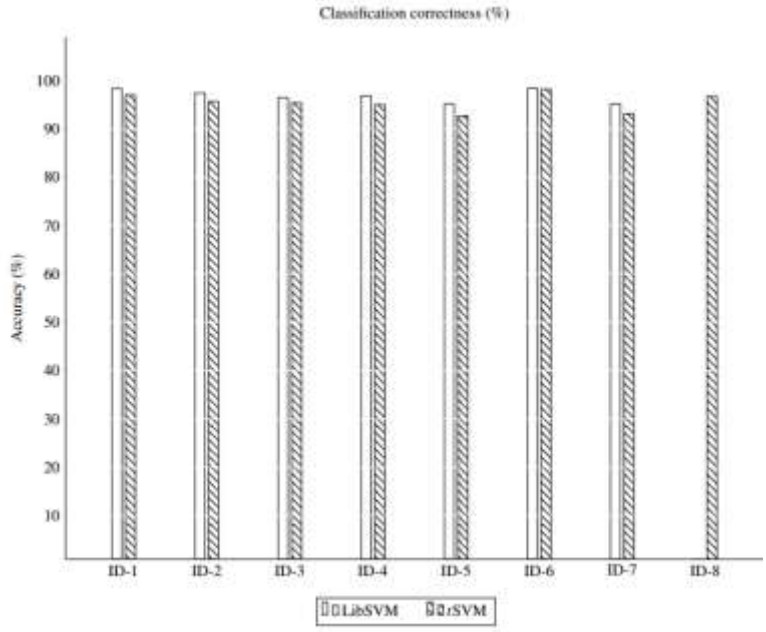


Hình 5. So sánh thời gian huấn luyện của LibSVM và *t*SVM trên 3 tập dữ liệu lớn

Xét thời gian huấn luyện mô hình phân lớp cho tập dữ liệu MNIST, giải thuật *t*SVM nhanh hơn LibSVM đến 12.30 lần.

Xét thời gian huấn luyện mô hình phân lớp cho tập dữ liệu ảnh ALOI. Tập dữ liệu này rất đặc biệt do có số phần tử lớn và số lớp của đối tượng là 1000, nếu huấn luyện mô hình đa lớp bằng cách xây dựng mô hình phân lớp nhị phân cho từng cặp lớp như LibSVM thì cần đến 499500 mô hình nhị phân. Trong khi đó, *t*SVM đã phân hoạch tập huấn luyện có 1000 lớp thành các phân vùng con, mỗi phân vùng có số lớp ít hơn so với tập huấn luyện đầy đủ. Kết quả là giải thuật *t*SVM nhanh hơn LibSVM đến 80 lần.

Đặc biệt, với tập dữ liệu Forest cover type (được xem như là tập dữ liệu khó đối với SVM phi tuyến [Yu et al., 2003], [Do & Poulet, 2004], LibSVM chạy đến 23 ngày vẫn chưa hoàn thành quá trình huấn luyện mô hình. Trong khi đó, *t*SVM thực hiện huấn luyện trong 179.8 giây (3 phút) và cho độ chính xác phân lớp là 96.73%.



Hình 6. So sánh độ chính xác phân lớp của LibSVM và *t*SVM trên 8 tập dữ liệu

V. CÁC NGHIÊN CỨU LIÊN QUAN

Để cải tiến việc huấn luyện giải thuật máy học SVM cho các tập dữ liệu lớn, các công trình nghiên cứu trong [Boser et al., 1992], [Chang & Lin, 2011], [Osuna et al., 1997], [Platt, 1998] đã chia bài toán quy hoạch toàn phương gốc thành các bài toán con để giải quyết.

Nghiên cứu của chúng tôi liên quan đến các giải thuật huấn luyện mô hình phân lớp phi tuyến cục bộ. Nhóm các giải thuật huấn luyện mô hình phân cấp cho vấn đề phân lớp, thực hiện phân lớp qua 2 bước chính: gom nhóm tập dữ liệu huấn luyện thành *k* nhóm (clusters); bước tiếp theo là huấn luyện các mô hình phân lớp cục bộ cho từng nhóm. Đề xuất của [Jacobs et al., 1991] sử dụng giải thuật cực đại kỳ vọng (Expectation-Maximization, EM [Dempster et al., 1977]) để chia dữ liệu thành *k* nhóm không tách rời (joint clusters); và huấn luyện các mô hình mạng nơ-ron (Neural Network) để phân lớp dữ liệu cục bộ cho từng nhóm. Nghiên cứu của [Collobert et al., 2002] chỉ khác với nghiên cứu của [Jacobs et al., 1991] là xây dựng *k* mô hình SVM [Vapnik, 1995] cục bộ. CSVM [Gu & Han, 2013] sử dụng giải thuật *k*-means [MacQueen, 1967] để phân hoạch tập dữ liệu huấn luyện thành *k* nhóm tách biệt; sau đó huấn luyện các mô hình SVM tuyến tính có trọng số từ các nhóm dữ liệu. Nghiên cứu gần nhất là giải thuật *k*SVM [Do, 2015] và *kr*SVM [Do & Poulet, 2015] xây dựng song song *k* mô hình SVM phi tuyến cục bộ trên máy tính đa nhân, bộ nhớ chia sẻ, để phân lớp cục bộ *k* nhóm, được phân hoạch từ tập dữ liệu huấn luyện với *k*-means. DTSVM [Chang et al., 2010] sử dụng giải thuật học cây quyết định [Breiman et al., 1984], [Quinlan, 1993] để phân hoạch tập dữ liệu huấn luyện thành các phân vùng tách rời nhau và xây dựng các mô hình SVM cục bộ cho các phân vùng. Các giải thuật này đều nhằm cải tiến tốc độ huấn luyện mô hình phân lớp.

Nhóm các nghiên cứu sau đây thực hiện huấn luyện mô hình phân lớp từ *k* láng giềng của phần tử mới *x* khi phân lớp. Mô hình học của [Bottou & Vapnik, 1992] tìm *k* láng giềng của phần tử mới *x*, thực hiện huấn luyện mô hình mạng nơ-ron để phân lớp *k* láng giềng này, dùng mô hình mạng nơ-ron cục bộ thu được để phân lớp phần tử *x*. [Vincent & Bengio, 2001] đề xuất giải thuật huấn luyện *k* siêu phẳng cục bộ (*k*-local hyperplane). Các nghiên cứu khác về giải thuật SVM cục bộ sử dụng các chiến lược khác nhau cho tìm kiếm *k* láng giềng, bao gồm SVM-*k*NN [Zhang et al., 2006] sử dụng các độ đo khoảng cách khác nhau, ALH [Yang & Kecman, 2008] sử dụng khoảng cách có trọng số và chọn lọc các đặc trưng quan trọng, FaLK-SVM [Segata & Blanzieri, 2010] tăng tốc quá trình tìm *k* láng giềng sử dụng cây chỉ mục cover tree [Beygelzimer et al., 2006].

Nghiên cứu và phân tích lý thuyết về các giải thuật huấn luyện mô hình phân lớp cục bộ được thảo luận trong [Bottou & Vapnik, 1992]. Nghiên cứu chỉ ra có sự dung hòa giữa khả tăng tổng quát của mô hình phân lớp cục bộ và số phần tử được sử dụng để huấn luyện một mô hình phân lớp cục bộ. Kích thước của tập dữ liệu cục bộ được dùng như một tham số tự do bổ sung để điều khiển tính cục bộ và khả năng tổng quát của mô hình phân lớp cục bộ.

VI. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chúng tôi vừa trình bày một giải thuật mới *t*SVM cho phép huấn luyện nhanh mô hình máy học véc-tơ hỗ trợ cục bộ để phân lớp dữ liệu phi tuyến chính xác cho các tập dữ liệu lớn. Giải thuật *t*SVM sử dụng máy học cây quyết định để phân hoạch nhanh tập dữ liệu lớn thành *k* nút lá. Với các nút lá có chứa các phần tử có nhãn không thuần nhất, giải thuật *t*SVM huấn luyện song song các mô hình SVM phi tuyến, mỗi mô hình để phân lớp cục bộ dữ liệu cho từng nút lá có nhãn không thuần nhất. Kết quả thực nghiệm trên các tập dữ liệu của UCI và 3 tập dữ liệu nhận dạng ký tự viết tay và tập dữ liệu phân lớp ảnh cho thấy giải thuật *t*SVM cho kết quả phân lớp nhanh, chính xác khi so sánh với giải thuật SVM chuẩn như LibSVM. Một ví dụ về tính hiệu quả của giải thuật *t*SVM là: thời gian huấn luyện trên tập dữ liệu Forest Cover Types (400.000 phần tử, 54 chiều, 7 lớp) chỉ có 179.8 giây và độ chính xác phân lớp tổng thể 96.73%.

Trong thời gian tới, chúng tôi dự định sẽ cung cấp thêm các thực nghiệm trên những tập dữ liệu lớn khác nữa và so sánh hiệu quả của *t*SVM với các giải thuật học máy khác. Một trong những hướng phát triển của nghiên cứu này trong tương lai là cải tiến độ chính xác phân lớp của *t*SVM.

TÀI LIỆU THAM KHẢO

[1] Beygelzimer, A., Kakade, S., Langford, J.: "Cover trees for nearest neighbor", in proc. of the 23rd intl conf. on Machine learning, pp. 97-104, 2006.

[2] Boser, B., Guyon, I., Vapnik, V., "An training algorithm for optimal margin classifiers", In proceedings of 5th ACM Annual Workshop on Computational Learning Theory, pp.144-152, 1992.

[3] Bottou, L., Vapnik, V., "Local learning algorithms", *Neural Computation* 4(6): 888-900, 1992.

[4] Breiman, L., Friedman, J .H., Olshen, R. A. and Stone, C.: "*Classification and Regression Trees*", Wadsworth International, 1984.

[5] Chang, C. C., Lin, C. J., "LIBSVM: a library for support vector machines", *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 27, pp.1-27, 2011 http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[6] Chang, F., Guo, C. Y., Lin, X. R., Lu, C. J.: "Tree decomposition for largescale SVM problems", *Journal of Machine Learning Research* 11:2935-2972, 2010.

[7] Collobert, R., Bengio, S., Bengio, Y.: "A parallel mixture of SVMs for very large scale problems", *Neural Computation* 14(5):1105-1114, 2002.

[8] Cristianini, N., Shawe-Taylor, J., "*An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*", Cambridge University Press, New York, NY, USA, 2000.

[9] Dempster, A. P., Laird, N. M., Rubin, D. B.: "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the royal statistical society*, series B, vol.39(1):1-38, 1977

[10] Do, T. N., Poulet, F.: "Random local SVMs for classifying large datasets", in proc. of Intl Conf. on Future Data and Security Engineering 2015 (FDSE 2015), Springer, 2015, pp. 3-15.

[11] Do, T. N.: "Non-linear classification of massive datasets with a parallel algorithm of local support vector machines", in Advanced Computational Methods for Knowledge Engineering Studies in Computational Intelligence, Springer, 2015, pp. 231-241.

[12] Geusebroek, J. M., Burghouts, G. J., Smeulders, A. W. M.: The amsterdam library of object images. *Intl Journal Computer Vision* 61(1): 103–112, 2005.

[13] Gu, Q., Han, J.: "Clustered support vector machines", in proc. of the Sixteenth Intl Conf. on Artificial Intelligence and Statistics, vol. 31, pp.307-315, 2013.

[14] Guyon, I., Web page on svm applications, 1999, http://www.clopinet.com/isabelle/Projects/SVM/applist.html.

[15] Jacobs, R. A., Jordan, M. I., Nowlan, S. J., Hinton, G. E.: "Adaptive mixtures of local experts", *Neural Computation* vol.3(1):79-87, 1991.

[16] LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., Jackel, L.: Backpropagation applied to handwritten zip code recognition. *Neural Computation* 1(4):541–551, 1989.

[17] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11): 2278–2324, 1998.

[18] Lichman, M.: UCI machine learning repository, 2013, http://archive.ics.uci.edu/ml.

[19] MacQueen, J.: "Some methods for classification and analysis of multivariate observations", in proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press 1, pp.281-297, 1967.

[20] Osuna, E., Freund, R., Girosi, F., "An improved training algorithm for support vector machines", *Neural Networks for Signal Processing VII*, J. Principe, L. Gile, N. Morgan, and E. Wilson Eds, pp.276-285, 1997.

[21] Platt, J.: "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines", Microsoft Research Technical Report MSR-TR-98-14, 1998.

[22] Quinlan, J. R.: "*C4.5: Programs for Machine Learning*", Morgan Kaufmann, 1993.

[23] Segata, N., Blanzieri, E.: "Fast and scalable local kernel machines", *Journal Machine Learning Research 11*, pp.1883-1926, 2010.

[24] Van der Maaten, L.: A new benchmark dataset for handwritten character recognition, 2009, http://homepage.tudelft.nl/19j49/ Publications files/characters.zip.

[25] Vapnik, V.: "*The Nature of Statistical Learning Theory*", Springer-Verlag, 1995.

[26] Vincent, P., Bengio, Y.: "K-local hyperplane and convex distance nearest neighbor algorithms", In Advances in Neural Information Processing Systems, The MIT Press, pp.985-992, 2001.

[27] Wu X. and Kumar V.: "*Top 10 Algorithms in Data Mining*", Chapman & Hall/CRC, 2009.

[28] Yang, T., Kecman, V.: "Adaptive local hyperplane classification", *Neurocomputing* vol.71(13-15): 3001-3004, 2008.

[29] Yu, H., Yang, J., Han, J.: "Classifying large data sets using SVMs with hierarchical clusters", In proceedings of the ACM SIGKDD Intl. Conf. on KDD, ACM, pp.306-315, 2003.

[30] Zhang, H., Berg, A., Maire, M., Malik, J.: "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition", In IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 2., pp. 2126-2136, 2006.

*t*SVM ALGORITHM FOR NON-LINEAR CLASSIFICATION OF VERY LARGE DATASETS

Thanh Nghi Do, Nguyen Khang Pham, Minh Thu Tran Nguyen, Huu Hoa Nguyen

ABSTRACT— *In this paper, we present the new support vector machines algorithm, called tSVM for effectively non-linear classification of large datasets. The tSVM algorithm performs the training task of large datasets with two main steps. The first one is to partition the full dataset into k terminal-nodes, and then the second one is to learn in parallel local SVM models for classifying impurity terminal-nodes with mixture of labels. The numerical test results on 4 datasets from UCI repository, 3 benchmarks of handwritten letters recognition and a color image collection of one-thousand small objects show that our tSVM algorithm is efficient compared to the standard SVM (LibSVM) in terms of training time and accuracy for dealing with large datasets.*