TRƯỜNG ĐẠI HỌC THỦY LỢI

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO BÀI TẬP LỚN MÔN HỌC**

**HỌC MÁY**

**ĐỀ TÀI: PHÁT HIỆN BỆNH PARKINSON DỰA TRÊN THUẬT TOÁN MÁY VECTO HỖ TRỢ (SVM)**

**Giảng viên hướng dẫn: PGS.TS. Nguyễn Quang Hoan**

**Sinh viên thực hiện: Trần Thảo Vân**

*Hà Nội, 2022*

# **MỤC LỤC**

# 

**Contents**

**MỤC LỤC**.. 2

**LỜI CẢM ƠN**.. 3

**LỜI MỞ ĐẦU**.. 4

**I.** **TỔNG QUAN VỀ ĐỀ TÀI**. 5

**1.1.** **Giới thiệu về bài toán**. 5

**1.2.** **Thuật toán SVM**... 5

**1.2.1.** **Khái niệm**.. 5

**1.2.2.** **Giới thiệu thuật toán hoạt động**. 6

**1.3.** **Các bước xử lý cho bài toán**. 8

**II.** **THUẬT TOÁN MÁY VECTO HỖ TRỢ (SVM)** 9

**2.1.** **Giải thuật thuật toán máy vecto hỗ trợ (SVM)** 9

**2.2.** **Ứng dụng**. 9

**III.** **CHƯƠNG TRÌNH PHÁT HIỆN NGƯỜI BỆNH PARKINSON SỬ DỤNG THUẬT TOÁN MÁY VECTO HỖ TRỢ (SVM)** 10

**3.1.** **Dữ liệu thực nghiệm**.. 10

**3.2.** **Các bước xử lý bài toán (chi tiết)** 11

**IV.** **KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ**.. 12

**V.** **KẾT LUẬN**.. 13

**TÀI LIỆU THAM KHẢO**.. 14

# **LỜI CẢM ƠN**

Lời đầu tiên, em xin chân thành gửi lời cảm ơn tới thầy phụ trách môn học PGS.TS Nguyễn Quang Hoan đã tận tình giảng dạy, truyền đạt cho em những kiến thức cũng như kinh nghiệm quý báu trong suốt quá trình học. Thầy đã tận tình theo sát giúp đỡ, trực tiếp chỉ bảo, trong suốt quá trình học tập của em. Trong thời gian học tập, em không những tiếp thu thêm nhiều kiến thức bổ ích mà còn học tập được tinh thần làm việc, thái độ, phương pháp học tập nghiêm túc, hiệu quả. Đây là những điều rất cần thiết cho em trong quá trình học tập và làm việc công tác sau này. Do thời gian thực hiện tiểu luận có hạn và kiến thức còn nhiều hạn chế nên bài báo cáo của em chắc chắn không tránh khỏi những thiếu sót nhất định. Em rất mong nhận được ý kiến đóng góp của thầy để em có thêm kinh nghiệm và tiếp tục hoàn thiện tiểu luận trong tương lai. Em xin chân thành cảm ơn!

# **LỜI MỞ ĐẦU**

Vào đầu năm 2019, tập đoàn IBM và Quỹ Michael J. Fox (MJFF) đã công bố kế hoạch hợp tác và sử dụng trí tuệ nhân tạo (AI) và học máy (machine learning) để giải mã những bí ẩn và phức tạp xung quanh các triệu chứng và sự tiến triển của bệnh Parkinson Disease (PD). Tuy nhiên, nhiều câu hỏi về căn bệnh mãn tính này vẫn chưa được giải đáp, nhưng sự hiểu biết rõ hơn thông qua các thử nghiệm lâm sàng có thể cải thiện việc quản lý, chăm sóc bệnh nhân và phát triển hiệu quả hơn các loại thuốc chữa bệnh.

Học máy đã giúp tìm hiểu tốt hơn và nỗ lực nắm bắt những phức tạp xung quanh bệnh Parkinson. Các nhà nghiên cứu đã thiết kế các thuật toán sáng tạo sử dụng các yếu tố có thể che giấu các biểu hiện bên ngoài của bệnh Parkinson của một người nào đó, bao gồm các loại thuốc có thể làm giảm các triệu chứng như run, cải thiện khả năng kiểm soát vận động và sửa đổi các triệu chứng phổ biến khác.

Vì vậy, trong bài báo cáo của tiểu luận, em xin trình bày về vấn đề bài toán và hướng xử lý bài toán phát hiện người bệnh mặc bệnh Parkinson sử dụng thuật toán học máy hỗ trợ SVM để có thể hiểu rõ hơn về thuật toán cũng như việc áp dụng của thuật toán trong vấn đề cụ thể trong cuộc sống.

# **I.** **TỔNG QUAN VỀ ĐỀ TÀI**

## **1.1.** **Giới thiệu về bài toán**

Bệnh Parkinson (PD) là một chứng rối loạn thần kinh thoái hóa được đánh dấu bằng việc giảm nồng độ dopamine trong não. Nó biểu hiện thông qua sự suy giảm chuyển động, bao gồm cả sự biểu hiện của run và cứng, thường có ảnh hưởng rõ rệt đến giọng nói, bao gồm loạn nhịp (khó phát âm thanh), giảm giọng (giảm âm lượng) và đơn âm (giảm âm vực). Ngoài ra, suy giảm nhận thức và thay đổi tâm trạng có thể xảy ra và nguy cơ sa sút trí tuệ được tăng lên.

Chẩn đoán truyền thống về bệnh Parkinson liên quan đến việc bác sĩ lâm sàng lấy tiền sử thần kinh của bệnh nhân và quan sát các kỹ năng vận động trong các tình huống khác nhau. Vì không có xét nghiệm xác định nào trong phòng thí nghiệm để chẩn đoán bệnh Parkinson nên việc chẩn đoán thường khó khăn, đặc biệt là trong giai đoạn đầu khi các ảnh hưởng vận động chưa nghiêm trọng. Theo dõi sự tiến triển của bệnh theo thời gian cần bệnh nhân phải đến phòng khám nhiều lần. Một quy trình sàng lọc hiệu quả, đặc biệt là quy trình không yêu cầu thăm khám tại phòng khám sẽ có lợi. Vì bệnh nhân bệnh Parkinson có các đặc điểm giọng nói đặc trưng, ​​bản ghi âm giọng nói là một công cụ hữu ích và không xâm lấn để chẩn đoán. Nếu các thuật toán máy học có thể được áp dụng cho tập dữ liệu ghi âm giọng nói để chẩn đoán chính xác bệnh Parkinson, thì đây sẽ là một bước sàng lọc hiệu quả trước cuộc hẹn với bác sĩ.

## **1.2.** **Thuật toán SVM**

### **1.2.1.** **Khái niệm**

Máy vecto hỗ trợ (Support Vector Machine: SVM) được Vapnik nghiên cứu từ những năm 1965. Đến năm 1995, thuật toán được Vapnik và Chervonenkis chính thức phát triển mạnh mẽ, đứng hàng thứ 10 trong khai phá dữ liệu; được coi là công cụ mạnh cho những bài toán phân lớp phi tuyến tính; được xem là một trong các phương pháp phân lớp có giám sát không tham số, tinh vi nhất cho đến nay. Phương pháp này thực hiện phân lớp dựa trên nguyên lý cực tiểu hóa rủi ro (sai số) có cấu trúc SRM (Structural Risk Minimization).

SVM có dữ liệu đầu vào là một hàm nhân (Kernel Function) tạo ra mô hình mới, dẫn tới giải bài toán quy hoạch toàn phương (Quadratic Programming) luôn có kết quả tối ưu toàn cục (Global Optimzation). Bằng việc sử dụng hàm nhân, SVM cung cấp các mô hình hiệu quả, khá chính xác cho các bài toán phân lớp, hồi quy tuyến tính và phi tuyến. Các hàm công cụ đa dạng của SVM cho phép tạo không gian chuyển đổi để xây dựng mặt phẳng phân lớp.

SVM là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc đệ quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta vẽ đồ thị dữ liệu là các điểm trong n chiều. Sau đó chúng ta thực hiện tìm ra một đường phân cách trong mặt phẳng hoặc tìm một siêu mặt phẳng (Hyperplane) trong không gian nhiều chiều, ký hiệu siêu phẳng là h quyết định tốt nhất có thể chia các điểm trên không gian này thành 2 phần (hay hai lớp) riêng biệt tương ứng hai lớp: lớp “+” và lớp “-“ hay lớp “1” và lớp “-1”.

***Hình 1. Hyper-plane***

Support Vectors hiểu một cách đơn giản là một biên giới để chia hai lớp tốt nhất.

### **1.2.2.** **Giới thiệu thuật toán hoạt động**

Theo tư tưởng của học máy, siêu phẳng tối ưu là siêu phẳng tách được hai lớp xa nhau nhất có thể. Siêu phẳng trong không gian n chiều được biểu diễn bằng véc tơ pháp tuyến w=[w1,…, w1,…, wn] và độ lệch b của siêu phẳng so với gốc của tọa độ. Máy học SVM tìm siêu phẳng (w, b) để phân chia dữ liệu theo cách sau đây (Hình 2):

**Hình 2. *Siêu phẳng (w, b), lề của phân lớp tuyến tính***

Có thể mô tả hai siêu phẳng song song bằng hai công thức sau:

(6.1)

(6.2)

Kết hợp (6.1) và (6.2) bằng cách nhân hai vế của (6.1) với ; nhân hai vế của (6.2) với ta có:

(6.3)

Hoặc

Hiệu quả xác định siêu mặt phẳng này được quyết định bởi khoảng cách của điểm gần mặt phẳng nhất của mỗi lớp. Khoảng cách càng lớn thì mặt phẳng quyết định càng tốt đồng nghĩa với việc phân loại càng chính xác và ngược lại. Mục đích cuối cùng của phương pháp là tìm được khoảng cách hay biên lớn nhất. Trong SVM sử dụng thuật ngữ lề (Margin) để chỉ khoảng cách giữa hai siêu phẳng (trong trường hợp không gian 2 chiều là đường thẳng) đến 2 điểm dữ liệu gần nhất tương ứng với 2 phân lớp.

Lề (Margin) là khoảng cách giữa hai siêu phẳng hỗ trợ song song

(6.4)

trong đó, là độ lớn của véc tơ trọng số W theo chuẩn 2 (2-norm)

Trong Hình 3 hai siêu phẳng để phân chia tập các điểm thành hai lớp: lớp trên (dương), lớp dưới (âm). Vậy, siêu phẳng như thế nào được coi là tối ưu? siêu phẳng tối ưu là siêu phẳng phân cách 2 lớp dữ liệu cách xa nhau nhất. Ngoài ra, người ta còn dùng khái niệm khoảng cách (Distance) dương d+ (và âm d-) để chỉ khoảng cách từ đường thẳng (hoặc siêu phẳng) dương (và âm) đến đường thẳng (siêu phẳng) trung bình (lấy trung bình giữa siêu phẳng dương và âm) thể hiện trên Hình 3.

**Hình 3. Mô tả lề và khoảng cách với dữ liệu tách rời**

## **1.3.** **Các bước xử lý cho bài toán**

Các bước xử lý bài toán được tiến hành theo mô hình dưới đây:

[**Hình 4. Các bước xử lý bài toán**](https://www.figma.com/file/C6VbhRuVRztMOPqSHma6dJ/ML_draw?node-id=0%3A1)

Cụ thể các bước như sau:

Bước 1: Download dữ liệu từ trang [kaggle](https://www.kaggle.com/datasets/debasisdotcom/parkinson-disease-detection), tên bộ dữ liệu Parkinson Disease Detection.

Bước 2: Xử lý dữ liệu để dữ liệu ở dạng chuẩn hóa (làm sạch dữ liệu).

Bước 3: Phân chia dữ liệu, chia dữ liệu thành 80% dữ liệu train và 20% dữ liệu test (con số này có thể thay đổi)

Bước 4: Đưa dữ liệu đã xử lý vào mô hình có thuật toán máy vecto hỗ trợ (SVM)

Bước 5: Từ bước này thực hiện test dữ liệu để dự đoán một mẫu data (một người) có mắc bệnh Parkinson hay không

Bước 5.1: Chọn 1 mẫu dữ liệu và lưu vào một biến

Bước 5.2: Đưa mẫu dữ liệu ở bước 5.1 vào mô hình có thuật toán máy vecto hỗ trợ (SVM)

Bước 5.3: Mô hình đưa ra kết quả dự đoán và đưa ra kết luận

# **II.** **THUẬT TOÁN MÁY VECTO HỖ TRỢ (SVM)**

## **2.1.** **Giải thuật thuật toán máy vecto hỗ trợ (SVM)**

***Đầu vào (Input)***

· Tập dữ liệu huấn luyện gồm *m* phần tử: *x1,…, xi,…, xm* trong không gian *n* chiều, có nhãn (lớp, hay giá trị đầu ra) tương ứng là: *y1,…, yi,…, ym*; trong đó nhãn *yi={1,-1};*

· Hàm ***K*** với các tham số;

·

***Huấn luyện (Traning)***

· Giải bài toán quy hoạch toàn phương

(6.9)

Với điều kiện ràng buộc:

· Kết quả thu được nhân tử Lagrange

· Tập véc tơ *SV* (

· Tính độ lệch ***b*** từ các véc tơ hỗ trợ

***Phân lớp (Classifying)***

Phân lớp cho phần tử mới ***x*** theo công thức (6.10):

*Predict(****x****)=)*

## **2.2.** **Ứng dụng**

SVM là viết tắt của cụm từ support vector machine. Đây là một thuật toán khá hiệu quả trong lớp các bài toán phân loại nhị phân và dự báo của học có giám sát. Thuật toán này có ưu điểm là hoạt động tốt đối với những mẫu dữ liệu có kích thước lớn và thường mang lại kết quả vượt trội so với lớp các thuật toán khác trong học có giám sát.

Ưu điểm của SVM đó là:

- Đây là thuật toán hoạt động hiệu quả với không gian cao chiều (high dimensional spaces).

- Thuật toán tiêu tốn ít bộ nhớ vì chỉ sử dụng các điểm trong tập hỗ trợ để dự báo trong hàm quyết định.

- Chúng ta có thể tạo ra nhiều hàm quyết định từ những hàm kernel khác nhau. Thậm chí sử dụng đúng kernel có thể giúp cải thiện thuật toán lên đáng kể.

Chính vì tính hiệu quả mà SVM thường được áp dụng nhiều trong các tác vụ phân loại và dự báo, cũng như được nhiều công ty ứng dụng và triển khai trên môi trường production. Chúng ta có thể liệt kê một số ứng dụng của thuật toán SVM đó là:

- Mô hình chuẩn đoán bệnh. Dựa vào biến mục tiêu là những chỉ số xét nghiệm lâm sàng, thuật toán đưa ra dự báo về một số bệnh như tiểu đường, suy thận, máu nhiễm mỡ,…

- Trước khi thuật toán CNN và Deep Learning bùng nổ thì SVM là lớp mô hình cực kì phổ biến trong phân loại ảnh.

- Mô hình phân loại tin tức. Xác định chủ đề của một đoạn văn bản, phân loại cảm xúc văn bản, phân loại thư rác.

- Mô hình phát hiện gian lận.

# **III.** **CHƯƠNG TRÌNH PHÁT HIỆN NGƯỜI BỆNH PARKINSON SỬ DỤNG THUẬT TOÁN MÁY VECTO HỖ TRỢ (SVM)**

## **3.1.** **Dữ liệu thực nghiệm**

Trong bài tiểu luận này, em sử dụng bộ dữ liệu tải từ trang [kaggle](https://www.kaggle.com/datasets/debasisdotcom/parkinson-disease-detection).

Những thông tin thêm cho bộ dữ liệu:

*Matrix column entries (attributes):*

*name - ASCII subject name and recording number*

*MDVP:Fo(Hz) - Average vocal fundamental frequency*

*MDVP:Fhi(Hz) - Maximum vocal fundamental frequency*

*MDVP:Flo(Hz) - Minimum vocal fundamental frequency*

*MDVP:Jitter(%),MDVP:Jitter(Abs),MDVP:RAP,MDVP:PPQ,Jitter:DDP - Several*

*measures of variation in fundamental frequency*

*MDVP:Shimmer,MDVP:Shimmer(dB),Shimmer:APQ3,Shimmer:APQ5,MDVP:APQ,Shimmer:DDA - Several measures of variation in amplitude*

*NHR,HNR - Two measures of ratio of noise to tonal components in the voice*

*status - Health status of the subject (one) - Parkinson's, (zero) - healthy*

*RPDE,D2 - Two nonlinear dynamical complexity measures*

*DFA - Signal fractal scaling exponent*

*spread1,spread2,PPE - Three nonlinear measures of fundamental frequency variation*

## **3.2.** **Các bước xử lý bài toán (chi tiết)**

Code bài toán được lưu lại trong [link google colab](https://colab.research.google.com/drive/1jptLqM6eMKvkdJnIL3XHxOSRBzwME1sa?usp=sharing).

- Import các thư viện cần thiết cho việc xử lý bài toán

import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn import svm

from sklearn.metrics import accuracy\_score

- Đọc file dữ liệu Parkinson

# loading the data from csv file to a Pandas DataFrame

parkinsons\_data = pd.read\_csv('Parkinsson disease.csv')

- Một số thao tác để xem trước dữ liệu: head, shape, info

# printing the first 5 rows of the dataframe

parkinsons\_data.head()

# number of rows and columns in the dataframe

parkinsons\_data.shape

# getting more information about the dataset

parkinsons\_data.info()

- Kiểm tra có dữ liệu thiếu trong file dữ liệu

# checking for missing values in each column

parkinsons\_data.isnull().sum()

- Thống kê dữ liệu trong file dữ liệu

# getting some statistical measures about the data

parkinsons\_data.describe()

- Count dữ liệu

# distribution of target Variable

parkinsons\_data['status'].value\_counts()

- Group dữ liệu theo status (0: không mắc bệnh Parkinson; 1: mắc bệnh Parkinson)

# grouping the data bas3ed on the target variable

parkinsons\_data.groupby('status').mean()

- Xử lý dữ liệu: chia dữ liệu train và dữ liệu test (loại bỏ cột name và status)

X = parkinsons\_data.drop(columns=['name','status'], axis=1)

Y = parkinsons\_data['status']

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=2)

print(X.shape, X\_train.shape, X\_test.shape)

- Chuẩn hóa dữ liệu

scaler = StandardScaler()

scaler.fit(X\_train)

X\_train = scaler.transform(X\_train)

X\_test = scaler.transform(X\_test)

- Train dữ liệu trên mô hình học máy vecto hỗ trợ

model = svm.SVC(kernel='linear')

# training the SVM model with training data

model.fit(X\_train, Y\_train)

ð SVC(C=1.0, break\_ties=False, cache\_size=200, class\_weight=None, coef0=0.0, decision\_function\_shape='ovr', degree=3, gamma='scale', kernel='linear', max\_iter=-1, probability=False, random\_state=None, shrinking=True, tol=0.001, verbose=False)

- Xây dựng mô hình dự đoán:

input\_data = (197.07600,206.89600,192.05500,0.00289,0.00001,0.00166,0.00168,0.00498,0.01098,0.09700,0.00563,0.00680,0.00802,0.01689,0.00339,26.77500,0.422229,0.741367,-7.348300,0.177551,1.743867,0.085569)

# changing input data to a numpy array

input\_data\_as\_numpy\_array = np.asarray(input\_data)

# reshape the numpy array

input\_data\_reshaped = input\_data\_as\_numpy\_array.reshape(1,-1)

# standardize the data

std\_data = scaler.transform(input\_data\_reshaped)

prediction = model.predict(std\_data)

print(prediction)

if (prediction[0] == 0):

print("The Person does not have Parkinsons Disease")

else:

print("The Person has Parkinsons")

ð [0]

The Person does not have Parkinsons Disease

# **IV.** **KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ**

- Mã chương trình đánh giá mô hình trên dữ liệu thực nghiệm:

# accuracy score on training data

X\_train\_prediction = model.predict(X\_train)

training\_data\_accuracy = accuracy\_score(Y\_train, X\_train\_prediction)

print('Accuracy score of training data : ', training\_data\_accuracy)

ð Accuracy score of training data : 0.8846153846153846

# accuracy score on training data

X\_test\_prediction = model.predict(X\_test)

test\_data\_accuracy = accuracy\_score(Y\_test, X\_test\_prediction)

print('Accuracy score of test data : ', test\_data\_accuracy)

ð Accuracy score of test data : 0.8717948717948718

- Đánh giá mô hình trên kết quả thực nghiệm:

Theo mã chương trình trên, đối với mô hình học máy vecto hỗ trợ (SVM), độ chính xác đạt 88% trên dữ liệu train và 87% trên dữ liệu test.

# **V.** **KẾT LUẬN**

Học máy vecto hỗ trợ (SVM) là một kĩ thuật phân lớp khá phổ biến, SVM thể hiện được nhiều ưu điểm trong số đó có việc tính toán hiệu quả trên các tập dữ liệu lớn. Có thể kể thêm một số ưu điểm của phương pháp này như:

- Xử lý trên không gian số chiều cao: SVM là một công cụ tính toán hiệu quả trong không gian chiều cao, trong đó đặc biệt áp dụng cho các bài toán phân loại văn bản và phân tích quan điểm nơi chiều có thể cực kỳ lớn.

- Tiết kiệm bộ nhớ: Do chỉ có một tập hợp con của các điểm được sử dụng trong quá trình huấn luyện và ra quyết định thực tế cho các điểm dữ liệu mới nên chỉ có những điểm cần thiết mới được lưu trữ trong bộ nhớ khi ra quyết định.

- Tính linh hoạt - phân lớp thường là phi tuyến tính. Khả năng áp dụng Kernel mới cho phép linh động giữa các phương pháp tuyến tính và phi tuyến tính từ đó khiến cho hiệu suất phân loại lớn hơn.

SVM là một phương pháp hiệu quả cho bài toán phân lớp dữ liệu. Nó là một công cụ đắc lực cho các bài toán về xử lý ảnh, phân loại văn bản, phân tích quan điểm. Một yếu tố làm nên hiệu quả của SVM đó là việc sử dụng Kernel function khiến cho các phương pháp chuyển không gian trở nên linh hoạt hơn.

# **TÀI LIỆU THAM KHẢO**

<https://www.youtube.com/watch?v=HbyN_ey-JVc&list=PLfFghEzKVmjvuSA67LszN1dZ-Dd_pkus6&index=14&ab_channel=Siddhardhan>

<https://www.kaggle.com/datasets/debasisdotcom/parkinson-disease-detection>

<https://drive.google.com/file/d/17i0c6SmncNuwSgr9W1MRRk3YYdEOP9Gd/view>

<https://www.linkedin.com/pulse/machine-learning-basics-support-vector-machines-amsal-gilani/>

<https://www.researchgate.net/publication/261428113_Support_Vector_Machine_Classification_of_Parkinson's_Disease_Essential_Tremor_and_Healthy_Control_Subjects_Based_on_Upper_Extremity_Motion>

<https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>