**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙠🕮🙢**

**Đồ án 9**

**Local Adaptive Support Vector Machine**

**for Objective Recognition**

**Nayyar A. Zaidi, David McG. Squire**

GVHD: TS. Trần Thái Sơn

Nhóm 9:

Ngành Khoa Học Máy Tính – Cao học khóa 23

1. Đỗ Đặng Minh - 1311015
2. Huỳnh Công Toàn - 1311026
3. Hồ Văn Tấn - 1311058

*Tp. Hồ Chí Minh, tháng 10 năm 2014*

# Thông tin nhóm

|  |  |  |  |
| --- | --- | --- | --- |
| **MSHV** | **Họ tên** | **Email** | **Điện thoại** |
| 1311015 | Đỗ Đặng Minh (nhóm trưởng) | [masterminh219@gmail.com](mailto:masterminh219@gmail.com) | 0168-993-5242 |
| 1311026 | Huỳnh Công Toàn | [alex7huynh@gmail.com](mailto:alex7huynh@gmail.com) | 0121-516-1090 |
| 1311058 | Hồ Văn Tấn | [tanhv90@gmail.com](mailto:tanhv90@gmail.com) | 090-290-9334 |

# Phân công công việc

|  |  |
| --- | --- |
| Nội dung tìm hiểu | |
| SVM | Toàn |
| Local SVM | Toàn |
| Local Adaptive SVM | Minh |
| Kết quả thực nghiệm | Tấn |
| Báo cáo & slide | |
| Viết báo cáo | Minh, Tấn |
| Viết slide | Toàn |

# Danh mục các kí hiệu, chữ viết tắt và ý nghĩa

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Nghĩa tiếng Anh** | **Nghĩa tiếng Việt** |
| KNN | K-Nearest Neighbor | K láng giềng gần nhất |
| LASVM | Local Adaptive Support Vector Machine | SVM thích ứng cục bộ |
| LSVM | Local Support Vector Machine | SVM cục bộ |
| MEGM | Mean-Square-Error Gradient Minimization | Tối thiểu gradient độ lỗi bình phương trung bình |
| MSE | Mean-Square-Error | Độ lỗi bình phương trung bình |
| SVM | Support Vector Machine | Máy vector hỗ trợ |

# Danh mục các hình

[**Hình 1:** So sánh tý lệ chính xác của phương pháp LSVM, LASVM với các phương pháp KNN, SVM, MEGM trên dữ liệu khuôn mặt, USPS, Isolet và Coil100 9](#_Toc401292915)

[**Hình 2:** So sánh tỷ lệ chính xác của LSVM với LSVM cho dữ liệu khuôn mặt, USPS, Isolet và Coil100 khi thay đổi kích thước vùng lân cận 9](#_Toc401292916)

[**Hình 3:** So sánh tý lệ chính xác của phương pháp LSVM, LASVM với các phương pháp KNN, SVM, OSVM, MEGM trên dữ liệu UCIML 10](#_Toc401292917)

[**Hình 4:** So sánh tỷ lệ chính xác của LSVM với LSVM cho dữ liệu UCIML khi thay đổi kích thước vùng lân cận 10](#_Toc401292918)

Nội dung

[Thông tin nhóm 1](#_Toc401299322)

[Phân công công việc 1](#_Toc401299323)

[Danh mục các kí hiệu, chữ viết tắt và ý nghĩa 1](#_Toc401299324)

[Danh mục các hình 1](#_Toc401299325)

[I. Tổng quan 3](#_Toc401299326)

[II. Giới thiệu về Local SVM 3](#_Toc401299327)

[III. Mục tiêu 6](#_Toc401299328)

[IV. Kết quả 7](#_Toc401299329)

[A. Dữ liệu khuôn mặt, USPS, Isolet và Coil 8](#_Toc401299330)

[B. Kho dữ liệu UCI-ML 9](#_Toc401299331)

[V. Chương trình minh họa 11](#_Toc401299332)

[VI. Tổng kết 11](#_Toc401299333)

[VII. Tài liệu tham khảo 11](#_Toc401299334)

# Tổng quan

Support Vector Machine (SVM) là một công cụ phân lớp hiệu quả. Tuy nhiên, SVM không phải là công cụ vạn năng. Cả việc huấn luyện (trainning) và kiểm thử (testing) đòi hỏi chi phí tính toán cao. Bên cạnh đó, việc điều chỉnh các thông số của hàm kernel là một quá trình phức tạp. Ngược lại, thuật toán người láng giềng gần nhất (KNN) thì lại tính toán phân lớp hiệu quả. Như vậy, để có thể đạt được việc phân lớp hiệu quả của SVM và chi phí tính toán tối ưu của KNN, như đã trình bày phía trên, thay vì huấn luyện một SVM đơn toàn cục, các SVM đã được phân chia sẽ được huấn luyện cho người láng giềng ở các điểm truy vấn.

Trong bài báo này, nhóm tác giả tiến hành mở rộng thuật toán SVM cục bộ (Local SVM – LSVM). Trong đó, thuật toán SVM thích ứng cục bộ (LASVM) sẽ được huấn luyện cho các SVM cục bộ ở các không gian láng giềng đã được tinh chỉnh của các điểm truy vấn. Mục tiêu chính của nghiên cứu này nhằm hai mục đích: đầu tiên, là trình bày lý thuyết thuật giải LASVM để huấn luyện cho các SVM cục bộ - LSVM. Sau đó, nghiên cứu sẽ đi sâu hơn vào chi tiết động lực nghiên cứu của LSVM và LASVM và những ảnh hưởng của nó trong việc điều chỉnh các tham số của hàm kernel của SVM. Bên cạnh đó, nghiên cứu cũng chỉ ra rằng việc huấn luyện SVM trong các cục bộ địa phương tưng ứng có thể mang lại hiệu quả trong việc tăng hiệu năng của phân lớp một cách đáng kể. Các thực nghiệm đã được đúc kết từ việc lựa chọn các cơ sở dữ liệu – của UCIML, khuôn mặt, vật và các chữ số.

# Giới thiệu về Local SVM

Các hàm phân lớp SVM, dựa trên tiêu chí là giảm thiểu các sai sót có cấu trúc, là một công cụ phổ biến cho việc phân lớp. Các hàm này đã cho thấy hiệu quả trên các tập dữ liệu phân lớp ở phạm vi rộng. Thực tế, một SVM sẽ tìm ra một siêu phẳng tối ưu để chia dữ liệu thành hai lớp. Cho trước tập dữ liệu huấn luyện , hàm quyết định được xác định bằng cách giải quyết vấn đề tối ưu hóa lồi như sau:

sao cho và .

Đồng thời, hàm hạt nhân được tính như sau: , (1)

và là nhân tử Larrange, là tham số hiệu chỉnh độ lỗi để loại bỏ những phân lớp sai và là hàm hạt nhân (kernel).

Mặc dù phổ biến và hiệu quả, SVM vẫn tồn tại ít nhất là hai vấn đề cần phải được giải quyết khi huấn luyện. Trước hết, SVM được sử dụng để giải quyết các bài toán phân lớp nhị phân. Đối với việc phân nhiều lớp, mỗi lớp phải sử dụng tới chiến thuật one-vesus-all hoặc all-vesus-one. Điều này cũng có nghĩa là hệ thống này sẽ ngày càng phức tạp và phức tạp hơn nữa khi mà số lượng lớp tăng lên. Hơn nữa, mỗi khi một lớp mới được thêm vào, tất cả các phân lớp đều phải được huấn luyện lại. Điều này làm tăng thêm quá trình huấn luyện và kiểm thử. Ngoài ra, vấn đề còn nằm ở việc điều chỉnh các tham số của SVM, chẳng hạn như và tham số tỷ lệ độ dài của hàm hạt nhân trong phương trình trên. Những tham số này thường được điều chỉnh thông qua biểu đồ kiểm thử chéo vốn có mức độ tính toán chi phí cao. Hơn thế nữa, để đơn giản tuyệt đối, hàm hạt nhân ở phương trình trên cần phải được xem xét dưới dạng đẳng hướng, có nghĩa là đối với tất cả các chiều d, bởi vì điều chỉnh các tham số đa quy mô thì không khả thi với quá trình kiểm thử chéo. Gần đây, tuy nhiên, Chapelle và các đồng nghiệp [2] đã cho thấy việc tinh chỉnh các tham số tỷ lệ cho hàm hạt nhân thì sẽ cải thiện hiệu năng phân lớp của SVM.

Các phương pháp cục bộ, chẳng hạn như hồi quy logic cục bộ (local logistic regression) và K-Nearest Neighbor (KNN), đã đưa ra một chiến lược để huấn luyện các mẫu toàn cục phức tạp. KNN đã chứng tỏ trong việc thể hiện hiệu quả cho các vấn đề phân lớp đã đề cập trong [3], [4]. KNN không chỉ phân lớp tính toán hiệu quả, mà nó còn cho thấy, với các hàm tính khoảng cách đúng đắn, nó có thể vượt xa hơn cả những phương pháp phức tạp khác [5] - [7]. Sức mạnh của thuật toán KNN xuất phát từ sự giản đơn của nó. Không giống như SVM, các hàm phân lớp của KNN giải quyết với các vấn đề về nhiều lớp – đa lớp một cách dễ dàng. Những láng giềng Κ của mỗi điểm truy vấn ứng với nhãn của lớp , sẽ được tính bằng công thức . Quá trình phân rã sẽ suôn sẻ cũng như là khoảng cách của láng giềng tới các điểm truy vấn cũng sẽ tăng (được điều chỉnh bởi hàm hạt nhân ).

Tương ứng với SVM, các tham số có thể điều chỉnh được của KNN, chẳng hạn như kích thước của các láng giềng và hàm hạt nhân của nó , thì không dễ tìm ra và là đối tượng cho rất nhiều nghiên cứu sau này. Sự thành công của phương thức KNN, tuy nhiên, đã đưa ra hai bài học có giá trị là: đầu tiên, kết quả nhận diện tốt có thể thu được bằng việc định nghĩa sự đo lường tương đương một cách đúng đắn với các nguyên mẫu. Các nguyên mẫu này được chứng thực thông qua các quá trình nghiên cứu nhận thức của con người [8]. Ngoài ra, phần lớn các thông tin yêu cầu để đưa ra quyết định về các nhãn của một truy vấn thì được hiện trong chính các láng giềng cục bộ của chính nó.

Xây dựng những điều trên trong các hàm cục bộ, và dựa trên các động lực đã được trình bày, trong bài nghiên cứu này sẽ đề xuất sử dụng SVM cục bộ. Từ đó, thay vì phải huấn luyện một phân lớp SVM toàn cục, chúng ta có thể huấn luyện một phân lớp SVM chỉ trong một láng giềng của một điểm truy vấn. LSVM, theo những cách nào đó, đã làm giảm bớt các vấn đề đã được đề cập ở trên với thuật toán SVM nguyên thủy. Một vài động lực phía sau các phương thức SVM cục bộ như sau:

* Nguyên nhân chính để sử dụng LSVM như đã được đề cập là lợi ích to lớn trong quá trình phân lớp. Như chúng ta sẽ thấy trong chương IV, kết quả đạt được của LSVM là rất tốt với một vài cơ sở dữ liệu, tuy nhiên lại rất thất trong hầu hết các trường hợp. Tuy nhiên, LSVM có một lợi thế so với các SVM tiêu chuẩn bởi không cần phải huấn luyện lại và khả năng mở rộng số lượng các lớp khả thi. Thời gian để kiểm thử cũng tăng lên, vì các phân lớp SVM phải được huấn luyện cho từng điểm. Mặc dù vậy, với mọi cơ sở dữ liệu có *Μ* lớp, và mỗi điểm được bao quanh bởi tối đa là *m* lớp, và thông thường là . Điều này có thể làm đẩy nhanh quá trình nhận dạng hơn là làm chậm nó.
* Ngoài ra, một nguyên nhân khác để thúc đẩy SVM cục bộ là các tham số tỷ lệ của hàm hạt nhân. Điều đó cho thấy, ít nhất là cục bộ, xác suất có điều kiện của các lớp tương đương sẽ đi qua tất cả các đặc trưng và sử dụng hàm hạt nhân đẳng hướng mà không tốn quá nhiều chi phí. Điều này sẽ tránh cho việc điều chỉnh các tham số của hàm hạt nhân trong SVM.

Thông thường trong lĩnh vực nghiên cứu thị giác máy tính như trong [10], [11], để tránh quá trình kiểm thử chéo gây chi phí cao, giá trị của tham số tỷ lệ () của hàm hạt nhân sẽ được gán là giá trị trung bình của khoảng cách bình phương của tất cả các điểm dữ liệu. Điều đó có nghĩa là : .Cách tính này không chỉ đảm bảo rằng giá trị khoảng của hàm hạt nhân sẽ nằm trong các các khoảng giới hạn, mà nó còn đảm bảo kết quả thu được sẽ là tối ưu giữa đường nghiêng và phương sai của các hàm phân lớp. Tuy nhiên, việc sử dụng như một giá trị trung bình của khoảng cách giữa các dữ liệu kiểm thử thì không tối ưu. Nhưng vì LSVM chỉ được huấn luyện cục bộ, việc sử dụng như một giá trị trung bình của khoảng cách giữa các điểm trong láng giềng thì lại tối ưu cục bộ. Điều đó cho thấy tại sao LSVM có thể cho kết quả phân lớp tốt hơn trong một vài trường hợp so với SVM nguyên thủy. Điều đó cũng chỉ ra rằng, hiệu nặng của LSVM sẽ được phụ thuộc vào kích thước của láng giềng mà nó huấn luyện. Điều này dẫn tới hiệu năng của LSVM sẽ tốt hơn với láng giềng có kích thước lớn hơn.

Truy nhiên, việc gia tăng kích thước của láng giềng sẽ dẫn tới việc tăng chi phí tính toán. Do đó, kích thước của láng giềng cần phải được điều chỉnh sao cho nó tối ưu với kết quả đạt được giữa quá trình tính toán và hiệu quả của quá trình phân lớp. Một cách khác để tăng hiệu quả của quá trình phân lớp mà vẫn giữ nguyên kích thức của láng giềng là là sẽ huấn luyện SVM cục bộ trong một không gian láng giềng thích ứng của các điểm truy vấn. Các công thức đó sẽ gọi là local adaptive support vector machine (LASVM). Một ưu điểm khác của LASVM là sự giảm đi các tác động của Curse-Of-Dimension (COD) trên SVM. Điều này đã được chứng minh rằng các phân lớp SVM đều không tránh khỏi COD. LASVM sử dụng chiến thuật inh chỉnh ma trận khoảng cách và do đó, các hạt nhân sẽ cung cấp trọng lượng tương ứng với mỗi đặc trưng. Những kết quả trong một số chiều đã hoàn toàn được loại bỏ (từ quá trình trích chọn đặc trưng), do đó làm giảm số chiều. Bên cạnh đó, LASVM cũng cung cấp một nền tảng cho việc tích hợp các hàm hạt nhân không cố định trong SVM.

Ý tưởng về SVM cục bộ đã đực tìm thấy trong chi tiết ở [13] và [9]. Trong [13], các tác giả đã đề xuất ra SVM cục bộ gọi là “Profile SVM” (PSVM). Dữ liệu ban đầu sẽ được gom cụm và các SVM phân tách sẽ được huấn luyện cho mỗi cụm. Mỗi điểm truy vấn sẽ được gán vào cụm gần nhất với và sẽ sử dụng SVM cục bộ tương ứng để gán nhãn. Mặc dù thú vị, điểm hổng to lớn của thuật toán này là xác định số lượng các cụm trước khi huấn luyện. Hơn thế nữa, các kết quả có thể phụ thuộc vào sự sắp đặt các cụm và số lượng của các cụm. Mặc dù các kết quả đươc báo cáo dựa trên cơ sở dữ liệu UCIML khi mà PSVM làm tốt hơn so với các SVM nguyên thủy, hiệu năng của PSVM trên các cơ sở dữ liệu nhận dạng đối tượng thì lại không được đề cập. Như đã trình bày, những nghiên cứu của chúng ta sẽ dựa trên và gần hơn với các kết quả nghiên cứu trong [9].

Cũng như [9], chúng ta cũng tiến hành huấn luyện các SVM phân chia cho mỗi truy vấn. Đồng thời, kích thước của láng giềng là không đổi trong suốt các cơ sở dữ liệu để mà hiệu quả của việc học LSVM có hệ thống, khi mà trong [9], kích thước này được xác định thông qua các quá trình kiểm thử chéo của mỗi tập dữ liệu. Trong chương IV, chúng ta sẽ bàn về chi tiết sau giải thuật này. Bên cạnh đó, cần phải nghiên cứu về những ảnh hưởng của sự biến đổi của các láng giềng trong hiệu năng mà không được đề cập trong [9]. Điều này sẽ cải thiện các kết quả nghiên cứu liên quan tới LSVM. Ngoài ra, không như [9], chúng ta sẽ đề xuất thêm Local Adaptive SVM (LASVM) mà đã được truyền cảm hứng từ những sự thành công gần đây trong phép học ma trận cho phương pháp KNN. Nó cho thấy việc các ma trận kết quả trong phép biến đổi tuyến tính của dữ liệu. LASVM sẽ huấn luyện SVM trong một không gian cục bộ được biến đổi bởi ma trận họ, Mặc dù vậy, SVM cục bộ có thể được tìm hiểu như đã đề cập, có thể trở thành tốt nhất trong kiến thức của chúng ta về LASVM.

# Mục tiêu

Ứng với mỗi điểm truy vấn , các SVM cục bộ sẽ hoạt động như sau:

* Tìm ra K láng giềng của . Nếu số tập dữ liêu huấn luyện, , thông thường .
* Nếu nhãn của toàn bộ điểm trong K là như nhau, thì điểm này sẽ thuộc và lớp tương ứng và kết thúc.
* Nếu tồn tại các nhãn khác nhau, SVM one-versus-all sẽ được huấn luyện cho từng lớp hiện diện, và khi đó sẽ được gán nhãn tương ứng.

Như đã trình bày phía trên, LASVM mở rộng từ LSVM bằng việc huấn luyện SVM cục bộ trong một không gian biến đổi., khi mà quá trình biến đổi dựa trên ma trận A được học từ việc sử dụng thuật giải học số liệu (metric learning). Trong nghiên cứu này, chúng ta sẽ xem xét việc sử dụng thuật giải MEGM vào việc học cho phép biến đổi ma trận A. Các thuật giải số liệu học đều nhằm mục tiêu tìm ra phép biến đổi tuyến tính của những dữ liệu, sang phép biến đổi không gian, khi mà phân lớp KNN hoạt động tốt hơn. Nếu chúng ta bao hàm một phép biến đổi ma trận A, việc học một số liệu hiệu quả sẽ cảm sinh bởi  như là . Mean-Square-Error(MSE) Gradient Minimization (MEGM) là một thuật giải học số liệu đơn giản dựa trên gradient mà nó đã cho thấy hiệu quả tốt trên phần lớn các cơ sở dữ liệu nhận dạng đối tượng [14]. MEGM thì dựa trên sự tối tiểu của một hàm mục tiêu MSE sử dụng thuật giải gradient descent ở trong KNN.

Một dàn các bước về thuật giải LASVM sẽ được trình bày dưới đây. Thông qua những công thức của LASVM tích hợp MEGM, những thuật giải học số liệu khác, chẳng hạn như NCA [15], LMNN [16] có thể được sử dụng. Có rất nhiều việc cần phải làm trong các thuật giải số liệu học cục bộ thích ứng. [7]. Một SVM cục bộ có thể được huấn luyện dựa trên láng giềng đã được biến đổi từ các số liệu cục bộ thích ứng. Chúng ta đã sử dụng MEGM, chúng ta có kết quả tốt hơn với MEGM so với NCA [15], LMNN[16] và DANN [7]. Việc so sánh hiệu năng của LASVM với các thuật giải số liệu học khác và với nhiều thuật giải số liệu học thích ứng còn để trong tương lai.

**Thuật giải LASVM:** huấn luyện phân lớp SVM trong láng giềng tương ứng với điêm truy vấn.

**Yêu cầu:**

* Dữ liệu kiểm thử: .
* Dữ liệu huấn luyện: khi mà  là một vector đặc trưng của chiều , là số lượng dữ liệu huấn luyện, là các nhãn huấn luyện với là số lượng các lớp.
* , thông thường .
* Học một ma trận khoảng cách phụ thuộc dữ liệu chẳng hạn sử dụng thuật giải số liệu học MEGM.
* Biến đổi tất cả dữ liệu huấn luyện và kiểm thử bằng ma trận A.
* Áp dụng LSVM vào các dữ liệu huấn luyện và kiểm thử đã được biến đổi.

# Kết quả

Trong phần này, nhóm tác giả nói về kết quả thu được ứng với nhiều loại dữ liệu. Dữ liệu khuôn mặt, USPS, Isolet, và Coil100 được sử dụng và chúng được tiền xử lý để tăng hiệu quả. Việc tiền xử lý hình ảnh sử dụng PCA là một hướng tiếp cận phổ biến trong nhận dạng đối tượng nhằm mục đích giảm số chiều của đối tượng; mà mục đích chính là giúp giảm chi phí trong tính toán. Trong thực nghiệm này, các kết quả nhận được bởi việc giảm số chiều của dữ liệu ảnh bằng cách chiếu chúng lên một vài eigenfaces. Số lượng eigenfaces được dùng cho mỗi loại dữ liệu như bảng 1.

Hiệu suất của LSVM và LASVM được so sánh dựa trên các kỹ thuật sau, lưu ý rằng tất cả các công thức SVM được huấn luyện với một hạt nhân Gaussian và sử dụng chiến lược onve-versus-all.

* KNN: Phân loại trên 1 lân cận gần nhất.
* SVM: Tham số C cho SVM được điều chỉnh thông qua kiểm tra chéo (tìm kiếm từ tập {1; 10; 100; 1000}). Giá trị của σ là tập hợp các khoảng cách trung bình của k lân cận gần nhất (đã được nói đến trong phần I).
* MEGM: phân loại trên 1 lân cận gần nhất trong không gian chuyển đổi. Việc chuyển đổi được kiểm soát bởi ma trận học được bằng thuật toán MEGM.
* OSVM: Đối với data UCIML, để có được sự so sánh tốt nhất giữa LSVM và LASVM với SVN chuẩn, cả tham số C và σ được tối ưa hóa cho SVM bằng cách **xác nhận chéo** (chúng ta gọi ‘optimized SVM (OSVM)). SVM cho kết quả tốt nhất với C = {1, 10, 100, 1000} và σ = {0.1, 0.5, 1, 2, 3, 5}.

Đối với dữ liệu khuôn mặt, USPS, Isolet và Coil100, mỗi thực nghiệm được lặp lại 10 lần để thu được kế quả trung bình và độ lệch chuẩn. Lưu ý rằng không có sự tối ưu nào được thực hiện cho cả LSVM và LASVM. Tham số *C* được thiết lập bằng 10 và σ bằng 1 cho tất cả các thực nghiệm cho LSVM và LASVM.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Database** | **#Data** | **Features** | **#Classes** | **PCA** | **#Train/Class** | **#Test/Class** |
| Yalefaces | 165 | 77760 | 15 | 50 | 4 | 7 |
| YalefacesB | 5850 | 307200 | 10 | 20 | 10 | 20 |
| Caltechfaces | 435 | 47500 | 29 | 30 | 5 | 10 |
| CaltechfacesB | 435 | 47500 | 2 | 30 | 50 | 100 |
| AT&Tfaces | 400 | 10304 | 40 | 50 | 5 | 5 |
| Coil100 | 7200 | 16384 | 100 | 50 | 10 | 10 |
| USPS | 9298 | 256 | 10 | 50 | 20 | 10 |
| Isolet | 6238 | 617 | 26 | 30 | 20 | 10 |

**Bảng 1:** Chi tiết các cơ sở dữ liệu được dùng để phân lớp

1. Dữ liệu khuôn mặt, USPS, Isolet và Coil

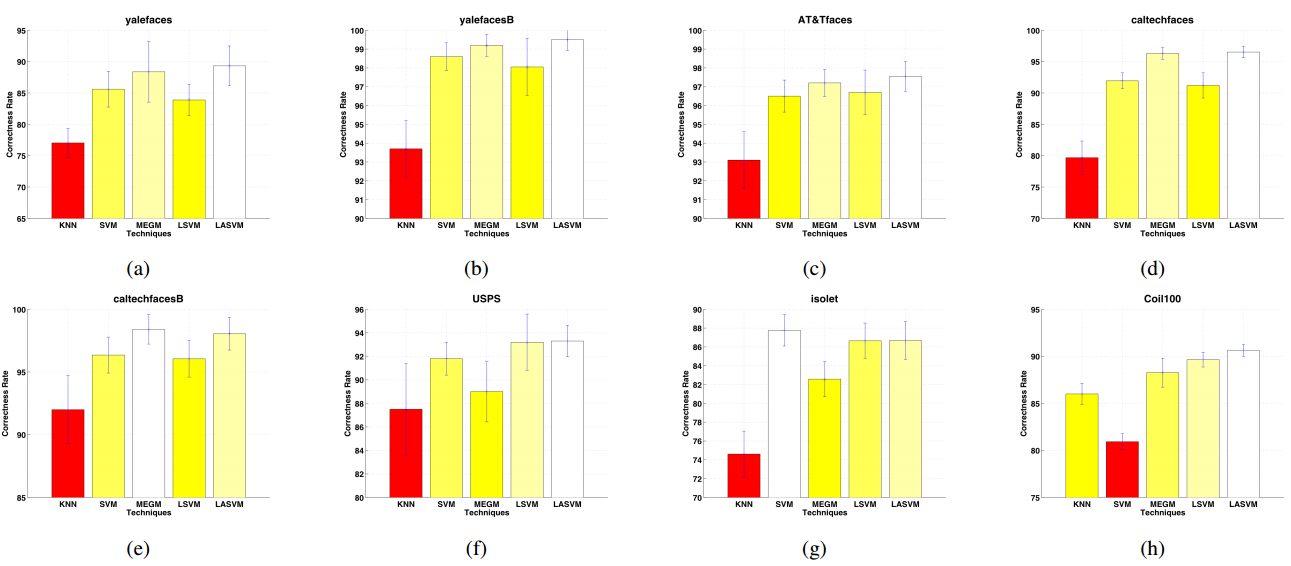
Yalefaces, YalefacesB, AT&T và Caltechfaces được sử dụng cho việc thực nghiệm SVM cục bộ. Dữ liệu Yalefaces, YalefacesB, và AT&T được biết đến rộng rãi trong các nghiên cứu nhận dạng khuôn mặt. Caltechfaces và CaltechfaceB là các hình ảnh được tạo thành từ các loại khuôn mặt trong cơ sở dữ liệu đối tượng Caltech-101.

Dữ liệu Caltech-101 bao gồm 435 hình ảnh của 20 người. Dữ liệu Caltechfaces trong bảng I dựa trên việc chia tách các dạng khuôn mặt thành 20 mục, mỗi mục thuộc về một người khác nhau. Mặc khác, CaltechfacesB trong bảng I được dựa trên việc phân chia Caltech-101 thành 2 lớp: nam và nữ.

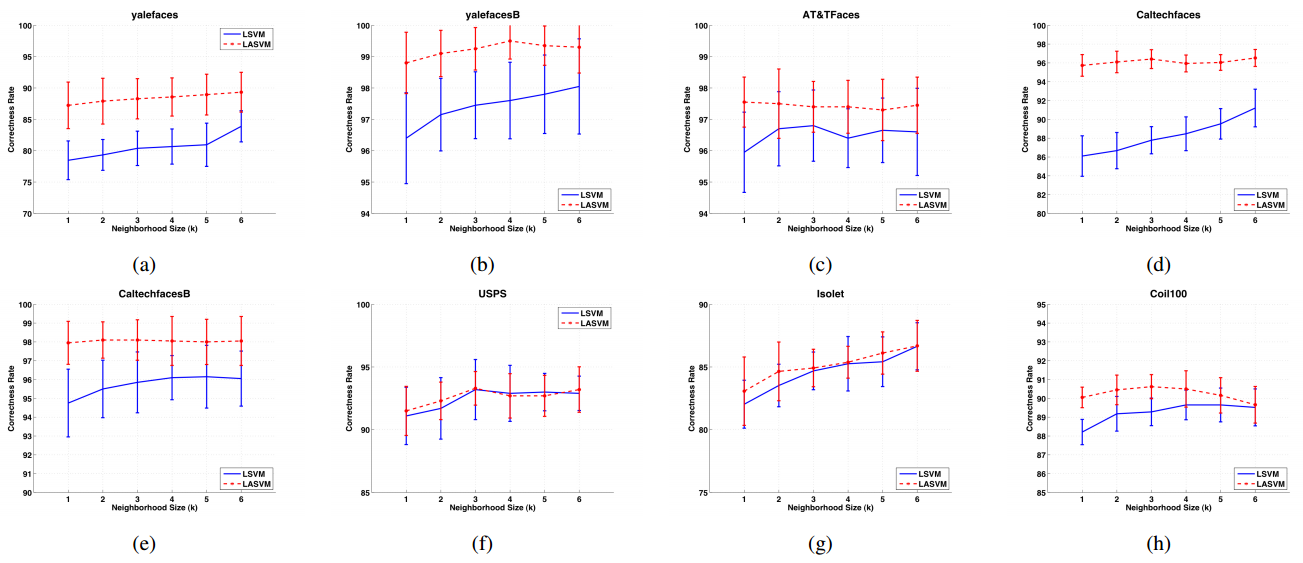
Kết quả của LSVM và LASVM cùng với phân lớp KNN, SVM và MEGM cho mỗi loại dữ liệu khác nhau được thể hiện ở hình 1. LASVM được đề xuất không chỉ cải thiện SVM chuẩn, mà còn thực hiện tốt hơn hoặc tương đương so với thuật toán cạnh tranh MEGM. LASVM cho kết quả tốt nhất trong 6 trên 8 dữ liệu, trong khi đó MEGM cho kết quả tốt nhất chỉ trên 1 dữ liệu. Mặt khác, LSVM cải thiện hiệu suất của SVM chuẩn trong các trường hợp dữ liệu USPS và Coil100. Trong các dữ liệu khác, nó đã không mang đến nhiều cải thiện so với SVM chuẩn. Điều nay cho thấy mặc dù LSVM tính toán hiệu quả, nhưng lại có thể không phải là lựa chọn tốt nhất khi phân loại dữ liệu.

Hình 2 so sánh LSVM và LASVM với kích thước lân cận thay đổi. Chúng tá có thể thấy được một điểm chung khá thú vị. Hiệu suất của LSVM tăng lên khi k tăng lên, trong khi đó hiệu suất của LASVM trong nhiều trường hợp không bị ảnh hưởng bởi k.

LASVM ít thiên về kích thước số lân cận so với LSVM cho thấy tầm quan trọng của các lân cận đúng với dự đoán. Vì trong không gian ban đầu, khu vực lân cận của các điểm truy vấn không giữ đủ thông tin để dự đoán, nên một khu vực lân cận lớn hơn là điều bắt buộc. Nhưng kể từ khi LASVM tìm thấy một ranh giới cục bộ trong không gian chuyển đổi, hiệu quả của nó là tốt như nhau trong khu vực lân cận lớn cũng như khu vực lân cận nhỏ hơn. Điều nay nhấn mạnh hiệu quả của LASVM.



**Hình 1:** So sánh tý lệ chính xác của phương pháp LSVM, LASVM với các phương pháp KNN, SVM, MEGM trên dữ liệu khuôn mặt, USPS, Isolet và Coil100

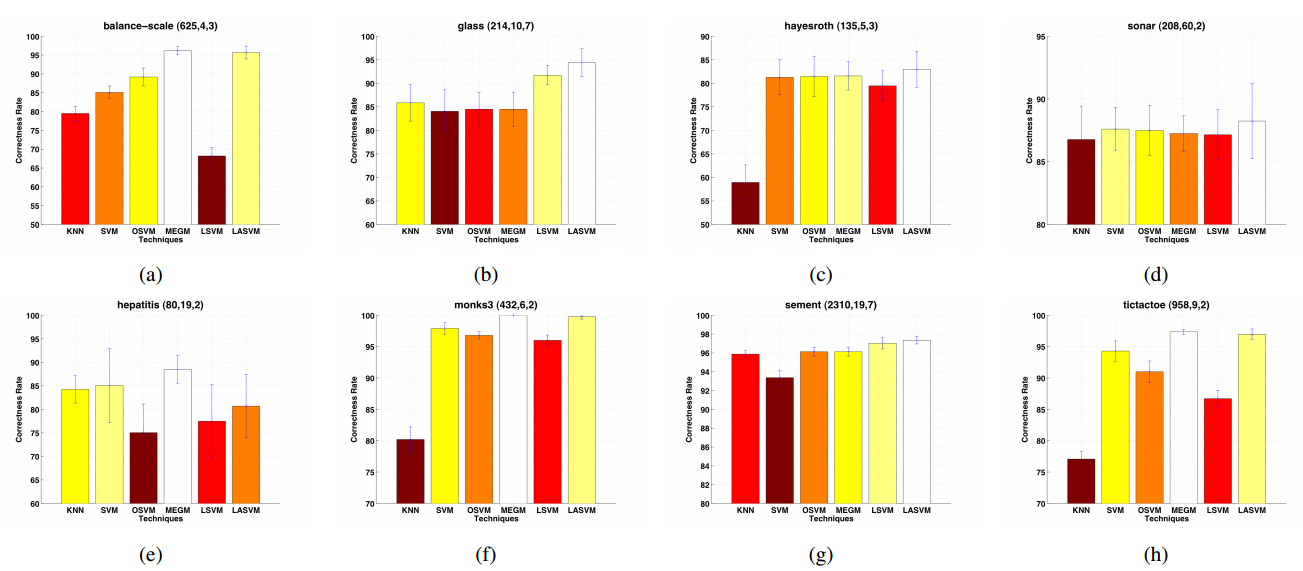


**Hình 2:** So sánh tỷ lệ chính xác của LSVM với LSVM cho dữ liệu khuôn mặt, USPS, Isolet và Coil100 khi thay đổi kích thước vùng lân cận

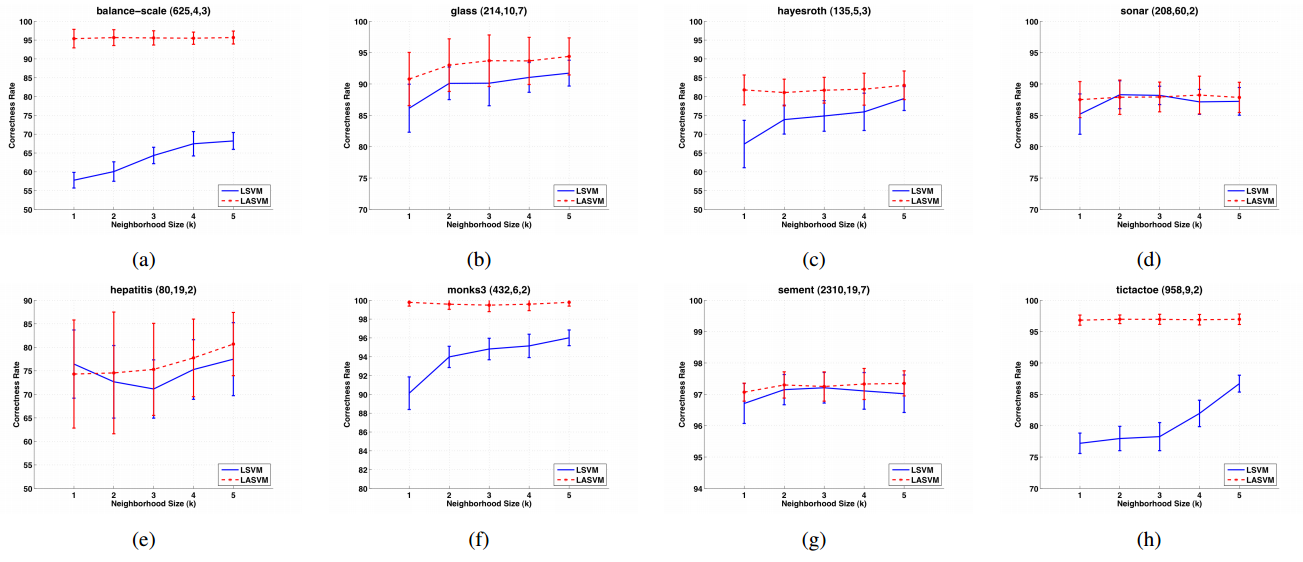
1. Kho dữ liệu UCI-ML

Tỷ lệ chính xác của mỗi phương pháp cho một số cơ sở dữ liệu UCI-ML được thể hiện trong hình 3. Số lượng dữ liệu, thuộc tính là lớp của mỗi cơ sở dữ liệu được trình bày trong tiêu đề. Tỷ lệ chính xác của từng phương pháp thu được bằng cách sử dụng 40 vòng kiểm tra chéo 2 lần. Trước khi huấn luyện, các thuộc tính được bình thường hóa để có một số không và nó có nghĩa là một đơn vị không đúng.

Như đã trình bày ở trên, một phần của việc so sánh kết quả LSVM với LASVM với KNN, SVM và MEGM, kết quả còn được so sánh với OSVM. Ta có thể thấy kết quả ở hình 3, LASVM và MEGM cho kết quả tốt trong hầu hết các cơ sở dữ liệu. Ngoài ra hiệu suất của LSVM và LASVM có xu hướng tương tự như các cơ sở dữ liệu ở phần A. Đó là, LASVM cho kết quả tốt hơn LSVM trong hầu hết cơ sở dữ liệu. Thứ hai, hiệu suất của LSVM được cải thiện khi mà kích thước vùng lân cận được tăng lên. Và thứ ba, LASVM thực hiện tốt hơn so với LSVM trên hầu hết các cơ sở dữ liệu và hiệu suất của nó thì không chịu nhiều ảnh hương của kích thước vùng lân cận (k).



**Hình 3:** So sánh tý lệ chính xác của phương pháp LSVM, LASVM với các phương pháp KNN, SVM, OSVM, MEGM trên dữ liệu UCIML



**Hình 4:** So sánh tỷ lệ chính xác của LSVM với LSVM cho dữ liệu UCIML khi thay đổi kích thước vùng lân cận

Có ba điều cần được đề cập. Thứ nhất, như đã mô tả, không có giá trị nào được tối ưu cho LSVM cũng như LASVM. Do đó kết quả thu được khuyến khích không chỉ cho LASVM mà còn cho LSVM. Trong một số trường hợp thì hiệu xuất của LSVM là tốt. Ví dụ, trong cơ sở dữ liệu Coil100, kết quả là LSVM đạt được hiệu suất lớn hơn so với SVM chuẩn. Thứ hai, do hạn chế về không gian, các kết quả đã không thống kê thời gian tính toán. Trong các thực nghiệm, chúng ta thấy rằng việc train cho LSVM và LASVM hiệu quả hơn về mặt tính toán so với SVM chuẩn, miễn là chúng ta giữ kích thước vùng lân cận nhỏ vừa phải. Vì vậy, với góc nhìn từ hiệu quả tính toán thì việc train LSVM và LASVM có thể được coi là hấp dẫn. Thứ ba, như mô tả trong kết quả, mặc dù LSVM cải thiện hiệu suất SVM trong một vài trường hợp, song có thể bị suy giảm hiệu xuất trong các trường hợp khác.

Điều này cho thấy thuật toán LSVM được đề xuất trong [9] không thể được sử dụng với khoảng cách Euclide. Lưu ý rằng Zhang trong [9] đã đề xuất sử dụng LSVM với một độ đo khoảng cách được thiết kế đặc biệt. Kết quả xây dựng SVM thích ứng cục bộ LASVM là vượt trội hơn so với LSVM.

# Chương trình minh họa

Nhóm đã tìm kiếm mã nguồn LSVM, MEGM và LASVM trong nhiều tuần nhưng không tìm được. Nhóm cũng đã liên hệ với hai tác giả chính của bài báo này (đồng thời cũng là tác giả viết bài báo về MEGM) để xin chương trình minh họa nhưng cũng không liên lạc được. Vì vậy trong thời điểm hiện tại nhóm chưa thể trình bày phần này.

# Tổng kết

Trong phạm vi nghiên cứu bài báo, nhóm tác giả đã đề xuất sử dụng thuật toán Support Vector Machine (SVM) trong phạm vi láng giềng địa phương thích ứng (LASVM). Thuật toán LASVM đã cho ra kết quả cải thiện đáng kể về hiệu năng không chỉ với LSVM, mà còn với SVM, KNN và phân lớp MEGM trên các khuôn mặt, đối tượng, ký tự và UCIML. Bên cạnh đó, không như LSVM, thuật giải LASVM có hiệu năng không bị ảnh hưởng bởi kích thước của láng giềng. Những kết quả này đã mang lại sự hứa hẹn và chỉ ra những hướng đi cho nghiên cứu tương lai.

Ngoài ra, các kết quả cũng nhấn mạnh những lợi thế của phương pháp LSVM và đưa ra cách để làm giảm đi những vấn đề trong việc các điều chỉnh tham số hạt nhân. Tuy các nghiên cứu trên LSVM không phải là thay thế hoàn toàn thuật toán SVM nguyên thủy, nhưng nó tạo ra một sự hứa hẹn trong việc kết hợp SVM và KNN. Thực tế, LSVM rõ ràng là tốt hơn trong trường hợp số lượng các lớp rất lớn, bởi nó nhanh hơn so với SVM và có hiệu năng tốt hơn so với KNN. Còn với các thuật giải LASVM, thì ngược lại, cần phải được nghiên cứu hơn nữa với các thuật toán số liệu học khác.

# Tài liệu tham khảo

1. Nayyar A. Zidi, David McG. Squire, “Local Adaptive SVM for Object Recognition”, Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference, Dec 2010.
2. O. Chapelle, V. Vladimir, O. Bousquet, and S. Mukherjee, “Choosing multiple parameters for support vector machines,” Journal of Machine Learning Research, 2002.
3. A. Frome, Y. Singer, and J. Malik, “Image retrieval and classification using local distance functions,” in Proceedings of Neural Inforamtion and Processing Systems, 2006.
4. M.-E. Nilsback and A. Zisserman, “A visual vocabulary for flower classification,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2006, pp. 1447–1454.
5. A. Berg, T. Berg, and J. Malik, “Shape matching and recognition using low distortion correspondence,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005.
6. S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” IEEE Transactions on Pattern Recognition, 2005.
7. T. Hastie and R. Tibshirani, “Discriminative adaptive nearest neighbor classification,” IEEE transactions on Pattern Analysis and Machine Intelligence, 1996.
8. E. Rosch, “Natural categories,” Cognitive Psychology, 1973.
9. H. Zhang, A. Berg, M. Maire, and J. Malik, “Svm-knn: Discriminative nearest neighbor classification for visual category recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006.
10. M. Varma and D. Ray, “Learning the discriminative powerinvariance trade-off,” in Proceedings of the International Conference on Computer Vision, 2007.
11. J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, “Local features and kernels for classification of texture and object categories: A comprehensive study,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (Workshop paper), 2006.
12. T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning. Springer Series in Statistics, 2001.
13. H. Cheng, P. Tan, and R. Jin, “Localized support vector machine and its efficient algorithm,” in Proceedings of the SIAM International Conference on Data Mining, 2007.
14. N. Zaidi and D. M. Squire, “A gradient-based metric learning algorithm for k-nn classifiers,” in Proceedings of the Australasian Joint Conference on Artificial Intelligence, 2010.
15. J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, “Neighborhood component analysis,” in Proceedings of Neural Inforamtion and Processing Systems, 2005.
16. K. Weinberger, J. Blitzer, and L. Saul, “Distance metric learning for large margin nearest neighbor classification,” in Proceedings of Neural Inforamtion and Processing Systems, 2006.