**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

A picture containing logo

Description automatically generated

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

Môn: Phân tích thống kê dữ liệu nhiều biến

Học kỳ II (2021-2022)

**ĐỀ TÀI**

**CLASSIFICATION**

Giảng viên hướng dẫn: PGS.TS Lý Quốc Ngọc

Nhóm đồ án : Warriors

Thành viên: 19120456 – Nguyễn Phan Quốc Bảo (nhóm trưởng)

19120656 – Phan Văn Thắng

19120716 – Lê Trọng Việt

**Mục lục**

[1 Giới thiệu vấn đề 1](#_Toc106917413)

[1.1 Khái niệm 1](#_Toc106917414)

[1.2 Ứng dụng 2](#_Toc106917415)

[2 Phân lớp bằng cực tiểu ECM 2](#_Toc106917416)

[2.1 Phân 2 lớp bằng cực tiểu ECM 2](#_Toc106917417)

[**2.1.1** Phát biểu bài toán 2](#_Toc106917418)

[**2.1.2** Phương pháp 2](#_Toc106917419)

[**2.1.3** Phân lớp với 2 quần thể có phân phối chuẩn 7](#_Toc106917420)

[2.2 Phân đa lớp bằng cực tiểu ECM 13](#_Toc106917421)

[**2.2.1** Phát biểu bài toán 13](#_Toc106917422)

[**2.2.2** Phương pháp 13](#_Toc106917423)

[**2.2.3** Phân đa lớp với các quần thể có phân phối chuẩn 15](#_Toc106917424)

[3 Linear Discriminant Analysis 18](#_Toc106917425)

[3.1 Giới thiệu 18](#_Toc106917426)

[3.2 Bài toán LDA cho 2 lớp 18](#_Toc106917427)

[3.3 Bài toán LDA cho nhiều lớp 22](#_Toc106917428)

[3.4 So sánh LDA với PCA 23](#_Toc106917429)

[4 Ứng dụng 24](#_Toc106917430)

[4.1 Cài đặt môi trường 24](#_Toc106917431)

[4.2 Phân lớp sử dụng LDA 24](#_Toc106917432)

[**4.2.1** LDA cho tập dữ liệu hoa Iris 24](#_Toc106917433)

[**4.2.2** LDA cho tập dữ liệu chữ số viết tay 31](#_Toc106917434)

[4.3 Phân lớp sử dụng ECM 37](#_Toc106917435)

[**4.3.1** Phân lớp tập dữ liệu hoa Iris bằng cực tiểu ECM 37](#_Toc106917436)

[**4.3.2** Phân lớp tập dữ liệu chữ số viết tay bằng cực tiểu ECM 40](#_Toc106917437)

# Giới thiệu vấn đề

## Khái niệm

Với sự bùng nổ lượng lớn dữ liệu như hiện nay, phân lớp dữ liệu (**classification**) đang là một trong những hướng nghiên cứu chính. Chúng ta cần phân biệt rõ giữa phân lớp dữ liệu (**classification**) và phân biệt dữ liệu (**discrimination**):

* Discrimination (**separation**) nhằm mô tả những đặc trưng khác biệt của dữ liệu quan sát được so với quần thể.
* Classification (**allocation**) sắp xếp dữ liệu quan sát được vào 2 hay nhiều lớp đã được gán nhãn trước.

Hình minh họa bài toán phân lớp:

Chart, scatter chart

Description automatically generated

Nguồn ảnh: <https://machinelearningcoban.com/2017/01/21/perceptron/>

Ở hình minh họa trên, số quần thể là 2. Tại hình 1, tam giác màu xám vẫn chưa thuộc quần thể nào. Đến hình 2, tam giác đã được phân vào quần thể màu đỏ.

Dù phân lớp (**classification**) và hồi quy tuyến tính (**linear regression**) đều là bài toán học có giám sát (**supervised learning**), nhưng khác với dự đoán dùng để xây dựng mô hình với các giá trị liên tục (**continous value**), thì phân lớp xây dựng mô hình với các giá trị rời rạc (**discrete value**) hay những nhãn xác định (**categorical label**). Có nghĩa là phân lớp thao tác với dữ liệu có giá trị đã được biết trước.

Quá trình phân lớp dữ liệu gồm 2 bước:

* Bước 1: quá trình phân tích phân biệt (discrimination)

Đầu vào là 1 tập dữ liệu training gồm các bộ dữ liệu (*data tuple*) đã được gán nhãn. Sau đó, mô hình sẽ mô tả đặc trưng khác biệt của bộ dữ liệu mình đang quan sát so với các bộ còn lại. Đầu ra bước này là quy tắc để phân lớp (mô hình phân lớp).

* Bước 2: quá trình phân lớp (classification)

Ở giai đoạn này, chúng ta sử dụng tập dữ liệu mới chưa được gán nhãn, sử dụng quy tắc phân lớp đã xây dựng, tiến hành đánh nhãn cho tập dữ liệu mới.

## Ứng dụng

Trong dự báo thời tiết, dựa vào các thông số độ ẩm, sức gió, nhiệt độ,... của ngày hôm nay và những ngày trước đó để dự báo ngày mai nắng hay mưa.

Trong y khoa, dựa vào các đặc điểm kích thước, tốc độ phát triển,... của khối u để dự đoán khối u lành tính hay ác tính.

# Phân lớp bằng cực tiểu ECM

## Phân 2 lớp bằng cực tiểu ECM

### Phát biểu bài toán

Cho là một vector ngẫu nhiên.

lần lượt là hàm mật độ xác suất của 2 quần thể và .

**Vấn đề đặt ra:** Với bộ giá trị dữ liệu cụ thể , ta phải tìm quy tắc phân lớp của từng quần thể để xác định sẽ được gán nhãn hay sao cho xác suất phân lớp lỗi là ít nhất có thể.

### Phương pháp

* Xét là không gian mẫu các trường hợp gán nhãn có thể cho .

là tập con của khi phân loại vào quần thể .

là tập con của mà chúng ta phân loại vào .

A picture containing text, accessory

Description automatically generated

Hình minh họa không gian mẫu với số chiều p = 2

* Xác suất có điều kiện của việc phân lớp vào là sai (khi thuộc về ):

(1)

* Xác suất có điều kiện của việc phân lớp vào là sai (khi thuộc về ):

(2)

**Dưới đây là hình ảnh minh họa cho xác suất phân lớp sai:**

Diagram

Description automatically generated

|  |
| --- |
| Giải thích cho hình minh họa |
| Đường thẳng vuông góc với trục x chính là đường phân chia 2 lớp và .  Vùng sọc thể hiện việc phân lớp sai đối tượng thuộc lớp vào lớp  Vùng sọc thể hiện việc phân lớp sai đối tượng thuộc lớp vào lớp  Việc của chúng ta là đi tìm đường thẳng đó, sao cho tổng diện tích của 2 phần phân lớp sai (trong hình là vùng sọc và vùng chấm).  Ta tính toán được: đường thẳng trên đi qua giao điểm của 2 hàm mật độ xác suất và sẽ cho kết quả tốt nhất (tổng diện tích của 2 phần phân lớp sai là nhỏ nhất). |

Gọi và lần lượt là xác suất của một đối tượng thuộc về quần thể và , khi đó:

Khi đó ta có được các xác suất có điều kiện sau:

* Gọi là chi phí phân lớp sai đối tượng vào khi đối tượng đó thuộc , với ,. Ta có bảng chi phí phân loại sai như hình dưới.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Classify as | |
|  |  |
| Truth |  | 0 |  |
|  |  | 0 |

* Từ đó, ta có hàm tính độ lỗi ECM (***excepted cost of misclassification***) của việc phân lớp sai:

(3)

* Để việc phân lớp đạt hiệu quả tốt nhất, ta cần cực tiểu hóa hàm tính độ lỗi ECM:

Thay (1) và (2) vào (3) ta được:

(4)

Vì , do đó:

* Thay vào (4):

Để ECM đạt giá trị nhỏ nhất thì:

Đưa ra kết luận, giá trị được phân lớp khi:

**:**

* Tương tự ta thay vào (4):

Để ECM đạt giá trị nhỏ nhất thì:

Đưa ra kết luận, giá trị được phân lớp khi:

**:**

|  |
| --- |
| Ví dụ 2.1.1.1 |
| Cho 2 quần thể có hàm mật độ xác suất lần lượt là:   1. Xác định quy luật phân lớp khi 2. Xác định quy luật phân lớp khi 3. Xác định quy luật phân lớp khi     b)  c) |

### Phân lớp với 2 quần thể có phân phối chuẩn

Các mô hình phân lớp với quần thể có phân phối chuẩn thường được sử dụng nhiều trong thống kê vì tính đơn giản và hiệu quả cao trên nhiều mô hình thực tế.

Gọi và là 2 hàm mật độ xác suất có phân phối chuẩn của lớp và với mean vector và covariance matrix tương ứng là , và ,

Hàm mật độ xác suất của lớp :

Của lớp :

##### Trường hợp

Thực hiện cực tiểu hàm ECM trên vùng ta được bất đẳng thức:

Tương tự, thực hiện cực tiểu hàm ECM trên vùng :

Như vậy ta có quy luật phân lớp:

* Gán nhãn một mẫu là nếu:
* Gán nhãn một mẫu là nếu:

Thực hiện phép biển đổi, ta được quy luật:

|  |
| --- |
| Chứng minh (2.1.3.1) |
| Ta có quy luật phân lớp ban đầu:  Thực hiện lấy logarit nepe 2 vế 2 bất đẳng thức:  Với x,y là 2 vector và A là 1 ma trận đối xứng, Ta có  Từ tính chất trên, ta phân tích vế trái bất đẳng thức:  Từ đó ta có được 2 bất đẳng thức ở (2.1.3.1) |

A picture containing racquetball

Description automatically generated

Ảnh minh họa: Phân lớp 2 quần thể có phân phối chuẩn với

Quy luật phân lớp phụ thuộc vào các giá trị của quần thể như , Thường sẽ không dễ tìm ra trong thực tế. Vì vậy để áp dụng quy luật phân lớp vào mẫu, ta cần thực hiện xấp xỉ những đại lượng của quần thể bằng đại lượng của mẫu.

Từ dữ liệu tập mẫu với biến thuộc và biến thuộc , ta tính được các đại lượng:

’

’

Ta có thể xấp xỉ các giá trị của quần thể như sau:

* Xấp xỉ trung bình quần thể bằng trung bình mẫu
* Do 2 quần thể có cùng covariance matrix, ta có thể xấp xỉ theo phương pháp trung bình có trọng số:

Quy luật phân lớp:

|  |
| --- |
| Ví dụ 2.1.3.1 |
| cho và là số quan sát thực hiện trên 2 quần thể đa biến có phân phối chuẩn có cùng covariance matrix  Các giá trị trung bình mẫu và ma trận hiệp phương sai mẫu như sau:   1. Xây dựng quy tắc phân lớp 2 quần thể 2. Mẫu quan sát mới thuộc về quần thể nào?   Giải  a)    =  Theo quy tắc ECM, ta cần so sánh biến vô hướng:  Với giá trị:  Biến thuộc nếu và thuộc trong trường hợp còn lại.  b)  Vậy, mẫu quan sát này thuộc về lớp |

##### Trường hợp

Biến đổi tương tự trường hợp cùng covariance matrix, ta được:

Trong đó:

Chart, line chart

Description automatically generated

Ảnh minh họa: Phân lớp 2 quần thể có phân phối chuẩn với

## Phân đa lớp bằng cực tiểu ECM

### Phát biểu bài toán

* Đặt số lớp (hay số quần thể) là ,

Cho là một vector ngẫu nhiên.

lần lượt là hàm phân phối xác suất của quần thể , ,…,

**Vấn đề đặt ra:** Với bộ giá trị dữ liệu cụ thể , ta phải tìm quy tắc phân lớp của từng quần thể để xác định sẽ được gán nhãn 1 trong các quần thể , ,…, .

### Phương pháp

* Đặt là hàm mật độ xác suất tương ứng với quần thể , với

Đặt là xác suất phân lớp đúng đối tượng vào lớp

Đặt là chi phí phân lớp sai đối tượng vào k khi đối tượng đó thuộc i, với . Khi thì

Đặt là tập các giá trị được phân loại vào quần thể

với . Và

* Cũng giống như bài toán phân 2 lớp, ta cần tìm 1 hàm độ lỗi để tính chi phí phân lớp sai cho đối tượng , và sau đó cực tiểu hóa hàm độ lỗi này.
* Tương tự công thức (3), độ lỗi của việc phân loại sai từ thành , hoặc , hoặc ,..., hoặc là:

Tương tự ta tính được .

* Với mỗi có điều kiện, ta nhân với xác suất phân lớp đúng . Ta được hàm tổng độ lỗi :

* Để phân lớp đạt hiểu quả tốt nhất, ta cần cực tiểu hóa hàm tổng độ lỗi . Thì khi đó, đạt nhỏ nhất khi ta phân lớp vào với

Giả sử, tất cả các chi phí phân loại sai bằng nhau (, với ). Khi đó, bài toán đưa về cực tiểu hóa hàm khi ta phân lớp vào với .

Tổng trên nhỏ nhất, khi số hạng bị bỏ qua đạt giá trị lớn nhất.

Do đó, trong trường hợp các chi phí phân loại sai bằng nhau, để phân lớp vào quần thể đạt hiểu qua nhất khi:

**, với và**

### Phân đa lớp với các quần thể có phân phối chuẩn

Gọi hàm mật độ xác suất theo phân phối chuẩn của lớp với có mean vector và covariance matrix là , :

*với*

Giả sử chi phí phân lớp lỗi là như nhau, tức: , ta thực hiện phân lớp vào nếu:

Hay:

Mà:

Nhận thấy rằng là một hằng số. Vậy, ta gán vào lớp nếu đại lượng discrimination score đạt max với:

#### Trường hợp các lớp đều có như nhau

discrimination score có dạng:

Để áp dụng vào bài toán phân lớp thực tế, ta cần xấp xỉ các giả trị quần thể trên bằng các giá trị của mẫu như sau:

* Xấp xỉ trung bình quần thể bằng trung bình mẫu
* X, với là trung bình có trọng số các ma trận hiệp phương sai mẫu của các lớp:

|  |
| --- |
| Ví dụ 2.2.3.1 |
| khảo sát 3 lớp ngẫu nhiên có phân phối chuẩn và có cùng covariance matrix, mỗi lớp có 3 mẫu như sau:  Biết rằng, xác suất của từng lớp là , hãy phân loại vào 1 trong các lớp trên  Giải  Trung bình mẫu và ma trận hiệp phương sai mẫu:  Trung bình trọng số các ma trận hiệp phương sai:  Discrimination score:  Thay vào các biểu thức:  Thấy rằng discrimination score có giá trị lớn nhất, vì vậy ta phân lớp vào lớp |

# Linear Discriminant Analysis

## Giới thiệu

Linear Discriminant Analysis (LDA) là một phương pháp giảm chiều dữ liệu cho bài toán classification. LDA có thể được coi là một phương pháp giảm chiều dữ liệu (dimensionality reduction), và cũng có thể được coi là một phương pháp phân lớp (classification), và cũng có thể được áp dụng đồng thời cho cả hai, tức giảm chiều dữ liệu sao cho việc phân lớp hiệu quả nhất. Số chiều của dữ liệu mới là nhỏ hơn hoặc bằng trong đó C là số lượng classes. Từ ‘Discriminant’ được hiểu là những thông tin đặc trưng cho mỗi class, khiến nó không bị lẫn với các classes khác. Từ ‘Linear’ được dùng vì cách giảm chiều dữ liệu được thực hiện bởi một ma trận chiếu (projection matrix), là một phép biến đổi tuyến tính (linear transform).

Giải bài toán LDA là đi tìm một phép chiếu sao cho tỉ lệ độ dị biệt giữa các lớp và độ dị biệt trong một lớp lớn nhất có thể.

## Bài toán LDA cho 2 lớp

* Ý tưởng cơ bản

Chart, histogram

Description automatically generated

Hình 3.1

* Hình 3.1a: khoảng cách giữa hai kì vọng lớn nhưng phương sai trong mỗi class cũng lớn khiến 2 phân phối chồng lên nhau (phần màu xám), tức là dữ liệu chưa discriminant
* Hình 3.1b: độ lệch chuẩn của hai class đều nhỏ, tức dữ liệu tập trung hơn. Tuy nhiên, vấn đề với trường hợp này là khoảng cách giữa hai class, được đo bằng khoảng cách giữa hai kỳ vọng , là quá nhỏ, khiến cho phần chồng lấn cũng chiếm môt tỉ lệ lớn, và cũng không tốt cho classification.
* Hình 3.1c: phương sai đủ nhỏ và khoảng cách giữa hai kỳ vọng đủ lớn, ta thấy rằng dữ liệu discriminant hơn.

Độ lệch chuẩn nhỏ thể hiện việc dữ liệu ít phân tán. Điều này có nghĩa là dữ liệu trong mỗi class có xu hướng giống nhau. Hai phương sai còn được gọi là các

Khoảng cách giữa các kỳ vọng là lớn chứng tỏ rằng hai classes nằm xa nhau, tức dữ liệu giữa các classes là khác nhau nhiều. Bình phương khoảng cách giữa hai kỳ vọng còn được gọi là .

Hai classes được gọi là discriminative nếu hai class đó cách xa nhau (between-class variance lớn) và dữ liệu trong mỗi class có xu hướng giống nhau (within-class variance nhỏ). Linear Discriminant Analysis là thuật toán đi tìm một phép chiếu sao cho tỉ lệ giữa between-class variance và within-class variance lớn nhất có thể.

* Xây dựng hàm mục tiêu

Giả sử có điểm dữ liệu trong đó điểm đầu tiên thuộc class thứ nhất và điểm sau thuộc class thứ hai. Ký hiệu là tập chỉ số các điểm thuộc class 1 và là tập các chỉ số thuộc class thứ 2. Phép chiếu dữ liệu xuống 1 đường thẳng có thể được mô tả bằng một vector hệ số , giá trị tương ứng của mỗi điểm dữ liệu mới là:

Vector kì vọng của mỗi class:

Khi đó khoảng cách giữa các kì vọng trong không gian mới là:

Các được định nghĩa là:

LDA là thuật toán đi tìm giá trị lớn nhất của hàm mục tiêu:

Tìm sự phụ thuộc giữa tử số và mẫu số của (4) vào :

* Tử số

còn được gọi là . Đây là một ma trận đối xứng nửa xác định dương.

* Mẫu số

Với được gọi là . Đây cũng là một ma trận đối xứng nửa xác định dương vì nó là tổng của hai ma trận đối xứng nửa xác định dương.

Như vậy, bài toán tối ưu cho LDA trở thành:

* Nghiệm của bài toán tối ưu

Nghiệm của trong (7) sẽ là nghiệm của phương trình đạo hàm của hàm mục tiêu J(w) bằng 0

Vì là một số vô hướng, ta suy ra w là một vector riêng của ứng với một trị riêng nào đó và trị riêng này bằng . Vậy để hàm mục tiêu đạt giá trị lớn nhất thì là trị riêng lớn nhất của

Nếu w là nghiệm thì cũng là nghiệm với k là một số thực khác 0 bất kỳ. Vậy ta có thể chọn w sao cho . Khi đó thay định nghĩa ở (5) vào (8) ta có:

Điều này có nghĩa ta có thể chọn

## Bài toán LDA cho nhiều lớp

* Bài toán đặt ra
* Độ dị biệt giữa các thành phần trong class (within class):

Within class variance của class k được tính

Với là ma trận có các cột giống hệt nhau và giống vector kì vọng

Có thể thấy với là ma trận có các cột giống hệt nhau và bằng vector kì vọng trong không gian ban đầu

Tổng độ dị biệt trong các class được tính theo công thức:

Với

Được gọi là within class covariance matrix

* Độ dị biệt giữa các class (between class)

Between-class lớn, có thể đạt được nếu tất cả các điểm trong không gian mới đều xa vector kỳ vọng chung 𝑒. Việc này cũng có thể đạt được nếu các vector kỳ vọng của mỗi class xa các vector kỳ vọng chung (trong không gian mới).

* Với cách định nghĩa và ý tưởng về within-class nhỏ và between-class lớn như trên, ta có thể xây dựng bài toán tối ưu như sau:

Tìm sao cho

* Nghiệm của bài toán

Để hàm 𝑓(𝑊) đạt cực đại thì đạo hàm của nó phải bằng 0

Suy ra các cột của ma trận là các vector riêng độc lập tuyến tính ứng với trị riêng lớn nhất của

## So sánh LDA với PCA

|  |  |
| --- | --- |
| PCA | LDA |
| Giảm số chiều của dữ liệu, chuyển dữ liệu từ không gian nhiều chiều sang không gian ít chiều hơn | |
| Xây dựng không gian mới với số chiều nhỏ hơn ban đầu sao cho giữ lại độ dị biệt lớn nhất có thể | Xây dựng không gian mới với số chiều nhỏ hơn ban đầu sao cho tỉ lệ độ dị biệt giữa các class và độ dị biệt của các thành phần trong từng class là lớn nhất |

# Ứng dụng

## Cài đặt môi trường

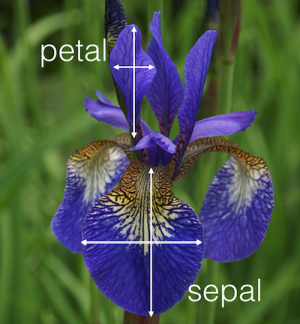
Các thuật toán phân lớp thống kê được nhóm cài đặt bằng ngôn ngữ Python 3 trên môi trường Jupiter Notebook. Các thư viện Python mà nhóm sử dụng cùng với cách cài đặt:

* Thư viện numpy để làm việc với ma trận và sử dụng các hàm toán học. Sử dụng pip để cài đặt
* Thư viện sklearn để lấy dữ liệu và chia dữ liệu ra thành tập train và tập test
* Thư viện matplotlib để trực quan hóa dữ liệu

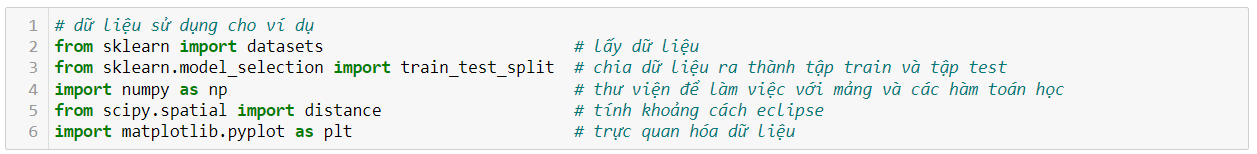
## Phân lớp sử dụng LDA

### LDA cho tập dữ liệu hoa Iris

* Tập dữ liệu Iris.csv:
* 50 mẫu của 3 loài Iris (setosa, virginica, versicolor)
* 4 biến: chiều dài, chiều rộng đài hoa, cánh hoa



* Chương trình
* Import các thư viện và các hàm cần thiết



* Chia data thành tập train và tập test

Graphical user interface, text

Description automatically generated

Một số dòng dữ liệu trong tập dữ liệu

Text

Description automatically generated

* Tạo class LDA

Graphical user interface, text

Description automatically generated

* Giảm chiều dữ liệu từ 4 chiều xuống 2 chiều

Text

Description automatically generated

Dữ liệu trong không gian mới

Chart, scatter chart

Description automatically generated

Chiếu tập dữ liệu test và gán nhãn dự đoán cho tập dữ liệu test bằng cách tính độ đo Euclid

Text

Description automatically generated

Dữ liệu trong không gian mới và các điểm trung tâm của các class

Chart, scatter chart

Description automatically generated

Độ chính xác:

Graphical user interface, text, application

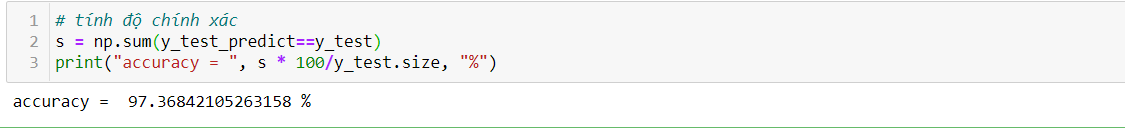
Description automatically generated

* Giảm chiều dữ liệu từ 4 chiều xuống 3 chiều

Graphical user interface, text, application, email

Description automatically generated

Độ chính xác



* Để nguyên số chiều để phân lớp

Graphical user interface, text, application, chat or text message

Description automatically generated

Độ chính xác

Graphical user interface, application

Description automatically generated

* Giảm số chiều xuống 1 chiều

Graphical user interface, text, application, email

Description automatically generated

Độ chính xác

Graphical user interface, text, application

Description automatically generated

* So sánh độ chính xác:

|  |  |  |  |
| --- | --- | --- | --- |
| Giữ nguyên số chiều | Giảm còn 3 chiều | Giảm còn 2 chiều | Giảm còn 1 chiều |
| 89.4% | 97.3% | 97.3% | 97.3% |

* Từ bảng so sánh trên, ta thấy đối với tập dữ liệu Iris, ta có thể giảm dữ liệu xuống còn 1 chiều để phân lớp và vẫn đảm bảo phân lớp với độ chính xác ở mức cao là 97.3%. Khi giữ nguyên số chiều của dữ liệu, các điểm dữ liệu không “Discriminant” nên việc phân lớp bằng Euclid không mang lại hiệu quả cao khi không giảm chiều.

### LDA cho tập dữ liệu chữ số viết tay

* Tập dữ liệu gồm có 5000 mẫu chia đều cho 10 lớp có nhãn là các số từ 0 đến 9. Với mỗi mẫu là một ảnh chữ viết tay có độ phân giải là 20x20, được lưu trữ dưới dạng ma trận (5000x400)
* Ví dụ 100 mẫu ngẫu nhiên được trực quan hóa

Shape, arrow

Description automatically generated

* Chương trình
* Import các thư viện và hàm cần thiết

Graphical user interface, text, application

Description automatically generated

* Tải dữ liệu từ file matlab

Graphical user interface, text, application, email

Description automatically generated

* Chia dữ liệu: với mỗi lớp, lấy 400 mẫu để train và 100 mẫu để test

Graphical user interface, text, application

Description automatically generated

* Thêm bias – sử dụng cho hàm softmax

Text

Description automatically generated

* Sử dụng softmax để phân lớp

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Hàm train và train mô hình

Text

Description automatically generated

Quá trình train

Text

Description automatically generated

Độ chính xác

Text

Description automatically generated with medium confidence

* Sử dụng LDA để phân lớp

Graphical user interface, text, application, email

Description automatically generated

Độ chính xác

Graphical user interface, text, application

Description automatically generated

Vẽ confusion matrix

Text

Description automatically generated

A picture containing diagram

Description automatically generated

Từ confusion matrix trên, ta thấy sử dụng LDA để phân lớp tập dữ liệu chữ số viết tay đưa ra kết quả rất tốt đối với tất cả các lớp, thấm chí nhãn số 1 có độ chính xác lên đến 99%

* So sánh softmax và LDA đối với bộ dữ liệu chữ viết tay

|  |  |  |
| --- | --- | --- |
|  | Thời gian chạy (giây) | Độ chính xác (%) |
| LDA | 1.2 | 87.6 |
| Softmax | 237.8 | 90.7 |

Từ bảng trên, ta có thể thấy đối với bộ dữ liệu chữ viết tay, LDA đem lại hiệu quả vượt trội cả về thời gian và độ chính xác so với softmax.

## Phân lớp sử dụng ECM

### Phân lớp tập dữ liệu hoa Iris bằng cực tiểu ECM

Sử dụng tập dữ liệu hoa Iris, tập dữ liệu gồm có

* + 50 mẫu của 3 loài Iris (setosa, virginica, versicolor) .
  + Mỗi mẫu gồm có 4 biến: chiều dài, chiều rộng đài hoa, cánh hoa.

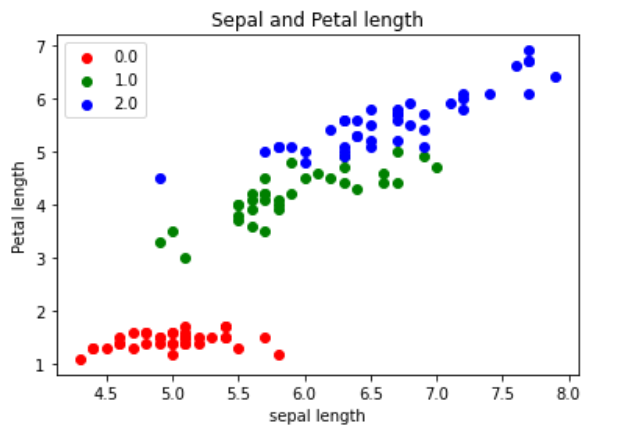
Một số scatter plot của dữ liệu:

* + Tương quan giữa petal length và petal width:

Chart, scatter chart

Description automatically generated

* + Tương quan giữa sepal length và petal length:



Để thực hiện phân loại bằng cực tiểu ECM, ta giả định:

* + Dữ liệu quần thể các hoa Iris có phân phối chuẩn.
  + Chi phí phân loại sai bằng nhau với mọi trường hợp phân loại sai.

Việc phân lớp mẫu vào quần thể đạt hiểu qua nhất khi:

, với và

Hàm phân lớp:

với k

Với đại lượng discrimination score theo lý thuyết:

Cài đặt:

|  |  |
| --- | --- |
| import numpy as np  import matplotlib.pyplot as plt  from sklearn import datasets  from sklearn.model\_selection import train\_test\_split  data = datasets.load\_iris()  X, y = data.data, data.target  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state =1 ) | Thư viện numpy phục vụ tính toán số và ma trận  Thư Viện matplotlib để vẽ đồ thị  Thư viện sklearn để lấy dataset và hàm chia dữ liệu  Dữ liệu dược chia ngẫu nhiên thành 112 mẫu train để xây dựng hàm phân lớp và 38 mẫu test để kiểm tra kết quả hàm phân lớp |

|  |  |
| --- | --- |
| train\_X = np.concatenate((X\_train, y\_train.reshape((y\_train.shape[0],1))),axis=1)  X\_0 = train\_X[train\_X[:,4] == 0]  X\_1 = train\_X[train\_X[:,4] == 1]  X\_2 = train\_X[train\_X[:,4] == 2]  train\_X = [X\_0, X\_1, X\_2]  p = []  for i in range(len(train\_X)):  p.append(len(train\_X[i])/ len(train)) | Sắp xếp lại theo thứ tự:  X\_0: hoa setosa  X\_1: hoa versicolor  X\_2: hoa virginica  Tính xác suất tiên nghiệm bằng cách lấy tổng số mẫu của từng loại hoa chia cho tổng số mẫu hoa. |

|  |  |
| --- | --- |
| Mean = []  for i in range(len(train\_X)):  Mean.append(np.mean(train\_X[i][:,:-1], axis=0))  Cov = []  for i in range(len(train\_X)):  Cov.append(np.cov(train\_X[i][:,:-1].T)) | Tính sample mean của từng loại họa  Tính covariance của từng loại hoa |

Tính discrimination score:

|  |
| --- |
| def discriminant\_score(x, prior, Mean, Cov):  score = np.log(prior) - (1.0/2) \* np.log(np.linalg.det(Cov))  score -= (1.0/2) \* np.dot(np.dot((x - Mean).T, np.linalg.inv(Cov)),(x-Mean))  return score |

Sau đó, khi có dữ liệu mới chưa có nhãn, ta thực hiện gán nhãn dữ liệu đó là nhãn có đại lượng discrimination score lớn nhất.

Thực hiện tính toán các đại lượng thống kê bằng 75% mẫu (112 mẫu) và đánh giá phương pháp bằng 25% mẫu còn lại (38 mẫu). Do dữ liệu khá đơn giản và tách bạch giữa các lớp, ta thu được độ chính xác là .

### Phân lớp tập dữ liệu chữ số viết tay bằng cực tiểu ECM

Bộ dữ liệu sử dụng gồm 5000 ảnh chữ số viết tay từ 0 đến 9, mỗi chữ số có 500 ảnh.

Một số hình ảnh trong bộ dữ liệu:

Shape, arrow

Description automatically generated

Dữ liệu sẽ được sử dụng như sau:

* + 4000 ảnh sẽ được sử dụng để tính toán các đại lượng thống kê (mỗi chữ số sử dụng 400 ảnh)
  + 1000 ảnh sẽ được sử dụng để đánh giá phương pháp (mỗi chữ số sử dụng 100 ảnh)

Cách cài đặt tương tự như phần phân lớp hoa Iris, tuy nhiên, ở bước tính discriminant score, ta sẽ tính toán trước đại lượng liên quan đến ma trận covariance của 10 lớp để giảm thiểu thời gian thực hiện tính toán (mỗi ảnh gồm 400 pixel, chuyển thành dữ liệu 400 chiều nên việc tính covariance mất nhiều thời gian).

Discriminant score:

|  |
| --- |
| def discriminant\_score(x, Mean, log\_prior\_log\_det\_Cov, pinv\_Cov):  score = log\_prior\_log\_det\_Cov  score -= (1.0/2) \* np.dot(np.dot((x - Mean).T, pinv\_Cov),(x-Mean))  return score |

Kết quả đánh giá trên tập test:

Confusion matrix:

A picture containing calendar

Description automatically generated

Các dữ liệu số 0, 2, 4 được dự đoán chính xác cao. Ngược lại, số 5 được dự đoán chính xác thấp nhất.

Số 9 bị dự đoán thành số 4 khá nhiều do sự tương đồng cao giữa 2 chữ số này.

Rất nhiều mẫu bị phân loại sai vào lớp 2.