

NHẬN DẠNG KÝ TỰ SỐ VIẾT TAY BẰNG GIẢI THUẬT MÁY HỌC

Đỗ Thanh Nghị¹ và Phạm Nguyên Khang¹

¹ Khoa Công nghệ Thông tin & Truyền thông, Trường Đại học Cần Thơ

Thông tin chung:

Ngày nhận: 17/04/2013

Ngày chấp nhận: 19/08/2013

Title:

Handwritten digit recognition using gist descriptors and random oblique decision trees

Từ khóa:

Nhận dạng ký tự viết tay, Đặc trưng GIST, Cây ngẫu nhiên xiên phân, Phân tích biệt lập tuyến tính

Keywords:

Handwritten Digit Recognition, GIST Descriptor, Random Oblique Decision Trees, Linear Discriminant Analysis

ABSTRACT

Our investigation aims at constructing random oblique decision trees to recognize handwritten digits. At the pre-processing step, we propose to use the GIST descriptors to represent digit images in large number of dimensions datasets. And then we propose a multi-class version of random oblique decision trees based on the linear discriminant analysis that is suited for classifying high dimensional datasets. The experimental results on MNIST dataset show that our proposal has very high accuracy compared to state-of-the-art algorithms.

TÓM TẮT

Trong bài viết này, chúng tôi trình bày giải thuật máy học rừng ngẫu nhiên xiên phân (rODT) cho nhận dạng ký tự số viết tay. Chúng tôi đề xuất sử dụng đặc trưng toàn cục (GIST) cho biểu diễn ảnh ký tự số trong không gian có số chiều lớn. Tiếp theo, chúng tôi đề xuất giải thuật học tự động rừng xiên phân ngẫu nhiên đa lớp, mỗi cây thành viên sử dụng siêu phẳng phân chia dữ liệu hiệu quả tại mỗi nút của cây dựa trên phân tích biệt lập tuyến tính (LDA). Việc xây dựng cây xiên phân ngẫu nhiên vì thế tạo cho giải thuật có khả năng làm việc tốt trên dữ liệu có số chiều lớn sinh ra từ bước tiền xử lý. Kết quả thử nghiệm trên tập dữ liệu thực MNIST cho thấy rằng giải thuật rODT do chúng tôi đề xuất nhận dạng rất chính xác khi so sánh với các phương pháp nhận dạng hiện nay.

1 GIỚI THIỆU

Nhận dạng chữ số viết tay là cần thiết và được ứng dụng rộng rãi trong nhiều lĩnh vực như nhận dạng các chữ số trên chi phiếu ngân hàng, mã số trên bì thư của dịch vụ bưu chính, hay các chữ số trên các biểu mẫu nói chung. Vấn đề nhận dạng chữ viết tay nói chung và nhận dạng chữ số viết tay nói riêng là một thách thức lớn đối với các nhà nghiên cứu. Bài toán lớn luôn đặt ra phía trước vì sự phức tạp của việc nhận dạng chữ viết phụ thuộc nhiều vào phong cách viết và cách thể hiện ngôn ngữ của người viết. Chúng ta không thể luôn luôn viết một ký tự chính xác theo cùng một cách. Do vậy, xây dựng hệ thống nhận dạng chữ viết có thể nhận dạng bất cứ ký tự nào một cách đáng

tin cậy trong tất cả các ứng dụng là điều không dễ dàng.

Hệ thống nhận dạng thường bao gồm hai bước: rút trích đặc trưng từ ảnh và học tự động từ các đặc trưng để có thể nhận dạng ký tự. Hiệu quả của hệ thống nhận dạng phụ thuộc vào các phương pháp sử dụng ở hai giai đoạn trên.

Hầu hết các hệ thống hiện nay (LeCun *et al.*, 1998), (Simard *et al.*, 2003), (Kégl & Busa-Fekete, 2009) đều sử dụng các đặc trưng cơ bản từ ảnh ký tự như đường biên, cạnh, độ dày, giá trị mức xám, haar-like, với các xử lý đặc thù như lấy mẫu, dao động các điểm ảnh, biến đổi ảnh, thêm dữ liệu ảo. Sau đó hệ thống nhận dạng huấn luyện

các mô hình học tự động như k láng giềng (kNN), mạng nơ-ron, máy véc tơ hỗ trợ (SVM), boosting.

Hệ thống chúng tôi đề xuất trong bài viết thực hiện hai bước: sử dụng đặc trưng toàn cục (GIST) cho biểu diễn ảnh ký tự số trong không gian có số chiều lớn (960 đặc trưng, chiều cho mỗi ảnh), huấn luyện rừng xiên phân ngẫu đa lớp dựa trên phân tích biệt lập tuyến tính (LDA), nhận dạng hiệu quả các ký tự số. Kết quả thử nghiệm trên tập dữ liệu thực MNIST (LeCun & Cortes, 1989) cho thấy phương pháp chúng tôi đề xuất, huấn luyện, nhận dạng nhanh và chính xác khi so sánh với các phương pháp hiện có.

Phần tiếp theo của bài viết được trình bày như sau: phần 2 trình bày ngắn gọn về trích đặc trưng GIST từ ảnh, phần 3 trình bày giải thuật ODT của chúng tôi đề xuất. Phần 4 trình bày các kết quả thực nghiệm tiếp theo sau đó là kết luận và hướng phát triển.

2 RÚT TRÍCH ĐẶC TRƯNG

Trong hệ thống nhận dạng, bước rút trích đặc trưng là rất quan trọng, ảnh hưởng lớn đến hiệu quả của việc huấn luyện mô hình học tự động. Các đặc trưng rút trích từ ảnh phải đạt được mục đích quan trọng là dựa trên các đặc trưng đó mà giải thuật học có thể phân biệt tốt nhất một ký tự số này với một ký tự số khác.

Các nghiên cứu tiên phong trong lĩnh vực nhận dạng (LeCun *et al.*, 1998), (Simard *et al.*, 2003), (Kégl & Busa-Fekete, 2009) đều sử dụng các đặc trưng cơ bản từ mức thấp nhất là giá trị mức xám của từng điểm ảnh, đường biên, cạnh, độ dày, tổ chức đồ haar-like, đến các phương pháp xử lý đặc biệt khác như lấy mẫu, dao động các điểm ảnh, biến đổi ảnh.

Trong mấy năm trở lại đây, cộng đồng nghiên cứu về thị giác máy tính và tìm kiếm ảnh đặc biệt quan tâm đến hai kiểu đặc trưng rất hiệu quả là đặc trưng cục bộ không đổi (SIFT) của (Lowe, 2004) và đặc trưng toàn cục GIST của (Oliva & Torralba, 2001). Các véc tơ mô tả SIFT rút trích từ ảnh có tính chất quan trọng là: không bị thay đổi trước những biến đổi tỉ lệ, tịnh tiến, phép quay, không bị thay đổi một phần đối với phép biến đổi hình học affine (thay đổi góc nhìn) và khả năng chịu đựng với những thay đổi về độ sáng, sự che khuất hay nhiễu. Tuy nhiên bất biến với phép quay của đặc trưng SIFT lại gây ra sự

bất lợi cho nhận dạng ký tự số (số 9 và 6 có thể như nhau). Hơn nữa, phương pháp SIFT cung cấp rất nghèo nàn về số lượng các đặc trưng từ ảnh ký tự số (ít hơn 10 đặc trưng). Trong khi đó sử dụng đặc trưng toàn cục GIST thì không gặp khó khăn như SIFT. Chính vì lý do đó, chúng tôi sử dụng đặc trưng toàn cục GIST để giải quyết vấn đề nhận dạng số viết tay. Phương pháp GIST rút trích từ ảnh tập hợp các đặc trưng quan trọng như tính tự nhiên, mở rộng, độ nhám, độ chắc chắn, cho phép trình bày cấu trúc không gian của một cảnh. Để tính toán đặc trưng mô tả GIST, ảnh đầu vào được đưa về dạng vuông, chia thành lưới 4 x 4, các tổ chức đồ theo hướng tương ứng được trích ra. Nguyên lý trích đặc trưng dựa vào phép biến đổi Gabor theo các hướng và tần số khác nhau. Mỗi ảnh ký tự số được rút trích các đặc trưng GIST (véc tơ có 960 chiều). Sau bước trích đặc trưng này, tập dữ liệu ảnh đưa về dạng bảng hay ma trận mà ở đó mỗi ảnh là một dòng có 960 cột (chiều), mỗi ký tự số được gán nhãn (lớp tương ứng là 0, 1, ..., 9).

3 RỪNG NGẪU NHIÊN XIÊN PHÂN CHO PHÂN LỚP ĐA LỚP

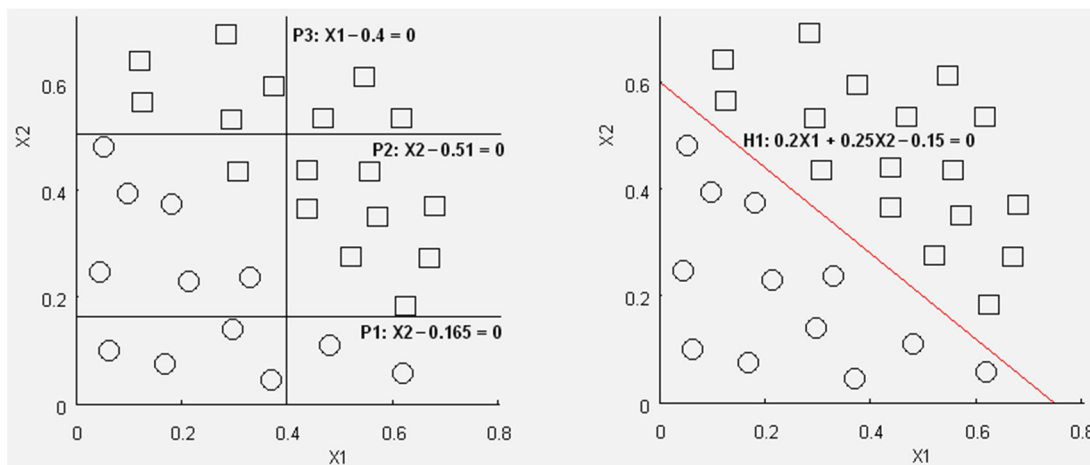
Bước tiền xử lý, rút trích đặc trưng ảnh ký tự số tạo ra tập dữ liệu có số chiều lớn. Giải thuật phân lớp được chọn tiếp theo sau phải có khả năng xử lý tốt dữ liệu có số chiều lớn. Một nghiên cứu trước đây trong (Do *et al.*, 2009), chúng tôi đã đề nghị giải thuật rừng ngẫu nhiên xiên phân RF-ODT cho phân lớp hiệu quả dữ liệu có số chiều lớn. Đây là sự mở rộng từ RF-CART được đề nghị bởi (Breiman, 2001).

Hiệu quả của một giải thuật học như đã nghiên cứu của (Breiman, 1996, 2001) dựa trên cơ sở của 2 thành phần lỗi là bias và variance mà ở đó, thành phần lỗi bias là lỗi của mô hình học so với Bayes và variance là lỗi do tính biến thiên của mô hình so với tính ngẫu nhiên của các mẫu dữ liệu. Trong nghiên cứu kết hợp nhiều mô hình phân loại thành tập hợp các mô hình phân loại để cho tính chính xác cao hơn so với chỉ một mô hình đơn. Giải thuật RF-CART của Breiman xây dựng một tập hợp các cây quyết định hiệu quả cao và đa dạng (có sự tương quan thấp giữa các cây thành viên). Để giữ được bias thấp, RF-CART xây dựng các cây đến độ sâu tối đa không cần cắt nhánh. Để giữ tính tương quan giữa các cây ở mức thấp, RF-CART sử dụng việc lấy mẫu có hoàn lại (bootstrap) từ tập dữ liệu ban đầu để xây dựng cây

thành viên và chọn ngẫu nhiên một tập con các thuộc tính để tính phân hoạch tốt nhất ở các nút trong của cây. RF-CART cho độ chính xác cao so với các giải thuật phân lớp tốt nhất hiện nay bao gồm Boosting (Freund & Schapire, 1995), SVM (Vapnik, 1995). Hơn nữa, nó học nhanh, chịu đựng nhiễu tốt.

Tuy nhiên, việc xây dựng cây của RF-CART chỉ chọn một chiều để phân hoạch dữ liệu tại các nút như đề nghị trước đây (Breiman *et al.*, 1984), (Quinlan, 1993). Do đó, độ chính xác của mô hình

cây bị giảm khi làm việc với các tập dữ liệu có số chiều lớn và phụ thuộc lẫn nhau. Ví dụ như trong Hình 1, bất kỳ việc phân hoạch đơn thuộc tính nào (song song với một trục tọa độ) đều không thể tách dữ liệu một lần duy nhất thành hai lớp một cách hoàn toàn mà phải thực hiện nhiều lần phân hoạch, nhưng việc phân hoạch đa chiều (xiên phân, kết hợp hai thuộc tính) có thể thực hiện một cách hoàn hảo với duy nhất một lần. Vì thế, việc phân hoạch đơn thuộc tính được dùng để xây dựng cây thông thường thì không hiệu quả trong trường hợp này.

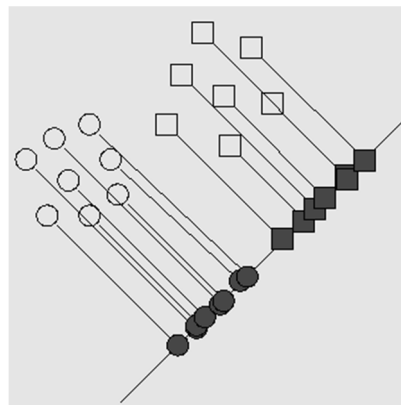


Hình 1: Phân hoạch đơn thuộc tính (trái), phân hoạch đa thuộc tính (phải)

Để khắc phục nhược điểm trên, nhiều giải thuật xây dựng cây quyết định sử dụng phân hoạch đa thuộc tính (xiên phân) tại các nút được đề nghị. (Murthy *et al.*, 1993) đã đưa ra giải thuật OC1, một hệ thống dùng để xây dựng các cây quyết định xiên trong đó dùng thuật toán leo đồi để tìm một phân hoạch xiên tốt dưới dạng một siêu phẳng. Vấn đề xây dựng cây quyết định xiên tối ưu đã được biết như là một vấn đề có độ phức tạp NP-hard. RF-ODT của (Do *et al.*, 2009) xây dựng các cây xiên phân ngẫu nhiên dựa trên siêu phẳng tối ưu (phân hoạch hiệu quả cao, khả năng chịu đựng nhiễu tốt) thu được từ huấn luyện SVM. Tuy nhiên, việc tìm siêu phẳng tối ưu của SVM mặc dù hiệu quả nhưng có độ phức tạp cao.

Để giảm độ phức tạp của cài đặt, chúng tôi đề xuất thay thế SVM bởi phân tích biệt lập tuyến tính LDA, và mở rộng cho vấn đề phân lớp đa lớp (lớn hơn 2). Xét vấn đề phân lớp nhị phân (2 lớp), ý tưởng chính của LDA (Fisher, 1936) là tìm siêu phẳng sao cho khi chiếu dữ liệu lên đó thì độ biệt

lập giữa trung bình dữ liệu của 2 lớp là lớn nhất và độ chồng lấp giữa 2 lớp là nhỏ nhất.



Hình 2: Minh họa véc tơ (w) dùng để chiếu dữ liệu 2 chiều

Một cách ngắn gọn, xét một ví dụ phân lớp nhị phân tuyến tính (hình tròn, vuông) như trong Hình 2, với m điểm dữ liệu x_i ($i=1, m$) trong không gian n chiều. Tập dữ liệu phân làm 2 lớp R_l (có N_l

phần tử), và R_2 (có N_2 phần tử). Để tìm véc tơ chiều tối ưu (w) ta cần tính như sau:

Trung bình (trọng tâm) mỗi lớp:

$$m_1 = \frac{1}{N_1} \sum_{x_i \in R_1} x_i, \quad m_2 = \frac{1}{N_2} \sum_{x_i \in R_2} x_i \quad (1)$$

Chiều trọng tâm của 2 lớp m_1, m_2 lên véc tơ w :

$$\begin{aligned} \tilde{m}_1 &= \frac{1}{N_1} \sum_{x_i \in R_1} w^T x_i = w^T m_1 \\ \tilde{m}_2 &= \frac{1}{N_2} \sum_{x_i \in R_2} w^T x_i = w^T m_2 \end{aligned} \quad (2)$$

Khoảng cách giữa m_1 và m_2 sau khi chiếu lên w (độ biệt lập tuyến tính):

$$|\tilde{m}_2 - \tilde{m}_1| = |w^T (m_2 - m_1)| \quad (3)$$

Mật độ phân bố (scatter) của dữ liệu 2 lớp sau khi chiếu:

$$\tilde{s}_1^2 = \sum_{y_i: x_i \in R_1} (y_i - \tilde{m}_1)^2 = \sum_{x_i \in R_1} (w^T x_i - w^T m_1)^2 = w^T S_1 w \quad (4)$$

$$s_2^2 = \sum_{y_i: x_i \in R_2} (y_i - \tilde{m}_2)^2 = \sum_{x_i \in R_2} (w^T x_i - w^T m_2)^2 = w^T S_2 w \quad \text{Với}$$

S_1, S_2 là:

$$\begin{aligned} S_1 &= \sum_{x_i \in R_1} (x_i - m_1)(x_i - m_1)^T \\ S_2 &= \sum_{x_i \in R_2} (x_i - m_2)(x_i - m_2)^T \end{aligned} \quad (5)$$

Với S_w là ma trận tán xạ bên trong mỗi lớp và S_B là ma trận tán xạ giữa 2 lớp:

$$S_w = S_1 + S_2 \quad (6)$$

$$S_B = (m_2 - m_1)(m_2 - m_1)^T \quad (7)$$

Thi tỉ số giữa độ biệt lập tuyến tính và tổng mật độ phân bố:

$$f(w) = \frac{(\tilde{m}_2 - \tilde{m}_1)^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{w^T S_B w}{w^T S_w w} \quad (8)$$

Mục tiêu của LDA là tìm w sao cho cực đại hoá $f(w)$, đưa đến việc giải bài toán tìm giá trị riêng suy rộng trong (9):

$$S_w^{-1} S_B w = \lambda w \quad (9)$$

LDA tìm siêu phẳng w tối ưu để phân hoạch xiên phân bằng cách thực hiện các đẳng thức tuyến tính trên. Chú ý rằng, siêu phẳng được cây quyết định xiên phân thực hiện rất nhiều lần cho đến nút lá chứ không phải chỉ thực hiện duy nhất một lần phân hoạch. Chính vì lý do đó, khi độ biệt lập tuyến tính của dữ liệu không dựa vào hai trọng tâm m_1 và m_2 (trường hợp dữ liệu phi tuyến), cây xiên phân vẫn xử lý được tình huống này.

Từ khi bắt đầu trình bày giải thuật đến giờ, chúng tôi chỉ tập trung vào vấn đề phân lớp nhị phân (2 lớp). Để mở rộng giải thuật cho vấn đề phân lớp đa lớp (có hơn 2 lớp). Vấn đề chính là chúng ta phải đưa bài toán về dạng 2 lớp để có thể thực hiện lại LDA như mô tả bên trên. Để làm được điều này, chúng tôi đề xuất mô hình phân cấp. Giả sử tại một nút của cây xiên phân, chúng ta có c lớp ($c > 2$). Chúng tôi đề xuất tạo ra 2 lớp (lớp dương và lớp âm), mà ở đó mỗi lớp có chứa dữ liệu từ các lớp khác. Tức là những dữ liệu của các lớp gần nhau được gom vào một trong 2 lớp dương, âm. Đến đây, dữ liệu tại một nút đã trở lại bài toán phân lớp nhị phân, chúng ta có thể áp dụng công thức LDA ở trên. Quá trình cứ thế tiếp tục đến khi phân hoạch hoàn toàn dữ liệu.

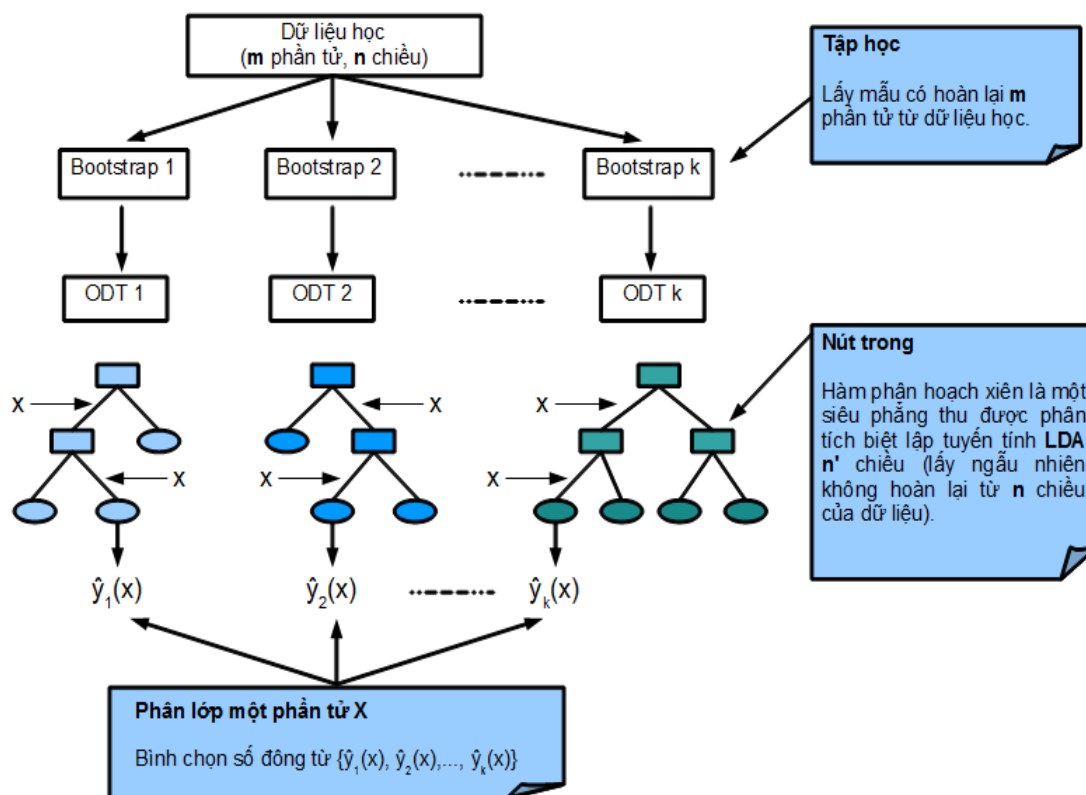
Giải thuật rừng ngẫu nhiên xiên phân ($rODT$) của chúng tôi đề xuất cho một vấn đề phân lớp tập dữ liệu m điểm dữ liệu x_i ($i=1, m$) trong không gian n chiều, được thực hiện như mô tả trong Hình 3. Một cây quyết định xiên (ký hiệu là ODT_k) trong rừng ngẫu nhiên gồm k cây được xây dựng như sau:

- Tập dữ liệu học B_k là m phần tử dữ liệu được lấy mẫu có hoàn lại từ tập dữ liệu ban đầu.

- Tại mỗi nút của cây, chọn ngẫu nhiên n' chiều ($n' < n$) và tính toán phân hoạch xiên phân (sử dụng LDA như mô tả ở trên) dựa trên n' chiều này.

- Cây được xây dựng đến độ sâu tối đa không cắt nhánh.

Rừng ngẫu nhiên xiên $rODT$ phân lớp phần tử x đưa vào bình chọn số đồng từ các phân lớp thu được của các cây thành viên.



Hình 3: Giải thuật rừng ngẫu nhiên xiên phân (rODT)

4 KẾT QUẢ THỰC NGHIỆM

Trong phần thực nghiệm, chúng tôi sử dụng tập dữ liệu MNIST (LeCun & Cortes, 1989) cung cấp, thường dùng để đánh giá hiệu quả của giải thuật nhận dạng ký tự số viết tay. Tập dữ liệu MNIST có nguồn gốc từ tập NIST do tổ chức National Institute of Standards and Technology (NIST) cung cấp, sau đó được LeCun cập nhật và chia thành 2 tập riêng biệt :

- Tập học (huấn luyện) gồm có 60.000 ảnh kích thước 28 x 28, của chữ số viết tay được dùng việc huấn luyện mô hình máy học tự động. Tất cả các ảnh trong tập học đều được canh chỉnh và biến đổi thành dữ liệu dạng điểm gồm 60.000 phần tử (ký tự số) có 784 chiều là giá trị mức xám của các điểm, 10 lớp (từ 0 đến 9).

- Tập kiểm tra gồm có 10.000 ảnh của kí số viết tay được dùng cho việc kiểm thử, tương tự các ảnh trong tập thử cũng được biến đổi và canh

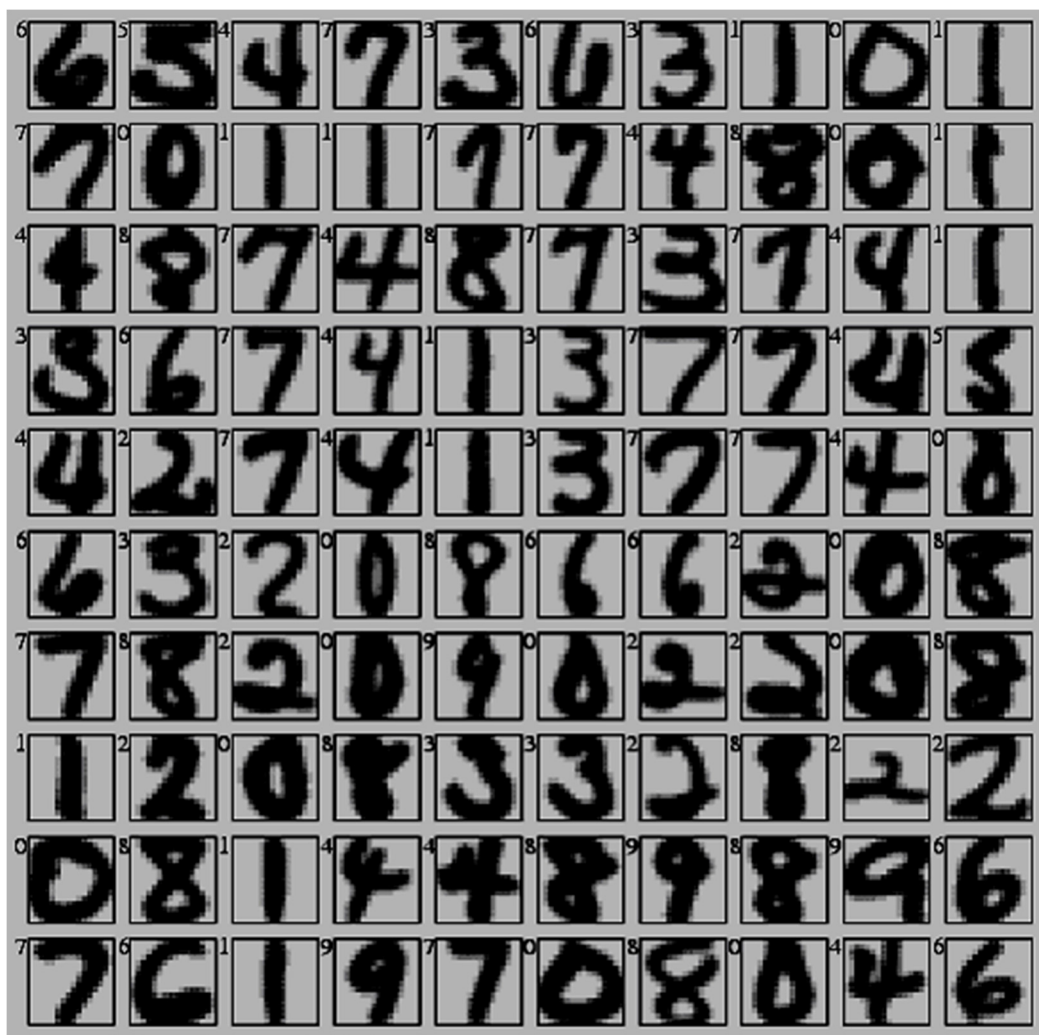
chỉnh thành dữ liệu điểm gồm 10000 phần tử trong 784 chiều, 10 lớp (từ 0 đến 9).

Để có thể đánh giá hiệu quả của phương pháp đề xuất (rODT, GIST), chúng tôi đã sử dụng chương trình của (Douze *et al.*, 2009) để rút trích đặc trưng và đồng thời chúng tôi cũng cài đặt giải thuật rODT bằng ngôn ngữ lập trình C/C++.

Chúng tôi thực hiện so sánh hiệu quả của (rODT và GIST) với các giải thuật hiện nay như giải thuật AdaBoost.M1 (Freund & Schapire, 1995), (Witten & Frank, 05), LibSVM (Chang & Lin, 2001), (Vapnik, 1995), mạng nơron tích chập CNN (Simard *et al.*, 2003), (O'Neill, 2006). Tất cả các kết quả đều được thực hiện trên một máy tính cá nhân (Intel 3GHz, 2GB RAM) chạy hệ điều hành Linux. Kết quả thu được độ chính xác như trình bày trong Bảng 1. Các kết quả tham khảo từ các phương pháp của (LeCun *et al.*, 1998), (Kégl & Busa-Fekete, 2009) cũng được trình bày trong bảng.

Bảng 1: Kết quả nhận dạng tập ký tự viết tay MNIST

STT	Phương pháp	Độ chính xác (%)
1	1-layer Neural nets (LeCun <i>et al.</i> , 1998)	88.00
2	Nearest-neighbor (Euclidean L2) (LeCun <i>et al.</i> , 1998)	95.00
3	Convolution net LeNet-1 (LeCun <i>et al.</i> , 1998)	98.30
4	Convolution net LeNet-4* (LeCun <i>et al.</i> , 1998)	98.90
5	Convolution net LeNet-5* (LeCun <i>et al.</i> , 1998)	99.15
6	Convolution Neural Net (CNN)* (Simard <i>et al.</i> , 03)	99.10
7	LIBSVM (RBF, $\gamma = 0, 05$, $c = 10^5$)	98.37
8	LIBSVM (Poly, $\deg = 5$, $c = 10^5$)	96.65
9	AdaBoost.M1 (100 trees with C4.5)	95.95
10	Products of boosted stumps (haar)* (Kégl & Busa-Fekete, 2009)	99.12
11	rODT (100 oblique decision trees, GIST)	99.12

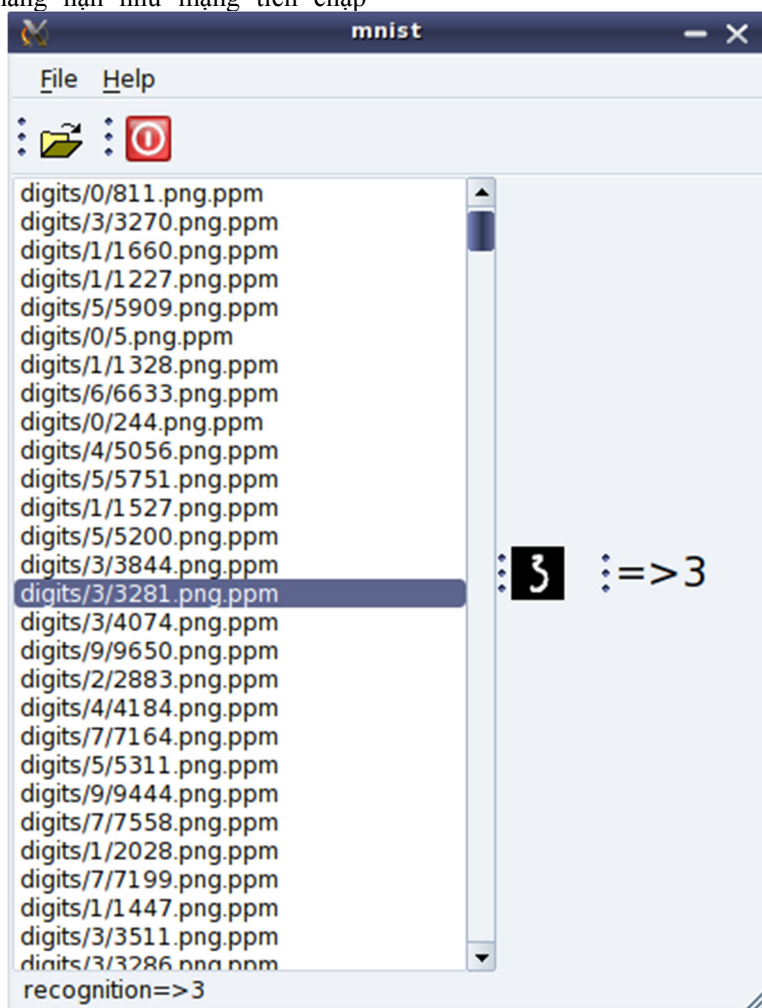


Hình 4: Mẫu ký tự số của MNIST

Kết quả thực nghiệm cho thấy rằng rODT sử dụng GIST cho kết quả nhận dạng chính xác đến 99.12%, là một trong ba phương pháp nhận dạng tốt nhất khi so sánh với tất cả các phương pháp khác.

Đi vào chi tiết kết quả thu được của từng phương pháp, những phương pháp có đánh dấu (*) cho biết tác giả đã sử dụng các xử lý đặc thù để thu được kết quả nhận dạng tốt. Thường các xử lý này khá phức tạp như trích đặc trưng haar-like, làm biến dạng, thay đổi dữ liệu và cả cài đặt giải thuật rất phức tạp như mạng tích chập CNN, tích các boosting. Chẳng hạn như mạng tích chập

CNN của (Simard *et al.*, 2003), (O'Neill, 2006) và các giải thuật AdaBoost.M1 (Freund & Schapire, 1995), (Witten & Frank, 2005), thời gian huấn luyện mạng gần 1 ngày để đạt được độ chính xác như trong bảng. Trong khi các giải thuật SVM không cần bất kỳ một xử lý đặc biệt nào, chỉ mất thời gian huấn luyện 30 phút (nhanh hơn 50 lần), cho độ chính xác gần tương đương (thấp hơn 0.5%). Giải thuật rODT cần thời gian huấn luyện khoảng 15 phút (nhanh hơn 100 lần so với mạng tích chập) vẫn cho độ chính xác trong top 3. Quan trọng là rODT rất nhanh, đơn giản, dễ cài đặt và tích hợp vào chương trình.



Hình 5: Chương trình nhận dạng ký tự số (rODT, GIST)

5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chúng tôi vừa trình bày giải thuật máy học rừng ngẫu nhiên xiên phân (rODT) sử dụng các đặc trưng toàn cục (GIST), cho phép nhận dạng

chính xác ký tự số viết tay. Bước tiền xử lý trích đặc trưng toàn cục từ ảnh ký tự số cho ra bảng dữ liệu có số chiều lớn. Chúng tôi đề xuất giải thuật học tự động rừng xiên phân ngẫu nhiên đa lớp, mỗi cây thành viên sử dụng siêu phẳng phân chia dữ liệu

hiệu quả tại mỗi nút của cây dựa trên phân tích biệt lập tuyến tính (LDA). Kết quả thử nghiệm trên tập dữ liệu thực MNIST cho thấy rằng giải thuật rODT do chúng tôi đề xuất nhận dạng rất chính xác khi so sánh với các phương pháp nhận dạng hiện nay. Phương pháp đề xuất đạt hiệu quả nhận dạng chính xác cao nhưng không cần bất cứ xử lý đặc biệt nào. Các thử nghiệm cho nhận dạng ký tự viết tay gồm ký tự số và 26 ký tự alphabet cho thấy phương pháp của chúng tôi thật sự tốt. Trong tương lai gần, chúng tôi kết hợp hệ thống này với các phương pháp khác cho phép trích, đọc số xe.

Hướng tiếp cận có thể áp dụng vào các vấn đề tương tự trong lĩnh vực nhận dạng, phân lớp, tìm kiếm ảnh.

TÀI LIỆU THAM KHẢO

1. L. Breiman, J.H. Friedman, R.A. Olshen and C. Stone. *Classification and Regression Trees*. Wadsworth International, 1984.
2. L. Breiman. Bagging predictors. *Machine Learning* 24(2):123–140, 1996.
3. L. Breiman. Random forests. *Machine Learning* 45(1):5–32, 2001.
4. C.C. Chang and C.J. Lin. Libsvm – a library for support vector machines. 2001. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
5. T.N. Do, S. Lallich, N.K. Pham and P. Lenca. Classifying very-high-dimensional data with random forests of oblique decision trees. in *Advances in Knowledge Discovery and Management Vol. 292*, Springer-Verlag, 2009, pp. 39-55.
6. M. Douze, M., H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid. Evaluation of GIST descriptors for web-scale image search. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, 2009, pp. 1–8.
7. Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Computational Learning Theory*, 1995, pp. 23–37.
8. B. Kégl and R. Busa-Fekete. Boosting products of base classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 497–504.
9. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 1998, pp. 2278–2324.
10. LeCun, Y. and C. Cortes. The MNIST database of handwritten digits. 1989.
11. D. Lowe. Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision*, 2004, pp. 91–110.
12. S. Murthy, S. Kasif, S. Salzberg and R. Beigel. Oc1: Randomized induction of oblique decision trees. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 1993, pp. 322–327.
13. A. Oliva and A. Torralba. Modeling the shape of the scene : A holistic representation of the spatial envelope. *International Journal of Computer Vision* 42, 145–175, 2001.
14. J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
15. Y. Simard, D. Steinkraus, J. Platt. Best Practices for Convolutional Neural Network Applied to Visual Document Analysis. in *Intl Conference on Document Analysis and Recognition*, 2003, pp. 958-962.
16. V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.
17. H. Witten and E. Frank. *Data Mining : Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.