**Báo cáo ngắn về LibSVM**

1. **Dữ liệu**

Định dạng dữ liệu đưa vào để train như sau (ở đây chỉ đề cập đến SVR)

[y] 1:[giá trị 1] 2:[giá trị 2] …. N:[giá trị N]

Với xác nhận rằng y là đầu ra thật sự và vector đầu vào x có N chiều và giá trị của từng thành phần trong x được scale về [-1;1]. Cụ thể như sau:

15 1:-1 2:0.027 3:0.04 4:-0.8 5:-0.6 6:-0.69 7:-0.7 8:-0.7

1. **Vai trò của các lớp chính**
2. SupportClass.cs

Chứa những thành phần hỗ trợ cho quá trình thực thi thuật toán.

* Một biến Random dùng trong quá trình cross validation
* Một lớp Tokenizer hỗ trợ quá trình tách các dòng dữ liệu từ định dạng ở trên.

1. Svm\_node.cs

Chứa class svm\_node là mô tả một node của vector đầu vào. Hiểu đơn giản node là 1 thành phần trong vector đầu vào N chiều, khi đó 1 vector đầu vào sẽ được thể hiện thành N node.

1. Svm\_parameter.cs

Chứa class svm\_parameter bao gồm các thành phần là tham số cho mô hình.

1. Svm\_problem.cs

Chứa class svm\_problem gồm tập các giá trị đầu ra và vector đầu vào, và số lượng sample trong tập đó. Trên thực tế, một đối tượng của lớp này được dùng như 1 tập training trong quá trình cross validation. Khi đó, nếu chia làm 5 fold: 4 train và 1 test, thì đối tượng của lớp này sẽ giữ thông tin của tất cả 4 fold train.

1. Svm\_model.cs

Chứa class svm\_model. Một đối tượng của lớp này sẽ chứa thông tin kết quả của quá trình train. Đại khái như các tham số, support vector, và một vài hệ số khác.

1. Svm.cs

Bao gồm khá nhiều lớp phục vụ cho quá trình chạy thuật toán. Ở đây ta nên để ý đến 3 lớp quan trọng: Cache, Kernel, SVR\_Q.

* Cache

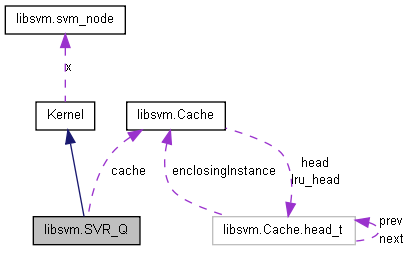
Đặc trưng của lớp này là chứa 1 cấu trúc dữ liệu dạng xâu vòng. Ngoài ra, những phần chi tiết chưa rõ lắm.

* Kernel

Hỗ trợ quá trình tính toán trong kernel.

* SVR\_Q

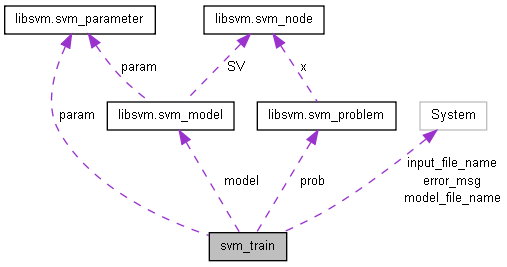
Kế thứa từ Kernel và override các phương thức đặc trưng cho nhân SVR.



*Sơ đồ lớp của 3 lớp liên quan*

1. Svm\_train.cs

Đây là lớp chạy thuật toán để train cho mô hình. Nhận tham số đầu vào dạng command line vào train thông quan cross validation. Sau khi train, kết quả của mô hình sẽ được lưu lại dưới dạng file để phục vụ cho quá trình dự đoán. Đây là sơ đồ lớp liên quan:



1. Svm\_predict.cs

Chức các phương thức cho quá trình dự đoán. Đầu vào là file chứa model đã train và file chứa các sample cần dự đoán, đầu ra là kết quả dự đoán và giá trị các độ đo (MSE, SCC).

* Lưu ý: Ở đây có 2 hàm main, trong svm\_predict.cs và svm\_train.cs. Nên muốn chạy thì chỉ chạy lần lượt bằng cách bao hàm main của cái còn lại.

1. **Các vấn đề khác**

Trên đây là tất cả những vấn đề tổng quan nhất về cách mà mô hình sẽ thực hiện. Đi sâu vào bên trong thì vấn đề cần tìm hiểu là các kỹ thuật mà libsvm đã xử lý trong giai đoạn training và một phần nhỏ trong giai đoạn predict. Do thời gian có hạn nên những chi tiết này chưa được thông suốt và cái cần xem xét kỹ tiếp theo là những phương thức trong svm.cs (phần lớn những kỹ thuật phức tạp nằm trong này).