**CHƯƠNG I**

**TỔNG QUAN ĐỒ ÁN**

1. **Nội dung chung.**

**\* Tên đề tài:**

Phương pháp Support Vector Machines: Lý thuyết và ứng dụng.

**\* Giảng viên hướng dẫn:**

ThS. Nguyễn Đình Hiển.

**\* Sinh viên thực hiện:**

Nguyễn Trí Hải, Nguyễn Hoàng Nghĩa.

**\* Khoá học:**

Khoá 2011 (Ngày nhập học: Tháng 09/2011).

**\* Thông tin liên lạc của sinh viên:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên** | **MSSV** | **Email** |
| 1 | Nguyễn Trí Hải | 11520094 | [11520094@gm.uit.edu.vn](mailto:11520094@gm.uit.edu.vn) |
| 2 | Nguyễn Hoàng Nghĩa | 11520603 | [11520603@gm.uit.edu.vn](mailto:11520603@gm.uit.edu.vn) |

**\* Chương trình, ứng dụng sử dụng:**

* Chương trình lập trình: Mathworks Matlab R2012b.
* Môi trường lập trình: Matlab.

1. **Phân công thực hiện.**

|  |  |
| --- | --- |
| **Phân công** | **Thực hiện** |
| Tìm kiếm, tổng hợp tài liệu | |
| Cả nhóm | Lên nội dung cần làm cho đề tài, tìm kiếm tài liệu:  - Thời gian thực hiện.  - Nội dung lý thuyết, chương trình, ứng dụng hỗ trợ. |
| Thực hiện đồ án | |
| Nguyễn Trí Hải | - Tìm hiểu, xây dụng nội dung ứng dụng |
| Nguyễn Hoàng Nghĩa | - Tổng hợp, xây dựng nội dung lý thuyết. |
| Cả nhóm | - Viết báo cáo, trình bày slide  - Sửa lỗi |

**CHƯƠNG II.**

**CÁC KHÁI NIỆM CƠ BẢN**

1. **Bài toán phân lớp và phương pháp Support Vector Machines.**

Phân lớp là quá trình “nhóm” các đối tượng “giống” nhau vào “một lớp” dựa trên các đặc trưng dữ liệu của chúng. Khi nghiên cứu một đối tượng, hiện tượng, chúng ta chỉ có thể dựa vào một số hữu hạn các đặc trưng của chúng. Nói cách khác, ta chỉ xem xét biểu diễn của đối tượng, hiện tượng trong một không gian hữu hạn chiều, mỗi chiều ứng với một đặc trưng được lựa chọn.

Bài toán phân lớp có thể được mô tả như sau:

Cho tập mẫu

Tìm ánh xạ , sao cho

Trong đó:

N: Số mẫu

Xi: Mẫu dữ liệu thứ i;

Ci: lớp của mẫu dữ liệu thứ i.

Ánh xạ ở đây có thể hiểu là một mô hình các quy tắc, các luật để xác định từng đối tượng đang xét thuộc về lớp nào dựa trên các đặc trưng của chúng. Trong thực tế, việc xác định các đặc trưng để phân lớp bị ảnh hưởng bởi nhiều yếu tố gây khó khăn cho quá trình phân lớp như: đối tượng có nhiều thuộc tích, xác định các thuộc tính nào là cần thiế­­t, các thuộc tính nào là không cần thiết.

Bài toán phân lớp sử dụng phương pháp Support vector machines (SVM) nhằm mục đích tìm một siêu phẳng có biên cực đại giữa lớp mẫu âm và lớp mẫu dương. Đồng thời cực tiểu hoá các mẫu không phân chia được trong tập huấn luyện. SVM đựa trên cơ sở toán học vững chắc. Tuy nhiên, việc huấn luyện mẫu sử dụng SVM đòi hỏi phải giải bài toán tối ưu nhiều biến. Ban đầu, SVM được phát triển để giải các bài toán phân lớp, về sau do tính ưu việt, nó còn được sử dụng rộng rãi để giải các bài toán hồi quy.

1. **Phân lớp tuyến tính.**

Việc phân lớp nhị phân được thực hiện bằng cách sử dụng hàm giá trị thực

là hàm tuyến tính, tương ứng với đầu ra {-1, 1}, được phát biểu như sau:

Đầu vào sẽ được gán vào lớp có nhãn 1 nếu , trường hợp ngược lại sẽ được gán vào lớp có nhãn -1. Trường hợp là hàm tuyến tính của ta có thể viết như sau:

Trong đó biểu thị tích vô hướng

Về mặt hình học, các phần tử của không gian đầu vào X sẽ được rơi vào một trong hai phần được phân tách bởi siêu phẳng xác định bằng biểu thức:

Trong đó: w là vector pháp tuyến cùa siêu phẳng, giá trị ngưỡng b thay đổi có thể tạo ra các siêu phẳng song song với nhau.

Với mỗi mẫu dữ liệu x chưa xác định sẽ đươc phân chia thành:

1. **Ma trận GRAM.**

Cho tập các vector trong không gian tích vô hướng X, ma trận G kích thước với được gọi là ma trận GRAM.

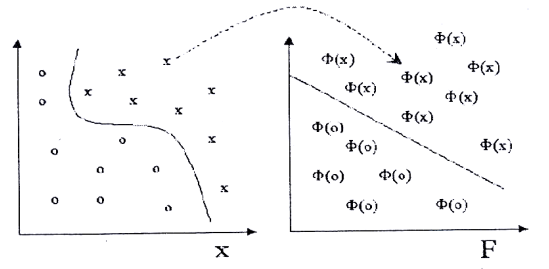
Đặc điểm quan trọng của ma trận GRAM là: các dữ liệu đầu vào cho các chương trình tổng hợp hoặc khái quát hoàn toàn có thể biểu diễn thông qua ma trận GRAM.

1. **Không gian đặc trưng.**

Sự phức tạp của hàm mục tiêu dẫn đến quá trình học phụ thuộc vào cách nó được diễn tả. Khi diễn tả dữ liệu một cách phù hợp, vấn đề học sẽ trở nên dễ dàng. Vì vậy, một việc làm rất phổ biến trong học máy là chuyển đổi dữ liệu từ không gian đầu vào X sang không gian đặc trưng:

Trong đó n là số chiều của đầu vào (số thuộc tính) và N là số chiều của không gian đặc trưng. Dữ liệu sẽ được chuyển vào không gian đặc trưng với N > n.

Không gian đặc trưng ký hiệu là F:



*Hình 2.1 Ánh xạ từ không gian dữ liệu X sang không gian đặc trưng F*

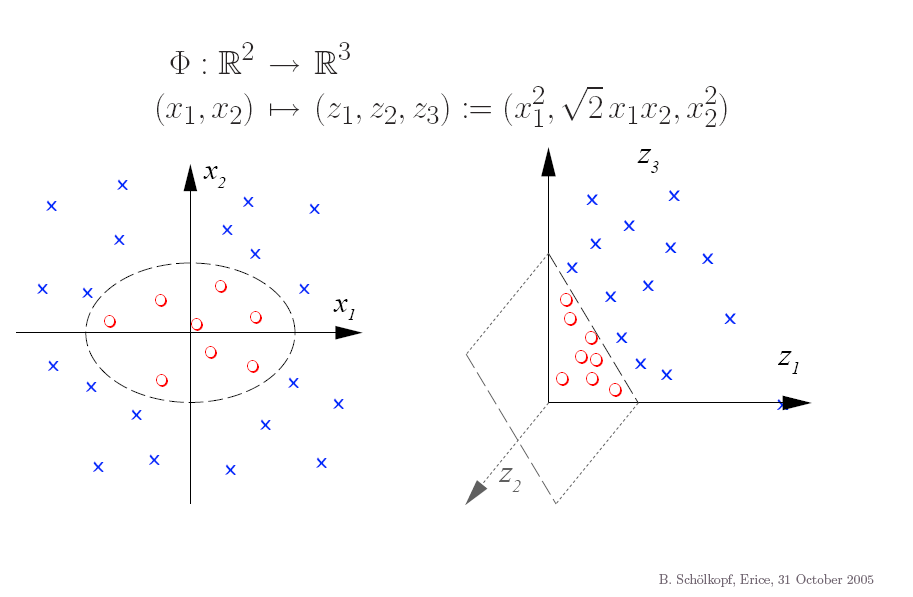
1. **Hàm hạt nhân.**
2. **Khái niệm:**

Một hàm hạt nhân là một hàm K sao cho với mọi , ta có:

Ở đây là tích vô hướng trong không gian đặc trưng.

Ví dụ:

Xét phép biến đổi dữ liệu từ không gian đầu vào vào không gian đặc trưng được cho bởi:



*Hinh 2.2 Ví dụ hàm hạt nhân*

Khi ta chọn x = (x1, x2) và z = (z1, z2) ta có:

Ta thấy khi chọn . Vì vậy ta không cần phải tính ánh xạ .

1. **Hàm hạt nhân trong máy học tuyến tính:**

Máy học không tuyến tính trên không gian đầu vào được xây dựng qua hai bước: trước tiên sử dụng một ánh xạ không tuyến tính để chuyển đổi dữ liệu vào không gian đặc trưng và sau đó sử dụng máy học phân lớp tuyến tính trong không gian đặc trưng.

Máy học tuyến tính trong không gian đặc trưng tương ứng với hàm:

Chúng ta không cần xác định tường minh trọng số w, khi triển khai tiếp bằng cách đưa vào vector , ta có:

Hơn nữa, cũng không cần xây dựng tường minh ánh xạ , nhờ sử dụng hàm hạt nhân nên:

Đặt SV là tập các chỉ số i thoả mãn , ta được:

Các ứng với được gọi là các **vectơ trợ giúp (Support vector)**, hàm phân lớp sẽ được xác định duy nhất qua các vector này.

1. **Một số hàm hạt nhân:**

* , với *d* là tham số do người dùng định nghĩa.
* với do người dùng định nghĩa.

1. **Kết chương.**

Chương 1 đã giới thiệu về bài toán phân lớp, các khái niệm về phân lớp tuyến tính, ma trận GRAM. Và quan trọng về không gian đặc trưng và hàm hạt nhân, khả năng biểu thị của hàm hạt nhân trong không gian đặc trưng.

Đây là các khái niệm cơ bản nhất có liên quan đến bài toán phân lớp và phương pháp SVM áp dụng cho việc giải quyết bài toán phân lớp.

**CHƯƠNG III.**

**PHƯƠNG PHÁP SUPPORT VECTOR MACHINES**

Trong thời đại công nghệ thông tin hiện nay, sự phát triển của công nghệ kéo theo sự gia tăng rất lớn của lưu lượng thông tin lưu trữ và trao đổi. Do đó, yêu cầu về tổ chức lưu trữ và truy cập thông tin sao cho hiệu quả được đặt lên hàng đầu. Hướng giải quyết được đưa ra là tổ chức, tìm kiếm và phân loại thông tin một cách hiệu quả. Bản thân con người trong đời sống cũng tiếp nhận thế giới xung quanh thông qua sự phân loại và tổ chức ghi nhớ tri thức một cách hiệu quả. Phân loại thông qua các lớp và mô tả các lớp giúp cho tri thức được định dạng và lưu trữ trong đó.

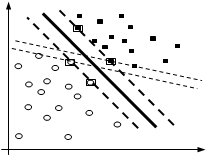
Support Vector Machines (SVM) là một phương pháp phân loại xuất phát từ lý thuyết học thống kê, dựa trên nguyên tắc tối thiểu rủi ro cấu trúc (Structural Risk Minimisation). SVM sẽ cố gắng tìm cách phân loại dữ liệu sao cho có lỗi xảy ra trên tập kiểm tra là nhỏ nhất (Test Error Minimisation). Đây là một phương pháp mới trong lĩnh vực trí tuệ nhân tạo. Vào thời kỳ đầu khi SVM xuất hiện, khả năng tính toán của máy tính còn rất hạn chế, nên phương pháp SVM không được lưu tâm. Tuy nhiên, từ năm 1995 trở lại đây, các thuật toán sử dụng cho SVM phát triển rất nhanh, cùng với khả năng tính toán mạnh mẽ của máy tính, đã có được những ứng dụng rất to lớn.

1. **Ý tưởng.**

Cho trước một tập huấn luyện, được biểu diễn trong không gian vector, trong đó mỗi tài liệu là một điểm, phương pháp này tìm ra một siêu phẳng f quyết định tốt nhất có thể chia các điểm trên không gian này thành hai lớp riêng biệt tương ứng là lớp “” và lớp “”. Chất lượng của siêu phẳng này được quyết định bởi khoảng cách (gọi là biên) của điểm dữ liệu gần nhất của mỗi lớp đến mặt phẳng này. Khi đó, khoảng cách biên càng lớn thì mặt phẳng quyết định càng tốt, đồng thời việc phân loại càng chính xác.

Ý tưởng của nó là ánh xạ (tuyến tính hoặc phi tuyến) dữ liệu vào không gian các vector đặc trưng (space of feature vectors) mà ở đó một siêu phẳng tối ưu được tìm ra để tách dữ liệu thuộc hai lớp khác nhau.

Mục đích của phương pháp SVM là tìm được khoảng cách biên lớn nhất:



*Hình 3.1 Mô tả phương pháp SVM*

Đường tô đậm là siêu phẳng tốt nhất và các điểm được bao bởi hình chữ nhật là những điểm gần siêu phẳng nhất, chúng được gọi là các vector hỗ trợ (support vector). Các đường nét đứt mà các support vector nằm trên đó được gọi là lề (margin).

1. **Cơ sở lý thuyết.**

SVM thực chất là một bài toán tối ưu, mục tiêu của thuật toán này là tìm được một không gian F và siêu phẳng quyết định f trên F sao cho sai số phân loại là thấp nhất.

Cho tập mẫu với , thuộc vào hai lớp nhãn là nhãn lớp tương ứng của các (-1 biểu thị lớp I, 1 biểu thị lớp II).

Ta có, phương trình siêu phẳng chứa vector trong không gian:

Đặt

Như vậy, biểu diễn sự phân lớp của vào hai lớp như đã nêu.

Ta nói nếu thuộc lớp I và nếu thuộc lớp II.

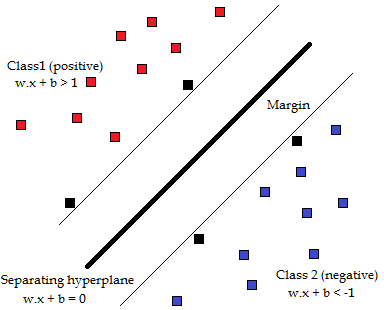
1. **Bài toán phân 2 lớp với SVM.**

Bài toán đặt ra là: Xác định hàm phân lớp để phân lớp các mẫut trong tương lai, nghĩa là với một mẫu dữ liệu mới thì cần phải xác định được phân vào lớp hay lớp .

Ta xét 3 trường hợp, mỗi trường hợp sẽ có 1 bài toán tối ưu, giải được bài toán tối ưu đó ta sẽ tìm được siêu phẳng cần tìm.

**Trường hợp 1:**

Tập D có thể phân chia tuyến tính được mà không có nhiễu (tất cả các điểm được gán nhãn +1 thuộc về phía dương của siêu phảng, tất cả các điểm được gán nhãn -1 thuộc về phía âm của siêu phẳng).



*Hình 3.2 Tập dữ liệu được phân chia tuyến tính*

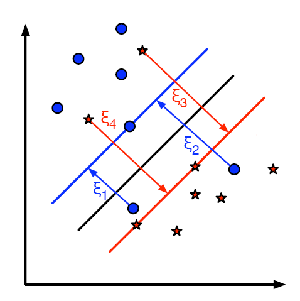
Ta sẽ tìm siêu phẳng tách với là vector trọng số, là hệ số tự do, sao cho:

Đặt

Lúc này ta cần giải bài toán tối ưu:

**Trường hợp 2:**

Tập dữ liệu D có thể phân chia tuyến tính được nhưng có nhiễu. Trong trường hợp này, hầu hết các điểm đều được phân chia đúng bởi siêu phẳng. Tuy nhiên có 1 số điểm bị nhiễu, nghĩa là: Điểm có nhãn dương nhưng lại thuộc phía âm của siêu phẳng, điểm có nhãn âm nhưng lại thuộc phía dương của siêu phẳng.

******

*Hình 3.3 Tập dữ liệu phân chia tuyến tính nhưng có nhiễu*

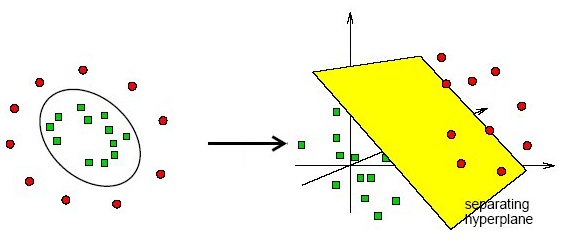
Trong trường hợp này, ta sử dụng 1 biến mềm sao cho:

Bài toán tối ưu trở thành:

Trong đó C là tham số xác định trước, định nghĩa giá trị ràng buộc, C càng lớn thì mức độ vi phạm đối với những lỗi thực nghiệm (là lỗi xảy ra lúc huấn luyện, tính bằng thương số của số phần tử lỗi và tổng số phần tử huấn luyện) càng cao.

**Trường hợp 3:**

Tập dữ liệu D không thể phân chia tuyến tính được, ta sẽ ánh xạ các vector dữ liệu x từ không gian n chiều vào một không gian m chiều (m>n), sao cho trong không gian m chiều, D có thể phân chia tuyến tính được.



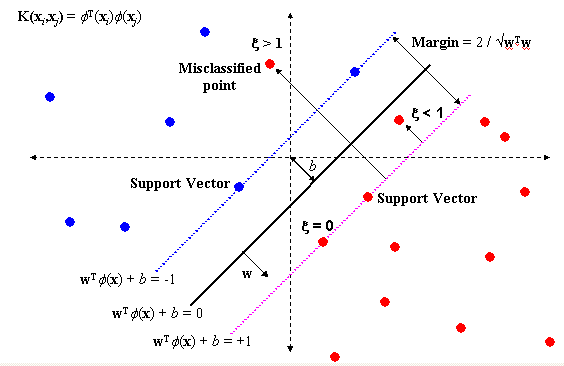
*Hình 3.4 Tập dữ liệu không phân chia tuyến tính.*

Gọi là một ánh xạ phi tuyến từ không gian vào không gian .

Bài toán tối ưu trở thành:

Ví dụ:

Để dễ hiểu hơn chúng ta xét ví dụ mô tả hình học sau: Xét trong không gian 2 chiều (n=2), tập dữ liệu được cho bởi tập các điểm trên mặt phẳng.



*Hình 3.5 Ví dụ biểu diễn tập dữ liệu trên không gian 2 chiều*

Bây giờ ta tiến hành tìm siêu phẳng phân lớp dựa trên phương pháp SVM (1). Ta sẽ tìm 2 siêu phẳng song song (nét đứt trong hình …) sao cho khoảng cách giữa chúng là lớn nhất để có thể phân tách lớp này thành 2 phía (Ta gọi là 2 siêu phẳng phân tách). Siêu phẳng (1) nằm giữa 2 siêu phẳng trên (nét đậm trong hình).

Hình trên cho ta tập dữ liệu có thể phân tách tuyến tính. Bây giờ ta xét trường hợp tập dữ liệu không thể phân tách tuyến tính. Bây giờ ta sẽ xử lý bằng cách ánh xạ tập dữ liệu đã cho vào một không gian mới có số chiều lớn hơn không gian cũ (Gọi là không gian đặc trưng) mà trong không gian này tập dữ liệu có thể phân tách tuyến tính. Trong không gian đặc trưng ta sẽ tiếp tục tìm 2 siêu phẳng phân tách như trường hợp ban đầu.

Các điểm nằm trên 2 siêu phẳng phân tách gọi là các vector hỗ trợ (Support vector). Các điểm này quyết định hàm phân tách dữ liệu. Từ đây, chúng ta có thể thấy phương pháp SVM không phụ thuộc vào các mẫu dữ liệu ban đầu, mà chỉ phụ thuộc vào các suport vector (quyết định 2 siêu phẳng phân tách). Cho dù các điểm khác bị xoá thì thuật toán vẫn cho ra các kết quả tương tự. Đây chính là điểm nổi bật của phương pháp SVM so với các phương pháp khác do các điểm trong tập dữ liệu đều được dùng để tối ưu kết quả.

Trên thực tế người ta sẽ không xác định cụ thể ánh xạ mà dùng hàm hạt nhân như đã trình bày ở mục 5 chương 1, nhằm đẩy nhanh tốc độ tính toán đồng thời đảm bảo dữ liệu sẽ gần như phân tách tuyến tính. Thông thường người ta thường sử dụng các hàm hạt nhân có sẵn.

1. **Bài toán phân đa lớp với SVM.**

Để phân đa lớp thì kỹ thuật SVM sẽ chia không gian dữ liệu thành 2 phần và tiếp tục với không gian đã được phân chia. Khi đó hàm quyết định phân dữ liệu vào lớp thứ i sẽ là:

Những phần tử x là support vector nếu thoả điều kiện:

Giả sử bài toán phân loại k lớp , ta sẽ tiến hành lần phân lớp nhị phân sử dụng phương pháp SVM. Mỗi lớp sẽ tiến hành phân tách với k-1 lớp còn lại để xác định k-1 hàm phân tách (chiến lược “một-đối-một” (one-against-one).

Kỹ thuật phân đa lớp bằng phương pháp SVM hiện vẫn đang được tiếp tục nghiên cứu và phát triển.

1. **Các bước chính của phương pháp SVM.**

* Tiền xử lý dữ liệu: Phương pháp SVM yêu cầu dữ liệu được diễn tả như các vector của các số thực. Như vậy nếu đầu vào chưa phải là số thực thì ta cần phải tìm cách chuyển chúng về dạng số của SVM. Tránh các số quá lớn, thường nên co giãn dữ liệu để chuyển về đoạn [-1,1] hoặc [0,1].
* Chọn hàm hạt nhân: Cần chọn hàm hạt nhân phù hợp tương ứng cho từng bài toán toán cụ thể để đạt được độ chính xác cao trong quá trình phân lớp.
* Thực hiện việc kiểm tra chéo để xác định các tham số cho ứng dụng.
* Sử dụng các tham số cho việc huấn luyện tập mẫu.
* Kiểm thử tập dữ liệu Test.

1. **So sánh và một số cải tiến.**

Một số phương pháp như mạng neuron, fuzzy logic, mạng fuzzy-neuron, …, cũng được sử dụng thành công để giải quyết bài toán phân lớp. Ưu điểm của các phương pháp này là không cần xác định mô hình toán đối của đối tượng (Giải quyết tốt với các hệ thống lớn và phức tạp).

SVM có 2 đặc trưng cơ bản:

* Nó luôn kết hợp với các dữ liệu có ý nghĩa về mặt vật lý, do vậy dễ dàng giải thích được một cách tường minh.
* Cần một tập các mẫu huấn luyện rất nhỏ.

Phương pháp SVM hiện nay được xem là một công cụ mạnh và tinh vi nhất hiện nay cho những bài toán phân lớp phi tuyến. Nó có mộ số biến thể như C – SVC, v – SVC. Cải tiến mới nhất hiện nay của phương pháp SVM đã được công bố là thuật toán NNSRM (Nearest Neighbor and Structural Risk Minimization) là sự kết hợp giữa 2 kỹ thuật SVM và Nearest Neighbor.

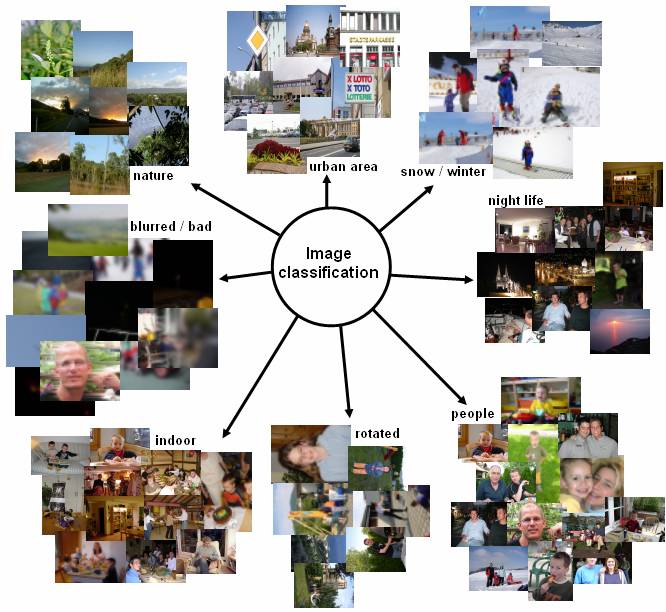
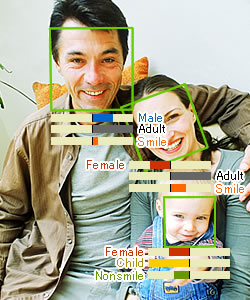
**CHƯƠNG IV.**

**ỨNG DỤNG**

1. **Một số ứng dụng của SVM.**

SVM và các biến thể, cái tiến của nó hiện nay có rất nhiều ứng dụng thiết thực vào giài quyết các vấn đề thực tế. Một số ứng dụng nổi bật của nó là:

* Chuẩn đoán virus máy tính: Giả sử có một tập các đối tượng cần chuẩn đoán trong không gian . Dựa vào các đặc trưng của các , ta sẽ áp dụng phương pháp SVM để phân vào lớp I: Có thể nhiễm virus V hoặc lớp II: Không nhiễm virus V.
* Lọc thư rác (Mail Classification): Với mỗi văn bản, ta sẽ trích, chọn đặc trưng và vector hoá nó thành một vector n chiều có thể biểu diễn dạng dữ liệu SVM được. Sau đó ta tiến hành cho chương trình học thông qua một tập dữ liệu vào dạng với . Ta sẽ xây dựng một siêu phẳng bằng phương pháp SVM trên tập dữ liệu mẫu. Từ đó với mỗi mẫu văn bản mới, ta sẽ vector hoá nó và so sánh dấu của nó so với siêu phẳng tìm được để phân loại.
* Ngoài ra, phương pháp SVM cho bài toán phân đa lớp còn được ứng dụng cho bài toán phân loại ảnh (Image classification) hay phân loại biểu cảm trên khuôn mặt (Facial expression classification).



*Hình 4.1 Phân loại biểu cảm Hình 4.2 Phân loại ảnh*

1. **Mô phỏng phương pháp SVM qua biểu diễn hình học bằng Matlab.**

Các tập tin trong project Matlab:

setup.m : Mô phỏng quá trình phân lớp bằng SVM

data1.mat : Tập dữ liệu mẫu 1

data2.mat : Tập dữ liệu mẫu 2

data3.mat : Tập dữ liệu mẫu 3

svmTrain.m : Hàm đào tạo (training) bằng SVM

svmPredict.m : Hàm dự đoán (prediction) bằng SVM

plotData.m : Biểu đồ dữ liệu 2 chiều

visualizeBoundaryLinear.m : Biểu đồ đường tuyến tính

visualizeBoundary.m : Biểu đồ đường phi tuyến

linearKernel.m : Hàm tuyến tính cho SVM

gaussianKernel.m: Hàm Gaussian cho SVM

dataset3Params.m: Tham số sử dụng cho tập dữ liệu mẫu 3

Chương mô phỏng các tập dữ liệu mẫu lên đồ thị 2 chiều và tìm siêu phẳng với các dữ liệu cho trước và hàm hạt nhân Gaussian (Gaussian Kernel):

1. **Xây dựng chương trình lọc thư rác.**

Các tập tin sử dụng trong project Matlab:

setup\_spam: Quá trình phân loại email spam

spamTrain.mat – Bộ training spam

spamTest.mat – Bộ training test

emailSample.txt - Mẫu email 1

emailSample2.txt – Mẫu email 2

spamSample.txt – Mẫu email spam 1

spamSample2.txt – Mẫu email spam 2

vocab.txt – Danh sách từ vựng

getVocabList.m – Mở danh sách từ vựng

porterStemmer.m – Hàm Stemming

readFile.m – Đọc file

processEmail.m – Tiền xử lý email

emailFeatures.m – Rút trích đặt trưng từ email

**Các bước thực hiện trong bài toán phân loại Email:**

Bước 1. Tiền xử lý email

Bước 2. Trích xuất đặc trưng email

Bước 3. Huấn luyện tuyến tính bằng SVM

Bước 4. Kiểm tra phân lớp Spam Email

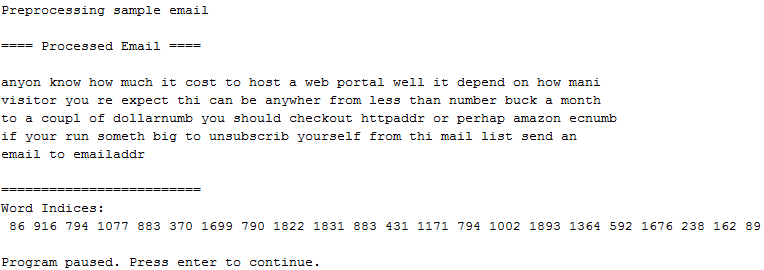
Bước 5. Chọn ra những từ có trong mail có khả năng cao là spam

Bước 6. Đưa email và và kiểm tra

Chi tiết các bước như sau:

**Bước 1. Tiền xử lý email** :

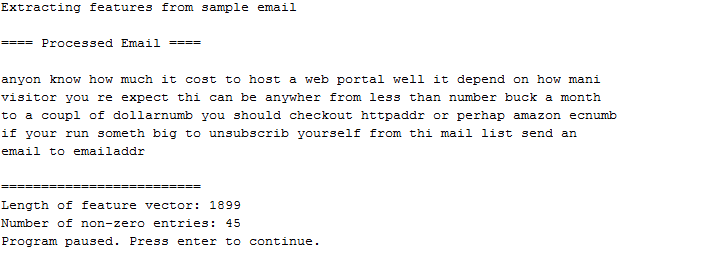
Dựa theo thư viện từ đã có (vocab.txt), xác định các từ trong email có trong thư viện từ hay không.

*Hình 4.3 Tiền xử lý email*

**Bước 2. Trích xuất thông tin email**:

Chương trình xuất ra chiều dài của các vectơ đặc trưng và số các đầu vào khác không.

Một đặc trưng của email có dạng:



*Hình 4.4 Trích xuất thông tin email*

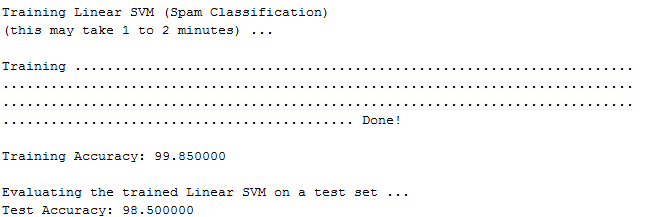
**Kết quả:**

+ Chiều dài của các vectơ đặc trưng : 1899

+ Số các đầu vào khác không: 45

**Bước 3, 4. Quá trình training dựa theo tập dữ liệu spamTrain.mat và kiểm tra quá trình training bằng dữ liệu kiểm tra spamTest.mat**

Quá trình huấn luyện với SVM và chuyển dữ liệu thành các vectơ thông qua hàm svmTrain.m. Sau đó tính độ chính xác của quá trình huấn luyện thông qua hàm svmPredict.m, cũng như tính độ chính xác của quá trình kiểm tra (Dùng dữ liệu spamTest.mat để kiểm tra)



*Hình 4.5 Quá trình học dữ liệu*

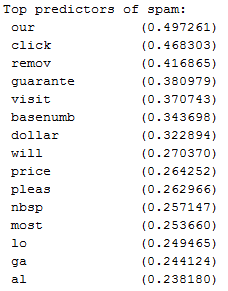
**Kết quả:**

+ Độ chính xác của quá trình huấn luyện : 99.85%

+ Độ chính xác của quá trình kiểm tra: 98.5%

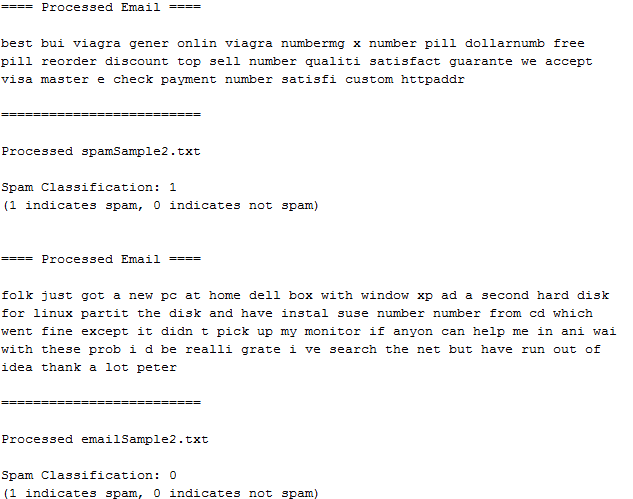
**Bước 5. Chọn ra những từ có trong mail có khả năng cao là spam**

Chương trình đưa ra các từ (word) có thể là spam trong mail.



*Hình 4.6 Các từ mẫu có khả năng là spam*

**Bước 6. Đưa email và và kiểm tra**



*Hình 4.7 Chạy thử chương trình*

Chương trình đưa vào 2 email spamSample2.txt và emailSample2.txt và cho kết quả đúng như đặc tính của mail đó. (Kết quả: 1 là spam, 0 là không spam).

**CHƯƠNG V.**

**TỔNG KẾT**

Phương pháp Support Vector Machines hiện là một trong các phương pháp tinh vi nhất cho bài toán phân lớp phi tuyến. Mở rộng và các ứng dụng của nó đang được tiếp tục nghiên cứu thêm và phát triển làm tăng tính hiệu quả.

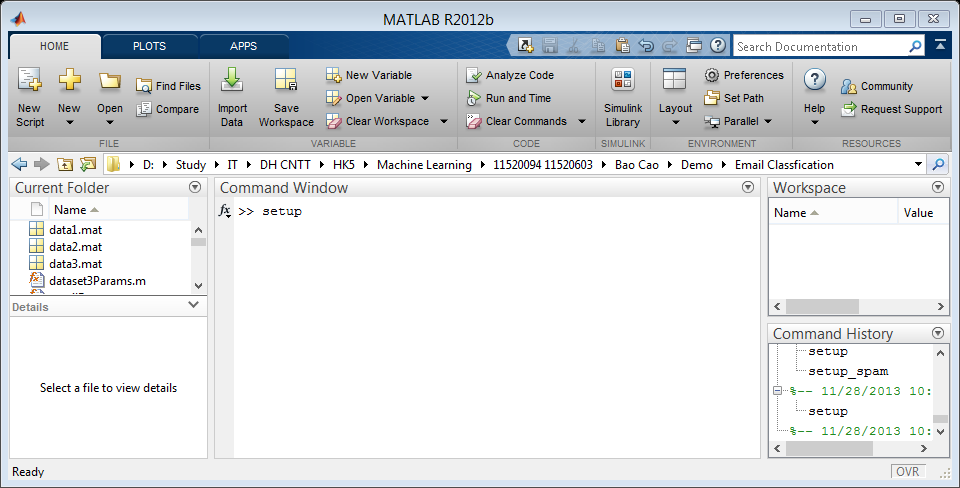
Đồ án nhằm cung cấp những khái niệm cơ bản, nội dung cũng như các bước thực hiện của phương pháp Support Vector Machines đơn giản. Đồng thời cũng cung cấp thêm một số thông tin về các mở rộng của phương pháp này. Phần ứng dụng giới thiệu một các ứng dụng của phương pháp có thể áp dụng và xây dựng lại chương trình lọc thư rác trên Matlab.

**PHỤ LỤC**

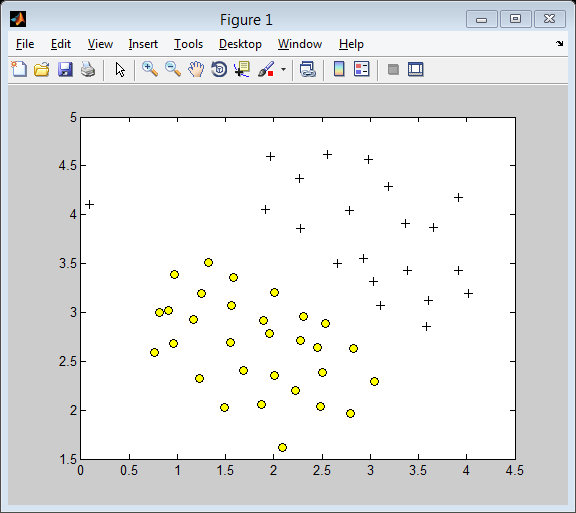
**\* Chạy thử mô phỏng phương pháp SVM qua biểu diễn hình học bằng Matlab.**

**Tập dữ liệu mẫu 1:**

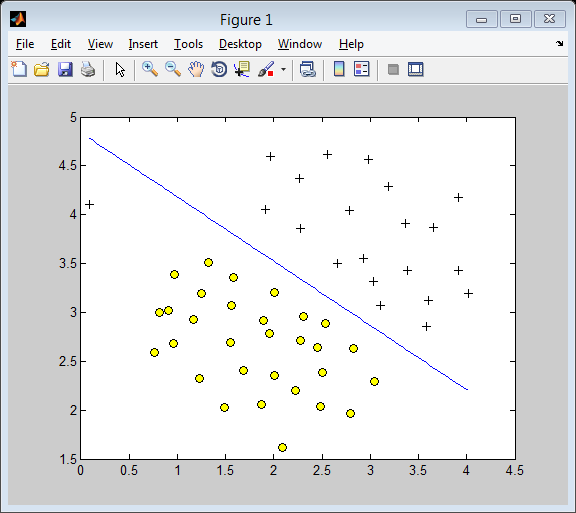
Chạy chương trình bằng cách gõ dòng lệnh : setup



Chương trình sẽ mở dữ liệu mẫu 1 và hiển thị lên sơ đồ



Tiếp tục nhấn Enter, quá trình training sẽ diễn ra và cho ra kết quả:

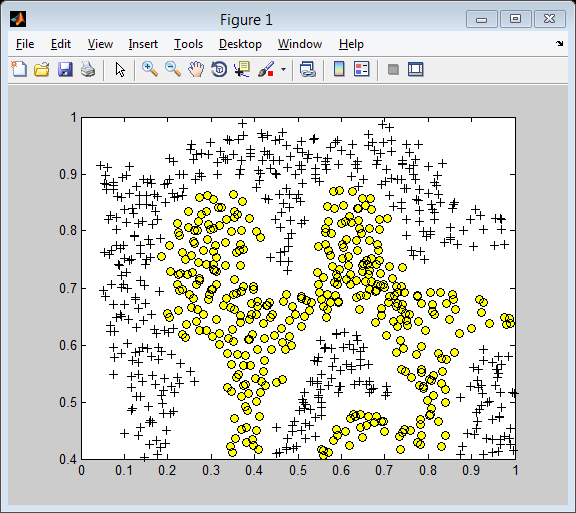


Kết quả đánh giá với hàm hạt nhân Gaussian cho tập dữ liệu mẫu 1:

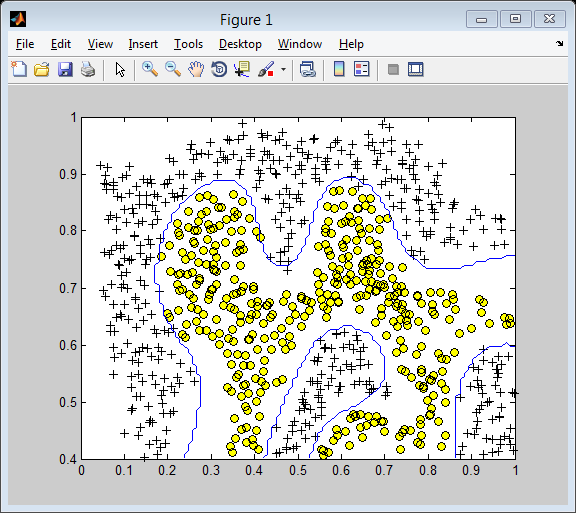
sigma = 0.3247

Hàm hạt nhân Gaussian ở giữa x1 = [1; 2; 1], x2 = [0; 4; -1], sigma = 0.5 : 0.324652

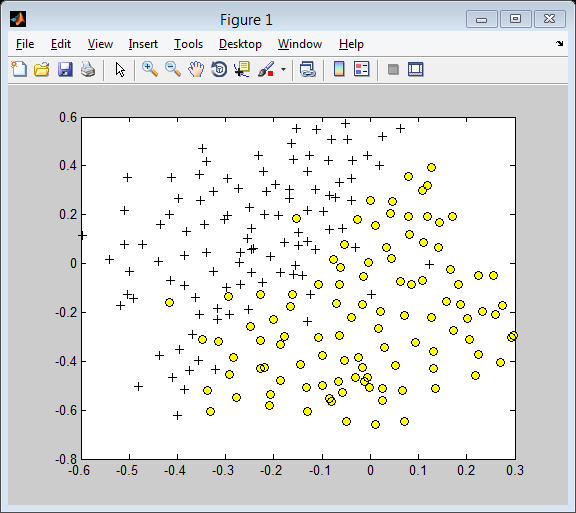
**Tập dữ liệu mẫu 2:**



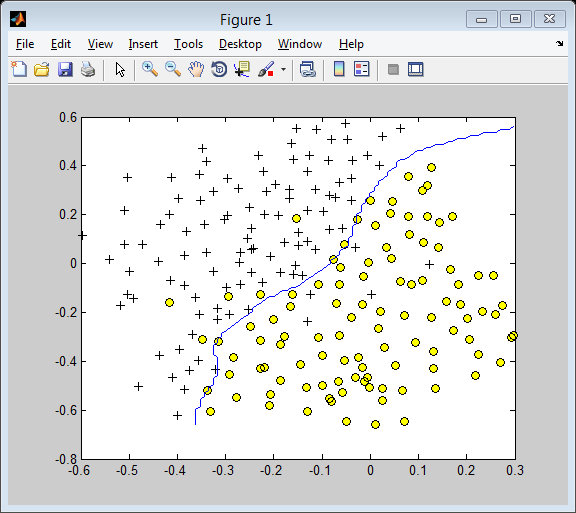
Quá trình training diễn ra với hàm hạt nhân Gaussian, kết quả:



**Tập dữ liệu mẫu 3:**



Quá trình training diễn ra với hàm hạt nhân Gaussian, kết quả:

****

**TÀI LIỆU THAM KHẢO**

[1] Thái Sơn: Luận văn thạc sỹ khoa học: Kỹ thuật Support Vector Machines và ứng dụng. Nghành toán tin ứng dụng: Đại học Bách khoa Hà Nội, 2006.

[2] PGS.TS Vũ Thành Nguyên, Thi Minh Nguyễn: Một số cải tiến của bài toán phân lớp văn bản sử dụng thuật toán SVM và áp dụng trong phân tích tiếng Việt. Đại học Công nghệ thông tin – ĐHQG HCM, 2011.

[3] SVM spam classification: https:\\githug.com\.