Kỷ yếu Hội nghị Quốc gia lần thứ VIII về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR); Hà Nội, ngày 9-10/7/2015
DOI: 10.15625/vap.2015.000193

PHÂN LỚP PHI TUYẾN DỮ LIỆU LỚN VỚI GIẢI THUẬT SONG SONG CHO MÔ HÌNH MÁY HỌC VÉCTƠ HỖ TRỢ CỤC BỘ

Đỗ Thanh Nghị[1], Phạm Nguyên Khang[1]
[1] Khoa Công nghệ Thông tin và Truyền thông, Trường Đại học Cần Thơ
dtnghi@cit.ctu.edu.vn, pnkhang@cit.ctu.edu.vn

TÓM TẮT *- Trong bài báo này, chúng tôi đề xuất một mô hình máy học véc-tơ hỗ trợ cục bộ mới dựa trên máy học véc-tơ hỗ trợ (SVM) và giải thuật gom cụm dữ liệu (clustering), gọi là kSVM, dùng để phân lớp phi tuyến dữ liệu lớn. kSVM sử dụng giải thuật k-means để phân hoạch dữ liệu thành k cụm (cluster). Sau đó, với mỗi cụm kSVM huấn luyện một mô hình SVM phi tuyến dùng để phân lớp dữ liệu của cụm. Việc huấn luyện các mô hình SVM trên từng cụm hoàn toàn độc lập với nhau, vì thế có thể được thực hiện song song trên các máy tính multi-core. Giải thuật song song để huấn luyện kSVM nhanh hơn rất nhiều so với các giải thuật SVM chuẩn như LibSVM, SVMLight trong bài toán phân lớp phi tuyến dữ liệu lớn. Kết quả thực nghiệm trên các tập dữ liệu của UCI và 3 tập dữ liệu nhận dạng ký tự viết tay cho thấy đề xuất của chúng tôi hiệu quả hơn mô hình SVM chuẩn.*

Từ khóa *- Máy học véctơ hỗ trợ, máy học véc-tơ hỗ trợ cục bộ, phân lớp phi tuyến dữ liệu lớn.*

I. GIỚI THIỆU

Trong những năm gần đây, mô hình máy học véctơ hỗ trợ (SVM) [1] và các phương pháp dựa trêm hàm nhân (kernel-based methods) đã cho thấy được tính hợp lý của nó trong các bài toán phân toán, hồi quy và phát hiện phần tử mới. 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 [2]. Các phương pháp này đã trở thành các công phân tích dữ liệu phổ biến. Mặc dù sở hữu nhiều ưu điểm, SVM vẫn thích hợp khi xử lý dữ liệu lớn. Lời giải của bài toán SVM là kết quả bài toán quy hoạch toàn phương (QP), vì thế độ phức tạp tính toán của các giải thuật SVM ít nhất là $O(m^2)$ với m là số phần tử trong tập huấn luyện. Hơn nữa, do yêu cầu bộ nhớ lớn nên việc sử dụng SVM trở nên khó khăn hơn khi đối mặt với dữ liệu lớn. Điều này dẫn đến yêu cầu mở rộng khả năng xử lý (scale up) của các giải thuật học để có thể xử lý các tập dữ liệu lớn trên các máy tính cá nhân (PCs).

Chúng tôi đầu tư đề xuất một giải thuật song song cho bài toán SVM cục bộ, gọi là kSVM, nhằm giải quyết bài toán phân lớp phi tuyến các tập dữ liệu lớn. Thay vì xây dựng một mô hình SVM toàn cục như các giải thuật cổ điển (rất khoa khi xử lý dữ liệu lớn), giải thuật kSVM xây dựng một tập các mô hình SVM cục bộ. Điều này có thể được thực hiện rất dễ dàng bằng cách áp dụng giải thuật SVM chuẩn trên các tập dữ liệu nhỏ. Giải thuật kSVM thực hiện việc huấn luyện qua hai giai đoạn. Trong giai đoạn đầu, sử dụng giải thuật k-means [3] phân hoạch tập dữ liệu huấn luyện thành k cụm (cluster). Trong giai đoạn thứ hai, với mỗi cụm dữ liệu xây dựng một mô hình SVM phi tuyến để phân lớp dữ liệu cho cụm.

II. MÁY HỌC VÉCTƠ HỖ TRỢ

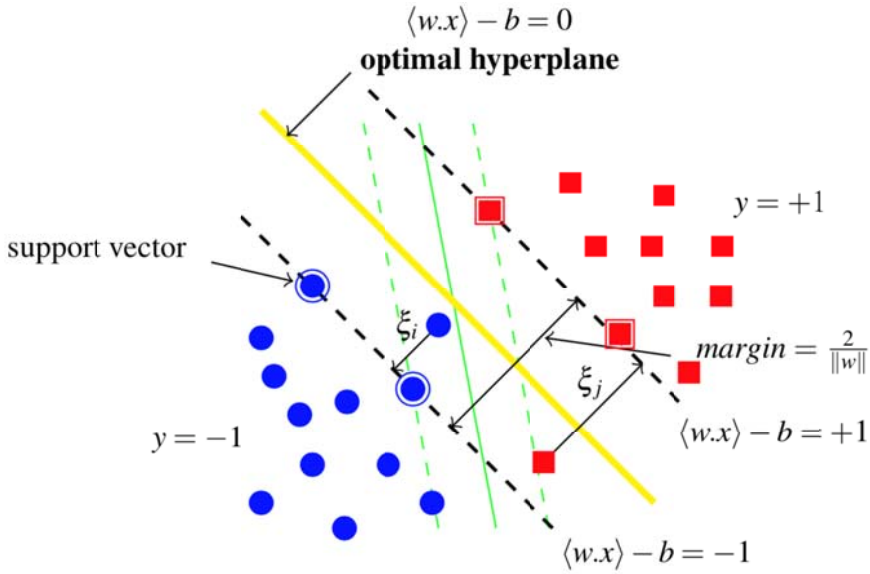
Xét bài toán phân lớp nhị phân như Hình 1, với m phần tử x_i (i = 1, 2, …, m) trong không gian n chiều, R^n. Mỗi phần tử có nhãn tương ứng $y_i \in \{-1, +1\}$. Với bài toán này, giải thuật SVM [1] cố gắng tìm một siêu phẳng tối ưu (biểu diễn bằng pháp véctơ $w \in R^n$ và độ lệch $b \in R$) tách các phần tử thành hai phần tương ứng với nhãn của chúng. Siêu phẳng tối ưu là siêu phẳng cách xa 2 lớp nhất. Bài toán này tương đương với việc cực đại hoá khoảng cách hay còn gọi là lề (margin) giữa hai siêu phẳng hỗ trợ của mỗi lớp ($x.w - b = 1$ đối với lớp +1 và $w.x$ – b = -1 đối với lớp -1). Khoảng cách giữa hai siêu phẳng hỗ trợ bằng $2/\|w\|$ trong đó $\|w\|$ là độ lớn (2-norm) của pháp véctơ w. Trường hợp dữ liệu không khả tách tuyến tính (linearly separable), ta xem mỗi phần tử nằm sai phía so với mặt phẳng hỗ trợ tương ứng với lớp của chúng là lỗi, khoảng cách từ phần tử lỗi đến siêu phẳng hỗ trợ được ký hiệu z_i ($z_i \geq 0$). Vì thế, bộ phân lớp SVM phải đồng thời cực đại hoá lề và cực tiểu hoá lỗi. Mô hình SVM chuẩn mô hình hoá bài toán tối ưu này về bài toán quy hoạch toàn phương (1).

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

với ràng buộc:

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

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 $K\langle x_i, x_j\rangle = \langle x_i \bullet x_j\rangle$.



Hình 1. Tách tuyến tính các phần tử thành hai lớp.

Giải bài toán quy hoạch toàn phương (1), ta thu được α_i (i = 1, 2, …, m). Các phần tử x_i tương ứng với $\alpha_i > 0$ được gọi là các véctơ hỗ trợ. Chỉ cần các véctơ này ta có thể dựng lại được các siêu phẳng hỗ trợ và tìm được siêu phẳng phân lớp tối ưu (nằm chính giữa hai siêu phẳng hỗ trợ). Việc phân lớp phần tử mới x với mô hình SVM được cho bởi:

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

Các biến thể của giải thuật SVM sử dụng các hàm phân lớp khác nhau [8]. Để có thể có hàm phân lớp khác, ta không cần thay đổi giải thuật mà chỉ cần thay đổi hàm nhân tuyến tính bằng các hàm nhân khác. Bằng cách này ta thu được các mô hình phân lớp dựa trên các véctơ hỗ trợ khác nhau. Hai hàm nhân phi tuyến phổ biến là:

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

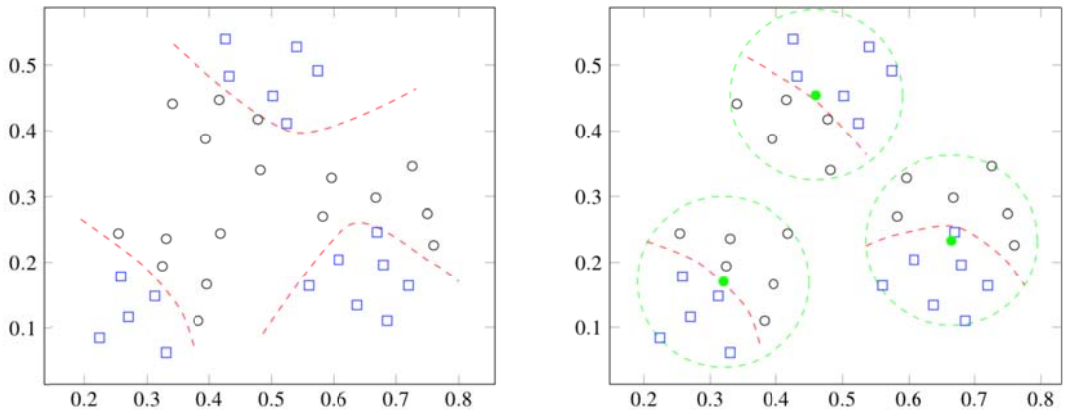
Mô hình máy học SVM cho kết quả cao, ổn định, chịu đựng nhiễu tốt và phù hợp với các bài toán như: phân lớp, hồi quy và phát hiện phần tử ngoại lai. Nhiều ứng dụng thành công của SVM đã được công bố bao gồm nhiều lĩnh vực: nhận dạng ảnh, phân loại văn bản và sinh-tin học [2].

III.GIẢI THUẬT SONG SONG CHO MÁY HỌC VÉC-TƠ HỖ TRỢ CỤC BỘ

Nghiên cứu trong [9] đã chỉ ra rằng độ phức tạp tính toán của SVM ít nhất là $O\left(m^2\right)$ trong đó m là số phần tử trong tập huấn luyện. Điều này làm SVM trở nên khó sử dụng trong các tập dữ liệu lớn. Huấn luyện một mô hình SVM toàn cục trên một tập dữ liệu lớn là một thách thức do độ phức tạp tính toán cao và cần nhiều bộ nhớ.

A. Huấn luyện các mô hình SVM

Giải thuật kSVM của chúng tôi sử dụng giải thuật k-means[3] để phân hoạch tập dữ liệu huấn luyện thành k cụm, và sau đó huấn luyện một mô hình SVM phi tuyến trên mỗi cụm. Hình 2 minh hoạ kết quả của mô hình SVM toàn cục (hình trái) và 3 mô hình SVM cục bộ (hình phải), sử dụng hàm nhân RBF với $\gamma = 10$ và hằng số dung hoà $C = 10^6$.



Hình 2. Mô hình SVM toàn cục (trái) và các mô hình SVM cục bộ (phải).

Bây giờ ta sẽ xem xét độ phức tạp của việc xây dựng k mô hình SVM cục bộ với giải thuật kSVM. Toàn bộ tập dữ liệu huấn luyện gồm *m* phần tử được phân hoạch thành *k* cụm (giả sử cân bằng). Vì thế, mỗi cụm có khoảng m/k phần tử. Độ phức tạp tính toán của *k* mô hình SVM cục bộ là $O\left(k\left(\frac{m}{k}\right)^2\right)=O\left(\frac{m^2}{k}\right)$. Việc phân tích này cho thấy rằng huấn luyện k mô hình SVM cục bộ trong giải thuật kSVM nhanh hơn huấn luyện một mô hình SVM toàn cục (độ phức tạp $O\left(m^2\right)$.

Cần phải chú ý rằng tham số k được sử dụng trong mô hình kSVM để điều chỉnh sự dung hoà giữa khả năng tổng quát hoá và chi phí tính toán của giải thuật. Trong [10, 11, 12], Vapnik đã đề cập đến sự dung hoà giữa khả năng tổng quát hoá và số phần tử trong tập học. Trong ngữ cảnh của mô hình kSVM (k SVM cục bộ), điều này có thể hiểu như sau:

- Nếu k lớn, thời gian huấn luyện của giải thuật kSVM giảm đáng kể (độ phức tạp của kSVM là $O\left(\frac{m^2}{k}\right)$) và kích thước của các cụm (cluster) nhỏ. Tính cục bộ sẽ tăng và khả năng tổng quát hoá thấp.
- Nếu k nhỏ, thời gian huấn luyện của giải thuật kSVM giảm không đáng kể. Tuy nhiên, do kích thước của các cụm lớn nên khả năng tổng quát hoá cao.

Điều này cho thấy rằng ta cần phải điều chỉnh k sao cho kích thước của các cụm đủ lớn (vd: 200 như đề nghị trong [11]). Hơn nữa, do kSVM huấn luyện k mô hình độc lập từ k cụm dữ liệu nên ta có thể song song hoá quá trình huấn luyện khá dễ dàng. Đây là một tính chất rất tuyệt vời của kSVM. Giải thuật kSVM song song tận dụng ưu điểm của các hệ thống tính toán hiệu năng cao như máy tính đa nhân hay hệ thống tính toán lưới. Việc cài đặt giải thuật kSVM song song đơn giản nhất là sử dụng mô hình lập trình đa xử lý sử dụng bộ nhớ chia sẻ openMPI [13] trên các máy tính đa nhân. Các bước cơ bản của quá trình huấn luyện kSVM song song được mô tả trong giải thuật 1.

Giải thuật 1: Giải thuật máy học véctơ hỗ trợ cục bộ kSVM

Đầu vào:

- Tập dữ liệu huấn luyện D
- Số mô hình cục bộ k
- Siêu tham số γ
- Hằng số C

Đầu ra:

- K mô hình SVM cục bộ

Bắt đầu

Áp dụng giải thuật gom cụm k-means lên tập D
thu được k cụm $D_1, D_2, \ldots, D_k$ và các tâm tương ứng $c_1, c_2, \ldots, c_k$
#pragma omp parallel for
for i = 1 **to** k **do**
/* Huấn luyện mô hình SVM cục bộ trên cụm D_i */
$lsvm_i$ = svm(D_i, γ, C)
end
return $kSVM = \{(c_1, lsvm_1), (c_1, lsvm_1), ..., (c_k, lsvm_k)\}$

Kết thúc

B. Phân lớp phần tử mới bằng các mô hình SVM cục bộ

Mô hình $kSVM = \{(c_1, lsvm_1), (c_1, lsvm_1), \ldots, (c_k, lsvm_k)\}$ được dùng để phân lớp dữ liệu mới, x, như sau. Trước hết, ta tìm cụm gần với x nhất (tìm cụm có tâm gần với x nhất).

$$c_{NN} = \underset{c}{\operatorname{argmin}}\, d(x, c) \tag{3}$$

trong đó d(x, c) là khoảng cách từ phần tử x đến tâm của cụm c.

Sau đó, sử dụng mô hình SVM cục bộ $lsvm_{NN}$ (tương ứng với c_{NN}) để dự báo lớp của x.

$$predict(x, kSVM) = predict(x, lsvm_{NN}) \tag{4}$$

IV. ĐÁNH GIÁ

Chúng tôi quan tâm đến hiệu quả của giải thuật SVM cục bộ song song được đề xuất (gọi là kSVM) cho bài toán phân lớp. Chúng tôi đã cài đặt giải thuật kSVM bằng ngôn ngữ C++ sử dụng thư viện OpenMP [13]. Để so sánh, chúng tôi sử dụng thư viện SVM chuẩn libVM [14]. Đánh giá hiệu quả phân lớp được thực hiện trên hai tiêu chí: độ chính xác phân lớp và thời gian huấn luyện. Chúng tôi quan tâm đến việc so sánh hiệu quả giải thuật kSVM và libSVM.

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.

Thí nghiệm được thực hiện trên 4 tập dữ liệu UCI [4] và 3 bộ dữ liệu ký tự viết tay chuẩn hai bộ cũ: USPS [5], MNIST [6] và một bộ dữ liệu ký tự viết tay mới [7]. 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 SVM thu được để phân lớp dữ liệu trong tập kiểm tra.

Chúng tôi đề xuất sử dụng hàm nhân RBF trong cả kSVM và SVM chuẩn vì tính tổng quát và tính hiệu quả của nó [15]. Chúng tôi cũng điều chỉnh siêu tham số *gamma* của hàm nhân RBF (hàm nhân RBF của hai phần tử x_i, x_j) và tham số *C* (tham số dung hoà lỗi và độ lớn của lề SVM) để có được kết quả cao nhất. Hơn nữa giải thuật kSVM của chúng tôi có sử dụng thêm một tham số *k*. Chúng tôi đề xuất chọn *k* sao cho mỗi cụm dữ liệu có khoảng *1000* phần tử. Ý tưởng chính là tạo ra một sự dung hoà giữa khả năng tổng quát hoá [12] và chi phí tính toán. Bảng 2 trình bày các siêu tham số được sử dụng cho kSVM và SVM.

Bảng 1. Bảng mô tả 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 Hand. Char. Rec.	40133	3136	36	36000 Trn - 4133 Tst
6	MNIST	70000	784	10	60000 Trn - 10000 Tst
7	Forest Cover Types	581012	54	7	400000 Trn - 181012 Tst

Bảng 2. Các siêu tham số của kSVM và SVM

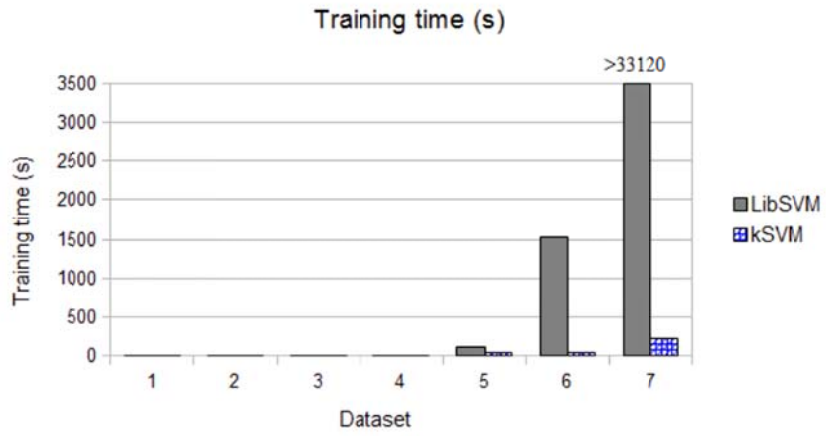
ID	Dataset	γ	C	k
1	Opt. Rec. of Handwritten Digits	0.0001	100000	10
2	Letter	0.0001	100000	30
3	Isolet	0.0001	100000	10
4	USPS Handwritten Digit	0.0001	100000	10
5	A New Benchmark for Hand. Char. Rec.	0.001	100000	50
6	MNIST	0.05	100000	100
7	Forest Cover Types	0.0001	100000	500

Kết quả phân lớp của libSVM và kSVM trên 7 tập dữ liệu được cho trong bảng 3 và các hình 3 và hình 4. Như mong đợi, giải thuật kSVM 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.

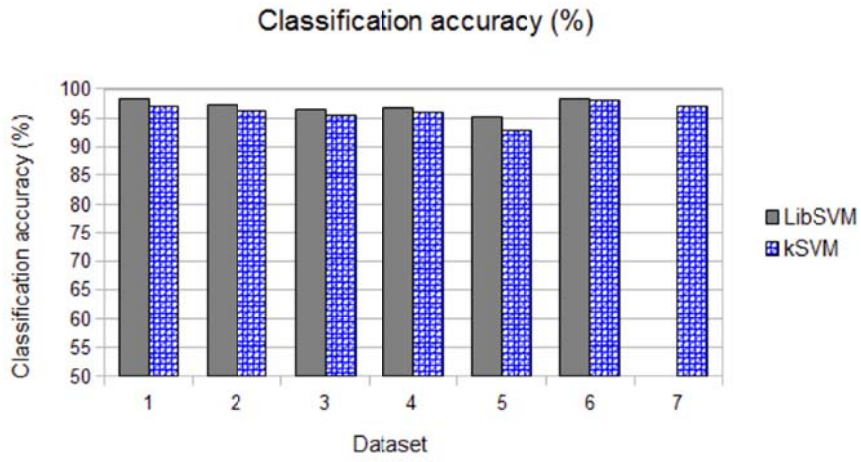
Với 5 tập dữ liệu nhỏ đầu tiên, cải tiến về mặt thời gian của kSVM là không đáng kể. Tuy nhiên với các tập dữ liệu lớn, kSVM tăng tốc đáng kể quá trình huấn luyện. Với tập dữ liệu MNIST, kSVM nhanh hơn libSVM đến 33.64 lần. Đặc biệt, với tập dữ liệu Forest cover type (được xem như là tập dữ liệu khó đối với SVM phi tuyến [16, 17]), libSVM chạy đến 23 ngày vẫn chưa cho ra lời giải. Trong khi đó, kSVM thực hiện huấn luyện trong 223.7 giây và cho độ chính xác phân lớp 97.06%!

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	kSVM	libSVM	kSVM
1	Opt. Rec. of Handwritten Digits	98.33	97.05	0.58	0.21
2	Letter	97.40	96.14	2.87	0.5
3	Isolet	96.47	95.44	8.37	2.94
4	USPS Handwritten Digit	96.86	96.86	5.88	3.82
5	A New Benchmark for Hand. Char. Rec.	95.14	92.98	107.07	35.7
6	MNIST	98.37	98.11	1531.06	45.50
7	Forest Cover Types	NA	97.06	NA	223.7



Hình 3. So sánh thời gian huấn luyện.



Hình 4. So sánh độ chính xác phân lớp.

V. THẢO LUẬN VỀ CÁC CÔNG TRÌNH CÓ LIÊN QUAN

Đề xuất của chúng tôi liên quan đến các giải thuật huấn luyện SVM trên một số khía cạnh. Các phương pháp cải tiến việc huấn luyện SVM đối với dữ liệu lớn bao gồm các phương pháp sử dụng heuristic để phân rã bài toán quy hoạch toàn phương gốc thành nhiều bài toán nhỏ [9, 14, 18, 19].

Mangasarian và các cộng sự đã đề xuất cải biên bài toán SVM để có được các mô hình máy học mới như Lagragian SVM [20], proximal SVM [21], Newton SVM [22]. Mô hình SVM bình phương tối thiểu (Least squares SVM), do Suykens và Vandewalle [23] đề xuất, thay đổi bài toán tối ưu SVM chuẩn thành bài toán SVM khác hiệu quả hơn (về mặt thời gian). Các giải thuật này chỉ cần giải hệ phương trình tuyến tính thay vì phải giải bài toán quy hoạch toàn phương. Điều này làm giảm đáng kể thời gian huấn luyện. Gần đây hơn, phương pháp giảm gradient ngẫu

nhiên (stochastic gradient descent) đã được đề xuất trong [24, 25] để giải bài toán SVM tuyến tính trong trường hợp dữ liệu lớn. Các công trình kế thừa phương pháp này cũng đã được đề xuất trong [17, 26, 27, 28, 29] với mục đích cải tiến độ phức tạp không gian (bộ nhớ sử dụng) bằng phương pháp huấn luyện tăng trưởng. Theo phương pháp này, dữ liệu được chia ra thành nhiều khối, giải thuật lần lượt xử lý từng khối một và cập nhật lời giải. Vì thế không cần phải nạp toàn bộ dữ liệu lên bộ nhớ. Các giải thuật song song và phân tán [27, 29, 30] cho bài toán phân lớp tuyến tính cải tiến hiệu quả huấn luyện trong trường hợp dữ liệu lớn bằng cách chia bài toán thành nhiều bài toán nhỏ và xử lý từng bài toán nhỏ trên nhiều máy tính, trên hệ thống tính toán lưới, trên máy tính đa nhân. Giải thuật SVM song song đề xuất trong [31] sử dụng bộ xử lý đồ hoạ (GPU) để tăng tốc quá trình huấn luyện.

Các giải thuật SVM tích cực đề xuất trong [32, 33, 34, 35] chọn một tập con dữ liệu (gọi là tập tích cực) để xây dựng mô hình thay vì phải huấn luyện trên toàn bộ tập dữ liệu gốc. Các giải thuật SVM [36, 37, 17, 38] sử dụng chiến lược boosting [39, 40] để huấn luyện các tập dữ liệu lớn trên các máy tính cá nhân.

Đề xuất SVM cục bộ của chúng tôi cũng liên quan đến các giải thuật huấn luyện cục bộ. Bài báo đầu tiên [41] đề xuất sử dụng giải thuật EM [42] phân hoạch tập huấn luyện thành k cụm; với mỗi cụm, huấn luyện một mạng nơ-ron. Các giải thuật huấn luyện cục bộ của Bottou và Vapnik [11] tìm k phần tử láng giềng của một phần tử kiểm tra, huấn luyện một mạng nơ-ron trên k phần tử láng giềng và sử dụng mạng này để dự đoán lớp cho phần tử kiểm tra. Các giải thuật k láng giềng sử dụng khoảng cách siêu phẳng cục bộ và k láng giềng sử dụng khoảng cách lồi được đề xuất trong [43]. Gần hơn nữa các giải thuật SVM cục bộ bao gồm SVM-kNN [44], ALH [45], FaLK-SVM [46], LSVM [47], LL-SVM [48, 49], CSVM [50]. Phân tích về mặt lý thuyết của các giải thuật SVM cục bộ như thế [10] đã cho thấy rằng có một sự dung hoà giữa khả năng tổng quát của giải thuật huấn luyện và số phần tử cục bộ. Kích thước của tập láng giềng được dùng như một tham số tự do bổ sung để điều khiển tính cục bộ của các giải thuật học cục bộ.

VI. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chúng tôi vừa mới trình bày một mô hình máy học véctơ hỗ trợ cục bộ mới kSVM và một giải thuật song song dùng để huấn luyện nó. Giải thuật đề nghị hiệu quả hơn các giải thuật SVM chuẩn cho bài toán phân lớp phi tuyến dữ liệu lớn. Giải thuật kSVM bắt đầu bằng việc phân hoạch tập dữ liệu huấn luyện thành k cụm với giải thuật k-means và xây dựng một mô hình SVM phi tuyến cho mỗi cụm dữ liệu. Việc xây dựng các mô hình SVM cho các cụm hoàn toàn độc lập và được thực hiện song song. Kết quả thực nghiệm trên các tập dữ liệu UCI và tập dữ liệu nhận dạng ký tự viết tay cho thấy rằng đề xuất của chúng tôi hiệu quả hơn về thời gian huấn luyện trong khi vẫn giữ được độ chính xác phân lớp khi so sánh với các giải thuật SVM chuẩn. Một ví dụ về tính hiệu quả của giải thuật kSVM là: thời gian huấn luyện trên tập dữ liệu Forest Cover Types (400.000 phần tử, 54 chiều, 7 lớp) chỉ có 223.7 giây và độ chính xác phân lớp tổng thể 97.06%.

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 kSVM 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 kSVM.

VII. TÀI LIỆU THAM KHẢO

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

[2] Guyon, I., Web page on svm applications, 1999, http://www.clopinet.com/isabelle/Projects/-SVM/app-list.html.

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

[4] Asuncion, A., Newman, D., UCI repository of machine learning databases, 2007.

[5] LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., Jackel, L., "Back-propagation applied to handwritten zip code recognition", *Neural Computation*, vol. 1, no. 4, pp.541-551, 1989.

[6] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., "Gradient-based learning applied to document recognition", In Proceedings of the IEEE, vol. 86, no. 11, pp.2278-2324, 1998.

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

[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] Platt, J., "Fast training of support vector machines using sequential minimal optimization", *Advances in Kernel Methods Support Vector Learning*, B. Scholkopf, C. Burges, and A. Smola Eds, pp.185–208, 1999.

[10] Vapnik, V., "Principles of risk minimization for learning theory", In: *Advances in Neural Information Processing Systems* 4, Denver, Colorado, USA, pp.831–838, 1991.

[11] Bottou, L., Vapnik, V., "Local learning algorithms", *Neural Computation*, vol. 4, no. 6, pp.888-900, 1992.

[12] Vapnik,V., Bottou, L., "Local algorithms for pattern recognition and dependencies estimation", *Neural Computation*, vol. 5, no. 6, pp.893-909, 1993.

[13] OpenMP Architecture Review Board: OpenMP application program interface version 3.0, 2008.

[14] 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.

[15] Lin, C., "A practical guide to support vector classification, 2003.

[16] 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.

[17] Do, T. N., Poulet, F., "Towards high dimensional data mining with boosting of psvm and visualization tools", In proceedings of 6th Intl. Conf. on Entreprise Information Systems, pp.36-41, 2004.

[18] 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.

[19] 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.

[20] Mangasarian, O., Musicant, D., "Lagrangian support vector machines", *Journal of Machine Learning Research* 1 pp.161-177, 2001.

[21] Fung, G., Mangasarian, O., "Proximal support vector classifiers". In proceedings of the ACM SIGKDD Intl. Conf. on KDD, ACM, pp.77-86, 2001.

[22] Mangasarian, O., "A finite newton method for classification problems", Technical Report, Data Mining Institute, Computer Sciences Department, University of Wisconsin, 2001.

[23] Suykens, J., Vandewalle, J. "Least squares support vector machines classifiers", *Neural Processing Letters*, vol. 9, no. 3, pp.293–300, 1999.

[24] Shalev-Shwartz, S., Singer, Y., Srebro, N., "Pegasos: Primal estimated sub-gradient solver for svm", In Proceedings of the Twenty-Fourth International Conference Machine Learning, ACM, pp.807-814, 2007.

[25] Bottou, L., Bousquet, O., "The trade offs of large scale learning", In Platt, J., Koller, D., Singer, Y., Roweis, S., eds., *Advances in Neural Information Processing Systems*. Volume 20, pp.161-168, 2008.

[26] Do, T. N., Poulet, F., "Incremental svm and visualization tools for bio-medical data mining", In proceedings of Workshop on Data Mining and Text Mining in Bioinformatics, pp.14-19, 2003.

[27] Do, T. N., Poulet, F., "Classifying one billion data with a new distributed svm algorithm", In proceedings of 4th IEEE Intl. Conf. on Computer Science, Research, Innovation and Vision for the Future, IEEE Press, pp.59-66, 2006.

[28] Fung, G., Mangasarian, O., "Incremental support vector machine classification", In proceedings of the 2nd SIAM Int. Conf. on Data Mining, 2002.

[29] Poulet, F., Do, T. N., "Mining very large datasets with support vector machine algorithms", *Enterprise Information Systems* V, O. Camp, J. Filipe, S. Hammoudi and M. Piattini Eds., pp.177-184, 2004.

[30] Do, T. N., "Parallel multiclass stochastic gradient descent algorithms for classifying million images with very-high-dimensional signatures into thousands classes", *Vietnam J. Computer Science*, vol. 1, no. 2, pp.107-115, 2014.

[31] Do, T. N., Nguyen, V. H., Poulet, F., "Speedup SVM algorithm for massive classification tasks", In Proceedings of ADMA, pp.147-157, 2008.

[32] 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.

[33] Do, T. N., Poulet, F., "Mining very large datasets with svm and visualization", In proceedings of 7th Intl. Conf. on Entreprise Information Systems, pp.127-134, 2005.

[34] Boley, D., Cao, D., "Training support vector machines using adaptive clustering", In Berry, M.W., Dayal, U., Kamath, C., Skillicorn, D.B., eds.: Proceedings of the Fourth SIAM International Conference on Data Mining, Lake Buena Vista, Florida, USA, pp.126-137, 2004.

[35] Tong, S., Koller, D., "Support vector machine active learning with applications to text classification", In proceedings of the 17th Intl. Conf. on Machine Learning, ACM, pp. 999-1006, 2000.

[36] Pavlov, D., Mao, J., Dom, B., "Scaling-up support vector machines using boosting algorithm", In 15th International Conference on Pattern Recognition, Volume 2, pp.219-222, 2000.

[37] Do, T. N., Le-Thi, H.A., "Classifying large datasets with svm", In proceedings of 4th Intl. Conf. on Computational Management Science, 2007.

[38] Do, T. N., Fekete, J. D., "Large scale classification with support vector machine algorithms. In Wani, M.A., Kantardzic, M. M., Li, T., Liu, Y., Kurgan, L. A., Ye, J., Ogihara, M., Sagiroglu, S., Chen, X.w., Peterson, L.E., Hafeez, K., eds., The Sixth International Conference on Machine Learning and Applications, ICMLA 2007, Cincinnati, Ohio, USA, pp.7-12, 2007.

[39] Freund, Y., Schapire, R., "A short introduction to boosting", *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp.771-780, 1999.

[40] Breiman, L., "Arcing classifiers", *The annals of statistics*, vol. 26, no. 3, pp.801-849, 1998.

[41] Jacobs, R. A., Jordan, M.I., Nowlan, S. J., Hinton, G. E., "Adaptive mixtures of local experts", *Neural Computation* vol. 3, no. 1, pp.79-87, 1991.

[42] 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, no. 1, pp.1-38, 1977.

[43] 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.

[44] 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.

[45] Yang, T., Kecman, V., "Adaptive local hyperplane classification", *Neurocomputing* vol. 71, no. 13-15, pp.3001-3004, 2008.

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

[47] Cheng, H., Tan, P. N., Jin, R., "Efficient algorithm for localized support vector machine", *IEEE Transactions on Knowledge and Data Engineering*, vo. 22, no. 4, pp.537-549, 2010.

[48] Kecman, V., Brooks, J., "Locally linear support vector machines and other local models", In the 2010 International Joint Conference on Neural Networks (IJCNN), pp.1-6, 2010.

[49] Ladicky, L., Torr, P. H. S., "Locally linear support vector machines", In Getoor, L., Scheffer, T., eds.: Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, Omnipress pp. 985-992, 2011.

[50] Gu, Q., Han, J., "Clustered support vector machines", In Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2013, Scottsdale, AZ, USA, Volume 31, pp.307–315, 2013.

NON-LINEAR CLASSIFICATION OF MASSIVE DATASETS WITH A PARALLEL ALGORITHM OF LOCAL SUPPORT VECTOR MACHINES

Do Thanh Nghi, Pham Nguyen Khang

ABSTRACT *- We propose a new parallel algorithm of local support vector machines, called kSVM for the effectively non-linear classification of large datasets. The learning strategy of kSVM uses kmeans algorithm to partition the data into k clus- ters, followed which it constructs a non-linear SVM in each cluster to classify the data locally in the parallel way on multi-core computers. The kSVM algorithm is faster than the standard SVM in the non-linear classification of large datasets while maintaining the classification correstness. The numerical test results on 4 datasets from UCI repository and 3 benchmarks of handwritten letters recognition showed that our proposal is efficient compared to the standard SVM.*