BÁO CÁO ĐỒ ÁN MACHINE LEARNING

CQ2016/2

-- ĐỒ ÁN VẤN ĐÁP MÔN MÁY HỌC --
PHÂN LỚP ẢNH CHỮ SỐ VIẾT TAY BẰNG SVM

Tài liệu này mô tả quá trình làm việc nhóm, ghi nhận các kết quả đã chạy thử nghiệm với các siêu tham số trong hai kernel của SVM (là linear kernel và RBF kernel) và đưa ra hàm dự đoán tốt nhất tương ứng trong đồ án.

Thông tin nhóm 22

<1612102> - <Phan Thành Đạt>

<1612406> - <Đặng Phương Nam>

Nhóm có bổ sung 2 câu hỏi tìm hiểu lại:

1. Nhận xét lại kết quả underfitting.

2. scikit-learn cài đặt phân đa lớp là ovo hay ovr?

-> Ở sklearn.svm.SVC có tham số decision_function_shape để quyết định phân đa lớp là 'ovr' hay 'ovo', ban đầu nó mặc định là 'ovr.' Nhưng thực chất khi phân đa lớp thì cái hàm đó sẽ luôn dùng 'ovo' và chúng ta có thể lựa chọn lại 'ovo' hay 'ovr' nếu muốn. (xem ở đây)

Khoa Công nghệ Thông tin
Đại học Khoa học Tự nhiên TP HCM

Tháng 06/2019

1 QUÁ TRÌNH LÀM VIỆC NHÓM

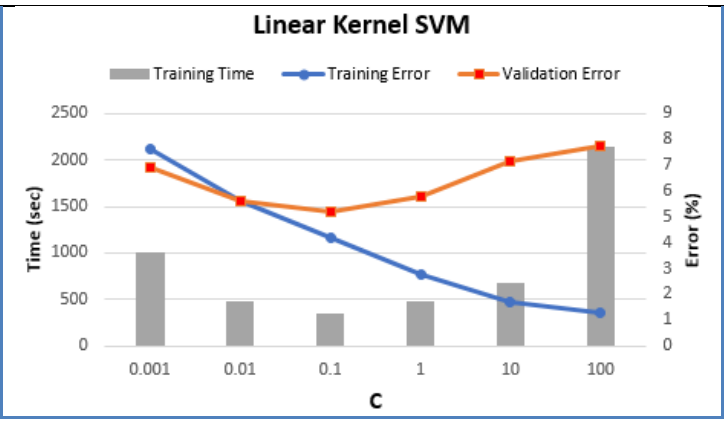
<Phần này ghi nhận lại quá trình làm việc của các thành viên trong nhóm>

Thời gian	Công việc	Thành viên
09/06/2019	Đọc thật kĩ yêu cầu nội dung đồ án, rút ra những điểm chính cần phải thực hiện trong đồ án.	Phan Thành Đạt Đặng Phương Nam
10/06/2019 – 14/06/2019	Xem lại phần lý thuyết SVM đã học và tiếp tục xem tiếp 2 video về Kernel Method và RBF. Thảo luận thắc mắc về nội dung trong video qua Facebook.	Phan Thành Đạt Đặng Phương Nam
15/06/2019 – 17/06/2019	Chạy thuật toán SVM với linear kernel với một vài giá trị C trải rộng để tìm giá trị C có độ lỗi nhỏ nhất. Ghi lại các thông số (độ lỗi trainning, độ lỗi validation và thời gian training), vẽ đồ thị và nhận xét để đưa ra giá trị C tốt nhất.	Đặng Phương Nam
18/06/2019 – 19/06/2019	Tiếp tục chạy thuật toán SVM với linear kernel với giá trị C xung quanh giá trị C tốt nhất đã tìm được. Ghi lại các thông số (độ lỗi trainning, độ lỗi validation và thời gian training), vẽ đồ thị và nhận xét để đưa ra giá trị C tốt nhất.	Phan Thành Đạt
20/06/2019	Nhận xét về kết quả đã chạy. (Linear kernel) Đưa ra hàm dự đoán tốt nhất và sử dụng tập test để đánh giá hàm dự đoán đó.	Đặng Phương Nam Phan Thành Đạt
21/06/2019 – 22/06/2019	Chạy thuật toán SVM với RBF Kernel, chọn cố định một tập giá trị C, thay đổi γ để xem sự ảnh hưởng của γ đến quá trình học *(phần 1)*. Và chọn cố định một tập giá trị γ, thay đổi C để xem sự ảnh hưởng của C đến quá trình học *(phần 2)*.	(1) Phan Thành Đạt (2) Đặng Phương Nam
24/06/2019 – 26/06/2019	Nhận xét về các kết quả đã chạy. (KBF kernel) Đưa ra hàm dự đoán tốt nhất và sử dụng tập test để đánh giá hàm dự đoán đó.	Đặng Phương Nam Phan Thành Đạt
27/06/2019	Thảo luận lại và đưa ra câu trả lời cho các vấn đề được đề cập trong nội dung đồ án	Đặng Phương Nam Phan Thành Đạt

2 GHI NHẬN KẾT QUẢ

2.1 LINEAR KERNEL – SVM

C	Training Error (%)	Validation Error (%)	Training Time (sec)
0.001	7.610	6.910	1002
0.01	5.594	5.630	478
0.1	4.188	5.190	352
1	2.754	5.770	474
10	1.692	7.160	673
100	1.300	7.770	2140



Dựa vào kết quả ta thấy:

- Ban đầu, khi C quá nhỏ (0.001 và 0.01) thì cho phép mức độ sai lệch lớn làm cho margin lớn (để chấp nhận nhiễu nhiều hơn) dẫn đến độ lỗi trong tập training cao và vì chấp nhận nhiều nhiễu sẽ khiến cho khả năng phân lớp không còn chính xác nữa làm cho độ lỗi trên tập validation cũng cao theo.
- Ta tiếp tục tăng C lên thì thấy độ lỗi trên tập training và validation càng giảm, nhưng khi C càng lớn (1, 10 và 100) thì lại dẫn đến trường hợp không cho phép chấp nhận độ lỗi lớn làm cho margin nhỏ lại để giúp cho việc phân lớp trên tập training đạt độ chính xác quả cao nhưng do dữ liệu train có nhiễu, vì thế đem ra chạy cho tập validation thì độ lỗi lại tăng lên.
- Rõ ràng khi C có giá trị dung dung (0.1) thì có vẻ cho độ lỗi trên 2 tập train và validtion chấp nhận được và độ lỗi validation là nhỏ nhất trong bảng trên.

Ta tiếp tục thử dò xung quanh giá trị C = 0.1 thu được ở trên, thì thấy với C = 0.05 có training error cao hơn và C = 0.15 tuy có training error thấp hơn nhưng cả hai đều có validation error cao hơn C = 0.1 nên có thể dừng lại ở đây và chọn C = 0.1 làm giá trị tốt nhất.

C	Training Error (%)	Validation Error (%)	Training Time (sec)
0.05	4.576	5.200	368
0.1	4.188	5.190	352
0.15	3.968	5.330	354

Kết quả hoàn toàn phù hợp với lý thuyết. Nhìn chung với C quá nhỏ hay quá lớn thì thời gian training sẽ lâu hơn so với C có giá trị dung dung.

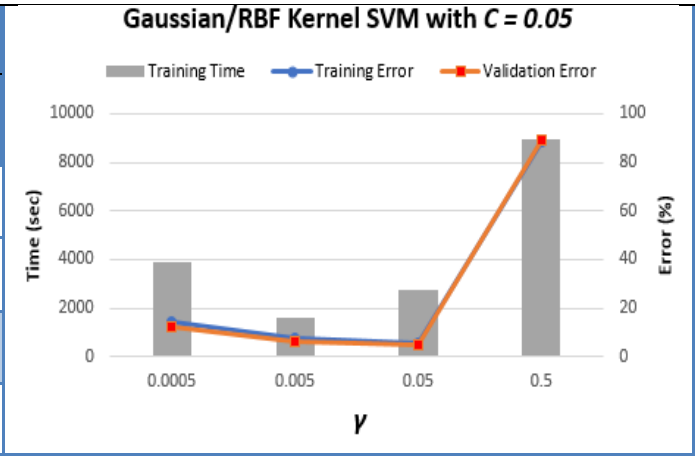
Do đó, nhóm chọn C = 0.1 với thời gian training thấp nhất.

Kết quả, độ lỗi trên tập test: **5.370 %**

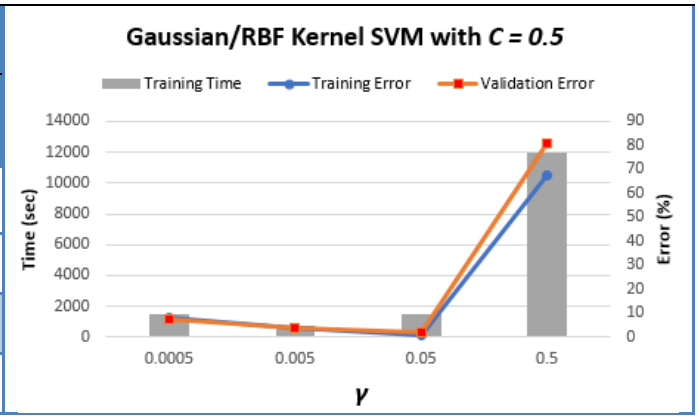
2.2 RBF KERNEL – SVM

❖ **Phần cố định một tập giá trị γ và thay đổi C để xem sự ảnh hưởng của C đến quá trình học**

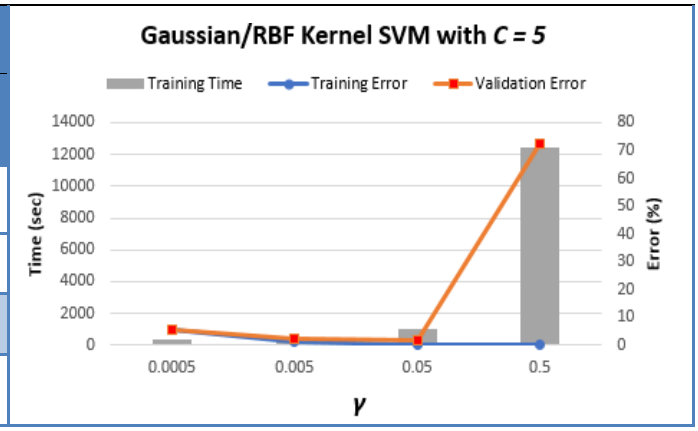
C = 0.05			
γ	**Training Error (%)**	**Validation Error (%)**	**Training Time (sec)**
0.0005	14.152	12.550	3890
0.005	7.452	6.510	1583
0.05	5.270	5.120	2778
0.5	88.644	89.360	8962



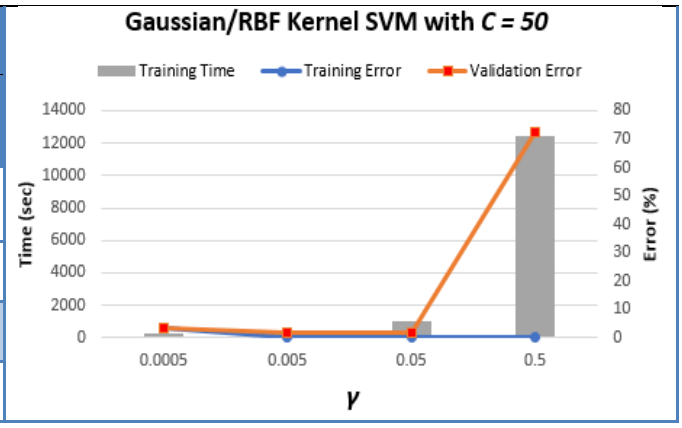
C = 0.5			
γ	**Training Error (%)**	**Validation Error (%)**	**Training Time (sec)**
0.0005	8.310	7.270	1449
0.005	3.920	3.770	739
0.05	0.480	1.980	1515
0.5	67.384	80.900	12007



C = 5			
γ	**Training Error (%)**	**Validation Error (%)**	**Training Time (sec)**
0.0005	5.668	5.470	382
0.005	1.048	2.160	228
0.05	0.000	1.670	1000
0.5	0.000	72.590	12400



C = 50			
γ	**Training Error (%)**	**Validation Error (%)**	**Training Time (sec)**
0.0005	3.336	3.650	229
0.005	0.016	1.950	230
0.05	0.000	1.670	1028
0.5	0.000	72.590	12409

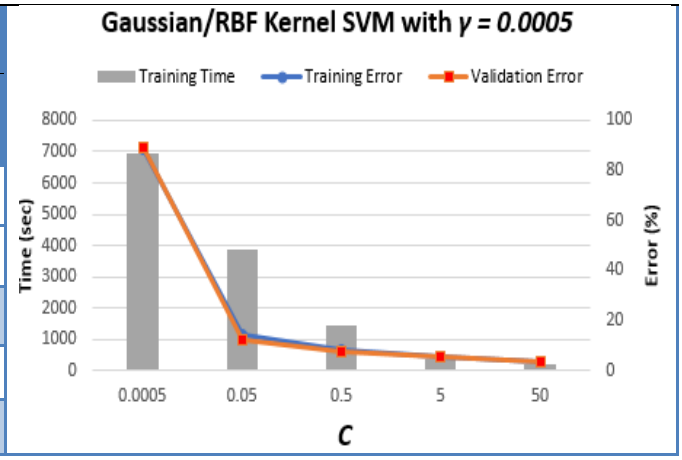


Dựa vào kết quả ta thấy:

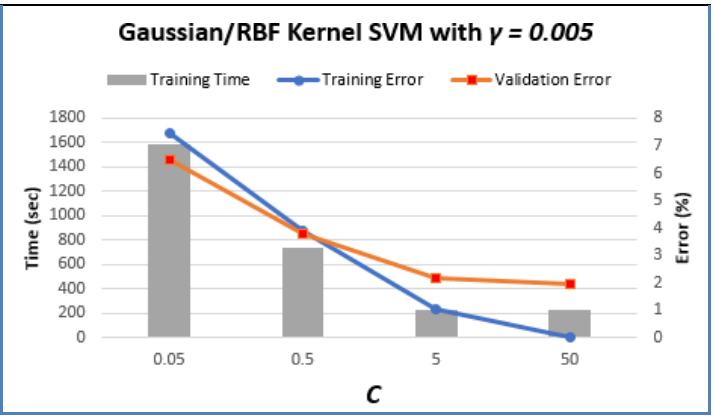
- Khi C quá nhỏ (0.005) thì giá trị độ lỗi thu được của hai tập training và validation với các tập giá trị γ cố định đều khá cao và xê xít nhau. Ta thấy rằng độ lỗi tại γ = 0.5 thì có độ lỗi trên cả 2 tập đều rất cao, dường như bị underfitting, đều này thì không phù hợp với lý thuyết (nhóm vẫn chưa thể giải thích được).
- Ta tăng C lên 0.05 thì thấy rằng độ lỗi ở hai tập training và validation đều đã giảm dần, chứng tỏ tăng C có tác dụng khi làm margin ngắn lại để tăng khả năng phân lớp chính xác, điều này phù hợp với lý thuyết.
- Nhưng khi C có giá trị lớn (5 và 50) tức margin ngày càng hẹp dần thì mặc dù kết quả phân lớp là rất tốt với các giá trị γ = [0.0005, 0.005, 0.05], nhưng lại bị overfitting tại γ = 0.5, điều này phù hợp với lý thuyết.
- Về thời gian training, ta thấy rằng khi C càng tăng thì thì thời gian training càng giảm.

❖ **Phần cố định một tập giá trị C và thay đổi γ để xem sự ảnh hưởng của γ đến quá trình học**

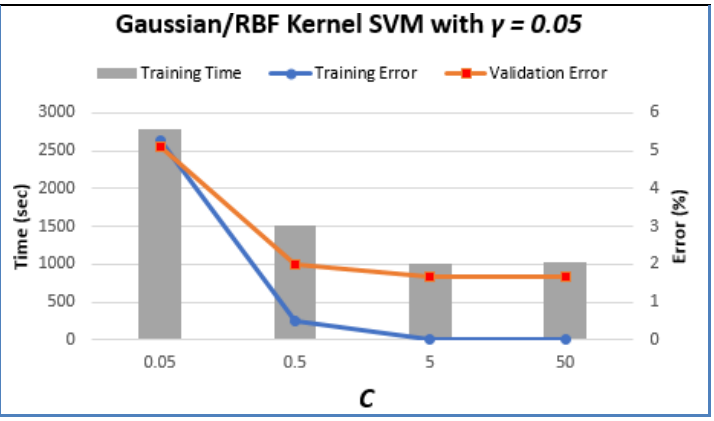
γ = 0.0005			
C	**Training Error (%)**	**Validation Error (%)**	**Training Time (sec)**
0.0005	88.644	89.360	6931
0.05	14.152	12.550	3890
0.5	8.310	7.270	1449
5	5.668	5.470	382
50	3.336	3.650	229



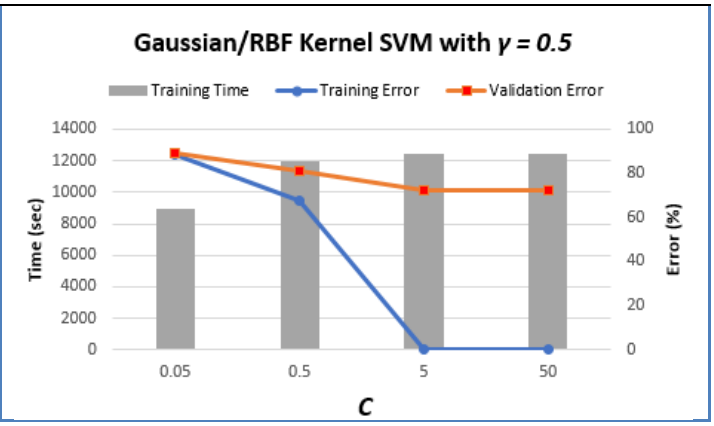
γ = 0.005			
C	Training Error (%)	Validation Error (%)	Training Time (sec)
0.05	7.452	6.510	1583
0.5	3.920	3.770	739
5	1.048	2.160	228
50	0.016	1.950	230



γ = 0.05			
C	Training Error (%)	Validation Error (%)	Training Time (sec)
0.05	5.270	5.120	2778
0.5	0.480	1.980	1515
5	0.000	1.670	1000
50	0.000	1.670	1028



γ = 0.5			
C	Training Error (%)	Validation Error (%)	Training Time (sec)
0.05	88.644	89.360	8962
0.5	67.384	80.900	12007
5	0.000	72.590	12400
50	0.000	72.590	12409



Dựa vào kết quả ta thấy:

- Khi γ quá nhỏ (0.0005) thì độ lỗi thu được trên hai tập training và validation đều khá cao, và cao nhất khi C quá nhỏ (0.0005) với độ lỗi trên cả hai tập đều trên 80% đây là trường hợp underfitting, do C và γ đều quá nhỏ, hiện tượng này hoàn toàn phù hợp với lý thuyết.
- Tăng γ lên (0.005 và 0.05) thì ta thu được kết quả rất tốt, tức γ có giá trị dung dung đang tỏ ra rất hiệu quả với các giá trị C, điều này phù hợp với lý thuyết.

- Đến lúc γ = 0.5 thì xảy ra quá hiện tượng underfitting tại C = [0.05, 0.5] và overfitting tại C = [5, 50], tức giá trị γ = 0.5 càng lớn sẽ khiến cho quá trình học trở nên không thực sự khả thi (overfitting thì phù hợp với lý thuyết, nhưng underfitting thì không phù hợp với lý thuyết).
- Về thời gian training, ta thấy rằng khi γ càng tăng thì thời gian training càng tăng.

Từ các bảng số liệu trên, ta thu được kết quả tốt nhất với **C = 5 và γ = 0.05** là:

training error = 0.000%, validation error = 1.670%, training time = 1000s

Nhưng để kiểm tra có phải là kết quả tốt nhất chưa thì ta tiếp tục thử dò xung quanh C = 5 thì phát hiện ra tại C = 4 đã xảy ra trường hợp Validation Error đã tăng lên. Và thấy rằng lúc C = 2 và C = 3 thu được độ lỗi trên tập validation đều min như nhau, nhưng khi so sánh về thời gian training và độ lỗi training ta thấy C = 3 thật sự tốt hơn C = 2.

C	Training Error (%)	Validation Error (%)	Training Time (sec)
2	4.10^{-5}	1.645	1503
3	2.10^{-5}	1.645	1497
4	0.000	1.670	1502

❖ **Cuối cùng:**

+ C quá lớn hay quá nhỏ đều không tốt.
+ γ quá lớn hay quá nhỏ đều không tốt.
+ C và γ có giá trị dung dung thì tốt.

Mọi thứ đều ổn và phù hợp với lý thuyết, nhưng ở trường hợp γ = 0.5 tương ứng với C = 0.05 (hay C = 0.5) thì xảy ra underfitting (nhóm vẫn chưa hiểu rõ chỗ này). Và so sánh các số liệu ở trên ta thấy trường hợp thu được Validation Error nhỏ nhất là khi C = 3 và γ = 0.05.

Do đó, nhóm chọn C = 3 và γ = 0.05.

Kết quả độ lỗi trên tập test: **1.730 %**