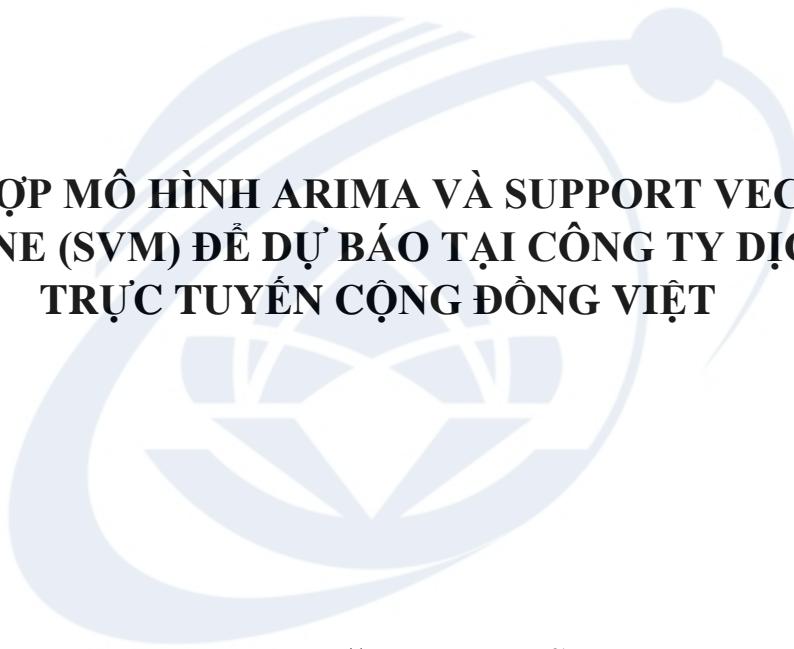


ĐẠI HỌC QUỐC GIA TP HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



Hồ Công Hoài

**KẾT HỢP MÔ HÌNH ARIMA VÀ SUPPORT VECTOR
MACHINE (SVM) ĐỂ DỰ BÁO TẠI CÔNG TY DỊCH VỤ
TRỰC TUYẾN CỘNG ĐỒNG VIỆT**



LUẬN VĂN THẠC SĨ
NGÀNH KHOA HỌC MÁY TÍNH

Mã số: 60.48.01.01

TP HỒ CHÍ MINH – NĂM 2017

LỜI CAM ĐOAN

Tôi xin cam đoan:

1. Những nội dung trong luận văn này là do tôi thực hiện dưới sự hướng dẫn trực tiếp của Thầy PGS. TS Nguyễn Đình Thuân.
2. Mọi tham khảo trong luận văn đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian công bố.

Mọi sao chép không hợp lệ, vi phạm quy chế đào tạo tôi xin chịu hoàn toàn trách nhiệm.

Tp. Hồ Chí Minh, ngày 19 tháng 01 năm 2017

Học viên

Hồ Công Hoài

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến Quý Thầy Cô, cán bộ công nhân viên của Trường Đại học Công nghệ Thông tin, Đại học Quốc gia Tp HCM đã chỉ dạy những kiến thức và tạo mọi điều kiện tốt nhất trong quá trình học tập tại trường. Đặc biệt em xin gửi lời cảm ơn đến Thầy PGS. TS Nguyễn Đình Thuân, cảm ơn Thầy đã hướng dẫn em thực hiện đề tài luận văn này. Chúc Thầy luôn dồi dào sức khỏe để tiếp tục nghiên cứu khoa học và giảng dạy.

Em cũng xin gửi lời cảm ơn đến Công ty Dịch vụ Trực tuyến Cộng Đồng Việt đã tạo điều kiện để em hoàn thành đề tài luận văn này. Đặc biệt em xin gửi lời cảm ơn đến anh Nguyễn Quốc Hương, Trưởng phòng Tích hợp hệ thống, cảm ơn anh đã hỗ trợ và tạo điều kiện để em thực hiện tốt đề tài.

Cuối cùng em xin gửi lời cảm ơn đến Cha Mẹ, gia đình, người thân, bạn bè và đồng nghiệp đã quan tâm, ủng hộ trong suốt quá trình học tập cao học.

Học viên

Hồ Công Hoài

MỤC LỤC

LỜI CAM ĐOAN	1
LỜI CẢM ƠN	2
MỤC LỤC.....	3
DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT	5
DANH MỤC CÁC BẢNG.....	7
DANH MỤC CÁC HÌNH VẼ VÀ ĐỒ THỊ	8
MỞ ĐẦU.....	10
Chương 1. TỔNG QUAN	12
1.1 Chuỗi thời gian và dự báo dữ liệu chuỗi thời gian	12
1.1.1 Chuỗi thời gian	12
1.1.2 Dự báo dữ liệu chuỗi thời gian.....	14
1.2 Tình hình dự báo dữ liệu chuỗi thời gian	15
1.3 Những vấn đề còn tồn tại.....	16
1.4 Mục tiêu, nội dung, phương pháp nghiên cứu.....	16
Chương 2. PHƯƠNG PHÁP DỰ BÁO DỮ LIỆU CHUỖI THỜI GIAN.....	19
2.1 Phương pháp xác suất – thống kê	19
2.1.1 Mô hình hồi quy	19
2.1.2 Mô hình trung bình động.....	20
2.1.3 Mô hình ARMA	21
2.2 Phương pháp máy học	22
2.2.1 Phương pháp mạng neural.....	22
2.2.2 Phương pháp thuật giải di truyền	25
2.3 Phương pháp logic mờ	26
2.3.1 Phương pháp chuỗi thời gian mờ	26
2.4 Phương pháp kết hợp	29
2.4.1 Kết hợp ARIMA và mạng neural	29
2.4.2 Mô hình ARIMA mờ.....	30
Chương 3. MÔ HÌNH KẾT HỢP ARIMA VÀ SUPPORT VECTOR MACHINE ..	33
3.1 Mô hình ARIMA	33
3.1.1 Tính dừng của chuỗi thời gian.....	33
3.1.2 Tính mùa của chuỗi thời gian.....	35

3.1.3 Hàm tự tương quan và hàm tự tương quan riêng phần	36
3.1.4 Giới thiệu mô hình.....	38
3.1.6 Ước lượng các tham số.....	43
3.1.7 Kiểm định mô hình.....	43
3.1.8 Dự báo	44
3.2 Support Vector Machine.....	45
3.2.1 Giới thiệu.....	45
3.2.2 Độ rộng của margin.....	47
3.2.4 Phương pháp Lagrange multipliers	57
3.2.5 Soft Margin và Kernel.....	62
3.2.6 Support Vector Machine trong dự báo chuỗi thời gian.....	65
3.3 Mô hình kết hợp ARIMA và Support Vector Machine	70
3.3.1 Giới thiệu.....	70
3.3.2 Nội dung	71
3.3.3 Một số kết quả tham khảo và đánh giá	72
Chương 4. DỰ BÁO TẠI CÔNG TY DỊCH VỤ TRỰC TUYẾN CỘNG ĐỒNG VIỆT	75
4.1 Giới thiệu về công ty và bài toán dự báo	75
4.2 Chuẩn bị và tiền xử lý dữ liệu	77
4.3 Dự báo.....	78
4.3.1 Dự báo thành phần tuyến tính bằng mô hình ARIMA	79
4.3.2 Dự báo thành phần phi tuyến tính bằng phương pháp SVM.....	84
4.3.3 Kết hợp các kết quả dự báo	88
4.4 Kết quả dự báo và đánh giá	88
4.4.1 Giới thiệu các độ đo	88
4.4.2 Kết quả dự báo và đánh giá	89
Chương 5. KẾT LUẬN VÀ KHUYẾN NGHỊ	93
5.1 Kết luận.....	93
5.2 Khuyến nghị.....	94
TÀI LIỆU THAM KHẢO.....	95
PHỤ LỤC	98

DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

ACF	Auto Correlation Function
AIC	Akaike Info Criterion
AR	Auto Regression
ARIMA	Auto Regression Integrated Move Average
ARMA	Auto Regression Move Average
ANN	Artificial Neural Network
BIC	Bayesian Information Criterion
BJ	Box – Jenkins
FARIMA	Fuzzy Auto Regression Integrated Move Average
GA	Genetic Algorithm
IID	Indentically Independently Distributed
KKT	Karush-Kuhn-Tucker
RMSE	Root Mean Square Error
MA	Moving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MSE	Mean Square Error
PACF	Partial Auto Correlation Function
SAC	Sample Auto Correlation
SARIMA	Seasonal Auto Regression Integrated Move Average
SANN	Seasonal Artificial Neural Networks

SEE	Standard Error of Estimate
SVR	Support Vector Regression
SVM	Support Vector Machine
QC	Quadratic Programming



DANH MỤC CÁC BẢNG

Bảng 3.1. Các dạng lý thuyết của ACF và PACF	42
Bảng 3.2. So sánh kết quả dự báo giá cổ phiếu Công ty Eastman Kodak	72
Bảng 3.3. So sánh kết quả dự báo sản lượng điện cung cấp của Công ty Heilongjiang of China từ 12/04/1999 đến 31/05/1999	73
Bảng 3.4. So sánh kết quả dự báo sản lượng xuất khẩu hoa lan của Thái Lan từ 01/2007 đến 03/2011	73
Bảng 3.5. So sánh kết quả dự báo sản lượng xuất khẩu thịt heo của Thái Lan từ 01/2007 đến 03/2011	74
Bảng 4.1. Các giá trị tiêu chuẩn BIC, AIC và ước lượng sai số chuẩn SEE của các mô hình ARIMA	83
Bảng 4.2. Kết quả dự báo của các mô hình ARIMA	84
Bảng 4.3. Kết quả dự báo của các mô hình SVM	87
Bảng 4.4. Kết quả dự báo của các mô hình.....	92

DANH MỤC CÁC HÌNH VẼ VÀ ĐỒ THỊ

Hình 1.1. Các thành phần chính của chuỗi thời gian	13
Hình 2.1. Mạng neural truyền thẳng 3 lớp	23
Hình 2.2. Logic mờ	27
Hình 3.1. Chuỗi thời gian không dùng	35
Hình 3.2. Chuỗi thời gian dùng.....	35
Hình 3.3. Sơ đồ mô phỏng phương pháp Box-Jenkins	40
Hình 3.4. Đồ thị hàm tự tương quan và hàm tự tương quan riêng phần	42
Hình 3.5. Bài toán phân lớp	46
Hình 3.6. Đường thẳng phân lớp.....	46
Hình 3.7. Khoảng cách trong phân lớp	47
Hình 3.8. Các đường thẳng phân lớp	47
Hình 3.9. Ví dụ về tính độ rộng của margin	48
Hình 3.10. Siêu phẳng tối ưu	51
Hình 3.11. Hai biên của margin	52
Hình 3.12. Khoảng cách giữa hai siêu phẳng	53
Hình 3.13.....	54, 55
Hình 3.14.....	56
Hình 3.15. Ví dụ về Soft Margin	62
Hình 3.16. Ví dụ về Kernel	63
Hình 3.17. SVR trong ước lượng hồi quy	66
Hình 3.18. SVR trong ước lượng hồi quy	67

Hình 3.19. Biến đổi dữ liệu từ không tuyến tính thành tuyến tính	69
Hình 3.20. Biểu đồ so sánh kết quả dự báo giá cổ phiếu Công ty Eastman Kodak....	73
.....	73
Hình 4.1. Quy trình khai thác dữ liệu.....	77
Hình 4.2. Biểu đồ số lượng giao dịch theo ngày từ 01/07/2014 đến 15/01/2015 ...	78
Hình 4.3. Đồ thị hàm PACF.....	80
Hình 4.4. Đồ thị hàm ACF	81
Hình 4.5. Kết quả dự báo của mô hình ARIMA(21, 0, 19) bằng phần mềm thống kê R	84
Hình 4.6. Kết quả dự báo thành phần phi tuyến tính của chuỗi thời gian	85
Hình 4.7. Kết quả khảo sát giá trị epsilon trong khoảng từ 0 đến 1 với độ rộng 0.1 ..	86
.....	86
Hình 4.8. Kết quả khảo sát giá trị epsilon trong khoảng từ 0 đến 0.2 với độ rộng 0.01	87
.....	87
Hình 4.9. Kết quả dự báo thành phần phi tuyến tính của chuỗi thời gian bằng phương pháp SVM	88
Hình 4.10. Kết quả dự báo của mô hình tự hồi quy	90
Hình 4.11. Kết quả dự báo của mô hình ARIMA	90
Hình 4.12. Kết quả dự báo của mô hình kết hợp ARIMA và mạng neural	91
Hình 4.13. Kết quả dự báo của mô hình kết hợp ARIMA và thuật giải di truyền..	91
Hình 4.14. Kết quả dự báo của mô hình kết hợp ARIMA và Support Vector Machine	92
.....	92

MỞ ĐẦU



Chuỗi thời gian là một dạng dữ liệu đặc biệt chứa nhiều thông tin quan trọng và hữu ích. Vì vậy mà khai thác dữ liệu chuỗi thời gian đã trở thành một trong những hướng nghiên cứu quan trọng trong lĩnh vực khai thác dữ liệu. Trong số những bài toán về khai thác dữ liệu dựa trên chuỗi thời gian thì bài toán dự báo chuỗi thời gian đã và đang được nhiều nhà khoa học quan tâm nghiên cứu.

Bên cạnh hướng tiếp cận tìm kiếm các phương pháp khai thác dữ liệu mới, cũng như hướng nghiên cứu cải tiến các phương pháp khai thác dữ liệu hiện tại, trong những năm gần đây nhiều nhà khoa học cũng bắt đầu nghiên cứu các phương pháp khai thác dữ liệu dựa trên sự kết hợp của hai hay nhiều phương pháp khai thác dữ liệu đã có. Sự kết hợp này bước đầu đã mang lại những kết quả tích cực khi các phương pháp khai thác dữ liệu kết hợp đã phát huy được phần nào những ưu điểm cũng như khắc phục được một số hạn chế của từng phương pháp khai thác dữ liệu đơn lẻ.

Nhằm mục đích tìm hiểu về hướng tiếp cận mới này trong lĩnh vực khai thác dữ liệu, cũng như khả năng ứng dụng của nó vào thực tế, luận văn xin trình bày về phương pháp dự báo dữ liệu chuỗi thời gian kết hợp giữa mô hình Auto Regression Integrated Move Average (ARIMA) và Support Vector Machine (SVM), cùng ứng dụng mô hình kết hợp này vào dự báo tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt.

Đối tượng nghiên cứu của đề tài tập trung vào các mô hình dự báo dữ liệu chuỗi thời gian, đặc biệt là các mô hình ARIMA, thuật giải SVM và phương pháp kết hợp mô hình ARIMA và SVM trong dự báo dữ liệu chuỗi thời gian. Bên cạnh đó đề tài còn trình bày kết quả áp dụng các mô hình dự báo dữ liệu chuỗi thời gian vào trong thực tế dựa trên bộ dữ liệu được thu thập tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt.

Phạm vi nghiên cứu của đề tài giới hạn trong việc tìm hiểu và ứng dụng các mô hình dự báo dữ liệu chuỗi thời gian như mô hình hồi quy, mô hình ARIMA, thuật giải SVM và mô hình kết hợp ARIMA và SVM.

Tuy phạm vi nghiên cứu của đề tài giới hạn trong việc tìm hiểu và ứng dụng các mô hình dự báo dữ liệu chuỗi thời gian nhưng đề tài cũng đã mang lại một số ý nghĩa về khoa học và thực tiễn. Về khoa học, kết quả thực nghiệm của đề tài cung cấp thêm tính đúng đắn của hướng tiếp cận kết hợp các mô hình dự báo dữ liệu chuỗi thời gian nói chung và mô hình dự báo dữ liệu chuỗi thời gian kết hợp ARIMA và SVM nói riêng. Về thực tiễn, kết quả dự báo của mô hình kết hợp ARIMA và SVM giúp ích cho Công ty Dịch vụ Trực tuyến Cộng Đồng Việt trong việc dự báo về số lượng giao dịch, số lượng khách hàng đến thanh toán theo từng ngày từ đó có kế hoạch bố trí nhân sự sao cho phù hợp hoặc có thể tham khảo kết quả dự báo của mô hình để có các chiến lược kinh doanh và marketing hiệu quả vào từng thời điểm.

Luận văn được trình bày thành 5 chương:

Chương 1. Tổng quan: Giới thiệu về chuỗi thời gian và dự báo dữ liệu chuỗi thời gian. Trình bày về tình hình nghiên cứu trong và ngoài nước, xác định những vấn đề còn tồn tại trong dự báo dữ liệu chuỗi thời gian. Xác định mục tiêu, nội dung và phương pháp nghiên cứu của đề tài.

Chương 2. Phương pháp dự báo dữ liệu chuỗi thời gian: Giới thiệu về các phương pháp dự báo dữ liệu chuỗi thời gian.

Chương 3: Mô hình kết hợp ARIMA và Support Vector Machine: Giới thiệu về mô hình kết hợp ARIMA và Support Vector Machine trong dự báo dữ liệu chuỗi thời gian.

Chương 4: Dự báo tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt: Giới thiệu về vấn đề cần dự báo và ứng dụng mô hình kết hợp ARIMA và Support Vector Machine vào dự báo tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt.

Chương 5: Kết luận và khuyến nghị: Đánh giá về các kết quả đạt được và hướng phát triển tiếp theo của đề tài.

Chương 1. TỔNG QUAN

Trong chương này sẽ trình bày các khái niệm, tính chất cơ bản của chuỗi thời gian, tổng quan về các phương pháp dự báo dữ liệu chuỗi thời gian. Ngoài ra chương này còn trình bày về những khó khăn, thách thức còn tồn tại trong lĩnh vực dự báo dữ liệu chuỗi thời gian.

1.1 Chuỗi thời gian và dự báo dữ liệu chuỗi thời gian

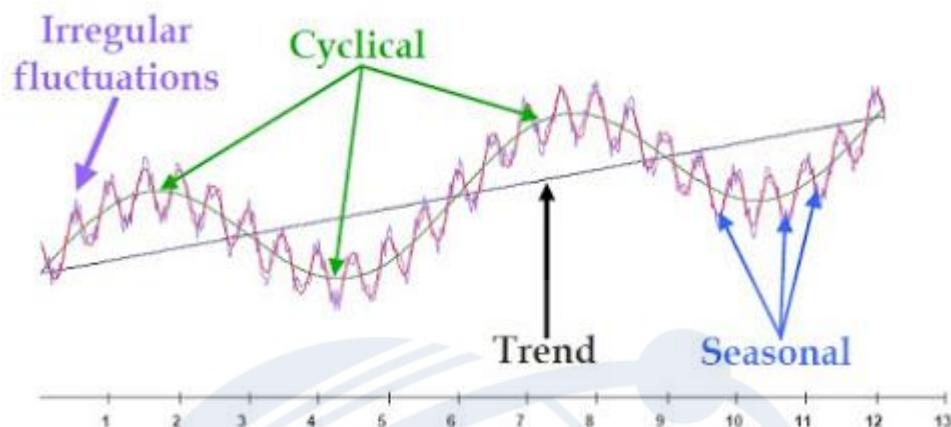
1.1.1 Chuỗi thời gian

Chuỗi thời gian (time series) là một tập hợp các điểm dữ liệu (data points) hay các điểm quan sát (observations) được thu thập và sắp xếp theo thứ tự thời gian. Trong Toán học chuỗi thời gian được định nghĩa là một tập các vector $z(t)$, $t = 0, 1, 2, \dots$ với t là các thời điểm thu thập dữ liệu. Biến $z(t)$ được xem như là một biến ngẫu nhiên [4].

Chuỗi thời gian được gọi là chuỗi thời gian đơn biến nếu trong mỗi điểm dữ liệu chỉ bao gồm một biến duy nhất. Ngược lại, nếu trong mỗi điểm dữ liệu bao gồm nhiều hơn một biến thì chuỗi thời gian đó được gọi là chuỗi thời gian đa biến. Ví dụ chuỗi thời gian là giá đóng cửa của một loại cổ phiếu theo từng ngày là chuỗi thời gian đơn biến, còn chuỗi thời gian là giá mở cửa, giá đóng cửa, giá cao nhất, giá thấp nhất theo từng ngày là chuỗi thời gian đa biến.

Chuỗi thời gian có thể là liên tục hoặc rời rạc. Chuỗi thời gian liên tục là chuỗi thời gian mà các điểm dữ liệu được thu thập một cách liên tục theo thời gian, ví dụ như chuỗi thời gian là nhiệt độ của một khu vực hay lưu lượng nước của một dòng sông. Chuỗi thời gian rời rạc là chuỗi thời gian mà các điểm dữ liệu được thu thập tại các thời điểm rời rạc, ví dụ số lượng sản phẩm được bán ra theo tuần hay tỉ giá quy đổi của hai loại tiền tệ theo ngày. Chuỗi thời gian rời rạc thường được thu thập theo từng khoảng thời gian như từng ngày, từng tuần, từng tháng, từng quý hoặc từng năm.

Chuỗi thời gian thường chịu ảnh hưởng hoặc bị tác động từ 4 yếu tố hay thành phần chính là: xu hướng (trend), chu kỳ (cyclical), mùa (seasonal) và khác thường (irregular)[4].



Hình 1.1. Các thành phần chính của chuỗi thời gian

Nguồn: *Các thành phần của chuỗi thời gian* [22]

- Sự tăng, giảm hoặc không thay đổi của một chuỗi thời gian trong một thời gian dài được gọi là xu hướng. Do đó xu hướng là yếu tố chỉ ra sự vận động lâu dài của một chuỗi thời gian. Ví dụ trong khoảng 15 năm trở lại đây giá xăng dầu trong nước nhiều lần thay đổi, có lúc tăng lúc giảm tuy nhiên về xu hướng giá xăng vẫn tăng.
- Chu kỳ là sự thay đổi của chuỗi thời gian theo một khoảng thời gian trung bình và sự thay đổi đó được lặp lại sau một khoảng thời gian. Thường thì khoảng thời gian của một chu kỳ lớn hơn 2 năm. Ví dụ như số lượng tivi bán ra tăng mạnh vào mỗi mùa Euro hoặc World Cup, và Euro hoặc World Cup diễn ra mỗi 4 năm một lần.
- Mùa là sự thay đổi của chuỗi thời gian theo các mùa trong năm. Có nhiều yếu tố gây ra sự thay đổi chuỗi thời gian theo mùa như yếu tố thời tiết, thói quen truyền thống,... Ví dụ máy lạnh, tủ lạnh thường được tiêu thụ nhiều vào mùa hè và giảm vào mùa đông.

Chương 1. Tổng quan

- Khác thường là yếu tố không thể dự đoán được của chuỗi thời gian. Nó không diễn ra theo lề thường và không lặp lại theo một khuôn mẫu nào cả. Ví dụ như thiên tai, chiến tranh,...

Dựa trên sự tác động của 4 thành phần trên mà có hai loại mô hình được sử dụng cho chuỗi thời gian, đó là mô hình nhân (Multiplicative model) và mô hình cộng (Additive model).

$$- \text{Mô hình nhân: } Y(t) = T(t) * S(t) * C(t) * I(t) \quad (1.1)$$

$$- \text{Mô hình cộng: } Y(t) = T(t) + S(t) + C(t) + I(t) \quad (1.2)$$

Với $Y(t)$ là các điểm dữ liệu, $T(t)$, $S(t)$, $C(t)$ và $I(t)$ lần lượt là các thành phần xu hướng, mùa, chu kỳ, khác thường của chuỗi thời gian.

Mô hình nhân dựa trên giả thiết các thành phần của chuỗi thời gian không nhất thiết độc lập với nhau, chúng có thể tác động lẫn nhau. Ngược lại mô hình cộng cho rằng các thành phần của chuỗi thời gian hoàn toàn độc lập với nhau [4].

1.1.2 Dự báo dữ liệu chuỗi thời gian

Phân tích chuỗi thời gian (time series analysis) là các thao tác dùng để mô hình hóa dữ liệu chuỗi thời gian, ước lượng các tham số của mô hình dựa trên những dữ liệu chuỗi thời gian trong quá khứ, từ đó đưa ra các dự báo về các giá trị của chuỗi thời gian trong tương lai.

Trong dự báo dữ liệu chuỗi thời gian, những giá trị trong quá khứ được thu thập và phân tích để tìm ra các mô hình phù hợp nhằm mô tả chuỗi thời gian. Giá trị tương lai của chuỗi thời gian được dự báo từ các mô hình đó. Do đó, dữ liệu trong quá khứ ảnh hưởng rất lớn đến quá trình xây dựng mô hình và cải thiện kết quả dự báo của mô hình.

Dự báo dữ liệu chuỗi thời gian được ứng dụng rộng rãi trong nhiều lĩnh vực, kết quả dự báo dữ liệu chuỗi thời gian là cơ sở để đưa ra các quyết định hay là căn cứ để so sánh kết quả của các phương án.

1.2 Tình hình dự báo dữ liệu chuỗi thời gian

Chính vì có nhiều ý nghĩa quan trọng và được ứng dụng rộng rãi trong nhiều lĩnh vực nên từ lâu đã có nhiều nhà khoa học tìm hiểu, nghiên cứu và mô hình hóa dữ liệu chuỗi thời gian để ứng dụng trong phân tích, dự báo. Trong những năm gần đây nhiều mô hình, phương pháp được đề xuất để cải thiện kết quả, tăng độ chính xác cho dự báo dữ liệu chuỗi thời gian nhưng nhìn chung các mô hình, phương pháp dự báo dữ liệu chuỗi thời gian tập trung vào các hướng nghiên cứu chính là:

- Các mô hình dự báo dựa trên các mô hình xác suất, thống kê như mô hình hồi quy (Auto Regression - AR) [9, 11], mô hình trung bình động (Moving Average - MA) [9, 11], mô hình tự hồi quy và trung bình động (Auto Regression Move Average - ARMA) [4, 9, 11], mô hình tự hồi quy kết hợp với trung bình động (Auto Regression Integrated Move Average) [4, 5, 9, 11]. Ngoài ra còn các mô hình là biến thể của các mô hình trên để phù hợp với đặc điểm của từng loại dữ liệu như mô hình SARIMA (Seasonal Auto Regression Integrated Move Average) [4, 9, 11].
- Hướng nghiên cứu thứ hai trong khai thác dữ liệu là hướng nghiên cứu tập trung vào các mô hình máy học (Machine Learning) như mô hình mạng neural (Neural Network) [10], thuật giải SVM (Support Vector Machine), thuật giải di truyền (Genetic Algorithm - GA) và các biến thể của các mô hình trên như SANN (Seasonal Artificial Neural Networks)[4].
- Một hướng nghiên cứu khác có nền tảng dựa trên lý thuyết logic mờ của GS. Lotfi Zadeh, đó là các phương pháp dự báo trên chuỗi thời gian mờ[17, 18].
- Trong những năm gần đây, hướng nghiên cứu kết hợp các mô hình dự báo dữ liệu chuỗi thời gian đang được nhiều nhà khoa học quan tâm nghiên cứu. Tiêu biểu là các mô hình kết hợp ARIMA và mạng neural [21], hay kết hợp mô hình ARIMA với thuật giải SVM [12, 13, 19], mô hình ARIMA mờ [20],...

Trong nước, những năm qua cũng có nhiều đề tài nghiên cứu và ứng dụng dự báo dữ liệu chuỗi thời gian như các mô hình dự báo sử dụng mô hình ARIMA [1, 3], mô hình chuỗi thời gian mờ [2].

1.3 Những vấn đề còn tồn tại

Thứ nhất, mỗi một mô hình, phương pháp dự báo dữ liệu chuỗi thời gian đều chỉ phù hợp với một số dạng dữ liệu đặc thù, mà chưa có một mô hình nào có thể dự báo tốt được cho tất cả các dạng dữ liệu, ví dụ như những mô hình dựa trên xác suất thống kê như mô hình hồi quy, mô hình trung bình động hay mô hình ARIMA chỉ phù hợp để dự báo cho các dữ liệu dạng tuyến tính, còn các mô hình máy học như ANN, SVM lại chỉ phù hợp để dự báo cho các dạng dữ liệu phi tuyến tính [12, 19]. Mặt khác, dữ liệu chuỗi thời gian trong thực tế đa số đều mang các đặc tính tuyến tính và phi tuyến tính, nên việc chỉ sử dụng một mô hình, phương pháp để dự báo dữ liệu chuỗi thời gian thường chưa mang lại kết quả như mong đợi. Do đó việc tìm hiểu và áp dụng kết hợp các mô hình, phương pháp dự báo dữ liệu chuỗi thời gian vào trong thực tế là cần thiết để tăng độ chính xác của kết quả dự báo.

Thứ hai, độ chính xác trong dự báo của mỗi mô hình, phương pháp dự báo dữ liệu chuỗi thời gian đơn lẻ thường chịu tác động từ nhiều yếu tố như mẫu dữ liệu dùng để xây dựng mô hình, mô hình không chính xác hay cấu trúc mô hình bị thay đổi trong quá trình vận hành thực tế,...những điều đó dẫn đến kết quả dự báo đôi khi bị sai lệch quá lớn so với thực tế. Đối với các phương pháp dự báo kết hợp nhiều mô hình, phương pháp dự báo lại với nhau, tuy cũng chịu những tác động tiêu cực như trên, nhưng do bản chất của phương pháp là sự kết hợp tương hỗ của các mô hình, phương pháp dự báo nên ít nhiều cũng giảm được sự tác động của các yếu tố ảnh hưởng đến kết quả dự báo của mô hình [21].

1.4 Mục tiêu, nội dung, phương pháp nghiên cứu

Mục tiêu của đề tài nhằm tìm hiểu và áp dụng kết hợp mô hình ARIMA và SVM trong dự báo dữ liệu chuỗi thời gian. Ứng dụng mô hình này vào dự báo số lượng giao dịch trên ngày cho Công ty Dịch vụ Trực tuyến Cộng Đồng Việt. Lý

Chương 1. Tổng quan

do đẽ tài lựa chọn mô hình ARIMA và phương pháp SVM để kết hợp dự báo dữ liệu chuỗi thời gian vì:

- Mô hình ARIMA và phương pháp SVM trong ước lượng hồi quy đều là những mô hình, phương pháp dự báo chuỗi thời gian cho kết quả dự báo tương đối tốt. Tùy thuộc vào tính chất của dữ liệu chuỗi thời gian mà mô hình ARIMA và phương pháp SVM thường được lựa chọn để thực hiện dự báo. Mô hình ARIMA được chọn để dự báo cho thành phần tuyến tính của chuỗi thời gian, còn phương pháp SVM thường được chọn để dự báo cho thành phần phi tuyến tính của chuỗi thời gian. Do đó mà mô hình kết hợp ARIMA và SVM trong dự báo dữ liệu chuỗi thời gian hy vọng sẽ phát huy được các ưu điểm của mô hình ARIMA cũng như phương pháp SVM để cho kết quả dự báo chính xác hơn là sử dụng một mô hình, phương pháp dự báo đơn lẻ.
- Thực tế đã có những nghiên cứu và ứng dụng cho thấy hiệu quả của phương pháp kết hợp ARIMA và SVM trong dự báo dữ liệu chuỗi thời gian như Ứng dụng mô hình kết hợp ARIMA và SVM trong dự báo chứng khoán [14]. Mô hình kết hợp ARIMA và SVM trong dự báo ngắn hạn, áp dụng trong lĩnh vực dự báo năng lượng [13] hay Ứng dụng mô hình kết hợp ARIMA và SVM trong dự báo trong lĩnh vực tròng trọt, chăn nuôi [19]. Tất cả các nghiên cứu và ứng dụng trên đều cho thấy kết quả dự báo của mô hình kết hợp ARIMA và SVM hiệu quả hơn so với các mô hình, phương pháp dự báo đơn lẻ.
- Mô hình ARIMA và phương pháp SVM đều là những mô hình, phương pháp dự báo dữ liệu chuỗi thời gian hiệu quả và đã được nghiên cứu từ lâu. Do đó mà các thư viện hỗ trợ cài đặt các mô hình, phương pháp này trong các ngôn ngữ lập trình nói chung và ngôn ngữ R nói riêng là tương đối đầy đủ. Chính vì vậy mà việc cài đặt và thử nghiệm mô hình kết hợp ARIMA và phương pháp SVM là tương đối thuận lợi và nhanh chóng. Bên cạnh đó các tài liệu nghiên cứu về mô hình ARIMA và phương pháp SVM cũng rất đa dạng và phong phú.

Chương 1. Tổng quan

Nội dung nghiên cứu của đề tài bao gồm:

- Tìm hiểu các mô hình dự báo dữ liệu chuỗi thời gian, tập trung tìm hiểu các mô hình hồi quy, mô hình ARIMA và mô hình kết hợp ARIMA với SVM.
- Tiền xử lý dữ liệu để biến đổi dữ liệu về dạng phù hợp với các mô hình dự báo.
- Tiến hành cài đặt và thử nghiệm các mô hình dự báo dựa trên tập dữ liệu được thu thập từ dữ liệu của Công ty Dịch vụ Trực tuyến Cộng Đồng Việt.
- So sánh, đánh giá kết quả dự báo của các mô hình với nhau và với dữ liệu thực tế.

Phương pháp nghiên cứu của đề tài:

- Tìm hiểu các mô hình, phương pháp trong dự báo dữ liệu chuỗi thời gian.
- Tìm hiểu mô hình ARIMA.
- Tìm hiểu về SVM và ứng dụng SVM vào dự báo dữ liệu chuỗi thời gian.
- Tìm hiểu phương pháp kết hợp mô hình ARIMA và SVM để tăng độ chính xác kết quả dự báo.
- Tìm hiểu về các độ đo để đánh giá kết quả dự báo dữ liệu chuỗi thời gian.
- Cài đặt thử nghiệm các mô hình, phương pháp dự báo dữ liệu chuỗi thời gian.

Chương 2. PHƯƠNG PHÁP DỰ BÁO

DỮ LIỆU CHUỖI THỜI GIAN

Hiện nay có rất nhiều phương pháp dự báo dữ liệu chuỗi thời gian từ các phương pháp xác suất - thống kê đến các phương pháp máy học. Trong mỗi phương pháp lại có những biến thể khác nhau tùy thuộc vào đặc điểm của từng loại hình dữ liệu. Do đó chương này sẽ tổng hợp các phương pháp dự báo dữ liệu chuỗi thời gian phổ biến hiện nay. Mỗi phương pháp sẽ được trình bày một cách tổng quan về ý tưởng xây dựng mô hình cũng như các bước thực hiện dự báo.

2.1 Phương pháp xác suất – thống kê

2.1.1 Mô hình hồi quy

Mô hình hồi quy (Auto Regression - AR) là một mô hình cơ bản trong xác suất – thống kê. Mô hình này dự báo giá trị của chuỗi thời gian dựa trên một hoặc nhiều giá trị của chuỗi thời gian trước đó cộng với một giá trị ngẫu nhiên gọi là *nhiều trắng* (white noise).

Tổng quát, một quá trình hồi quy (Autoregressive process) bậc p được biểu diễn như sau [9]:

$$z_t - \mu = \phi_1(z_{t-1} - \mu) + \phi_2(z_{t-2} - \mu) + \dots + \phi_p(z_{t-p} - \mu) + a_t \quad (2.1)$$

Trong đó:

+ z_t là giá trị của chuỗi thời gian tại thời điểm t .

+ μ là giá trị trung bình của chuỗi thời gian.

+ $\phi_1, \phi_2, \dots, \phi_p$ là các tham số của mô hình.

+ a_t là nhiễu trắng tại thời điểm t . Đây là giá trị ngẫu nhiên có phân phối độc lập (indentically independently distributed - IID) với giá trị trung bình là 0 và phương sai là σ_a^2

Chương 2. Phương pháp dự báo dữ liệu chuỗi thời gian

Nếu bậc $p = 1$ thì quá trình hồi quy AR(1) được gọi là quá trình Markov (Markov process).

Với toán tử backward shift (backward shift B operator) được định nghĩa như sau:

$$B^k z_t = z_{t-k} \quad (2.2)$$

Phương trình (2.1) được biểu diễn dưới dạng:

$$z_t - \mu = \phi_1(Bz_t - \mu) + \phi_2(B^2 z_t - \mu) + \cdots + \phi_p(B^p z_t - \mu) + a_t \quad (2.3)$$

Vì B như là một toán tử đại số nên ta có thể biến đổi phương trình (2.3) về dạng:

$$(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p)(z_t - \mu) = a_t \quad (2.4)$$

hay

$$\phi(B)(z_t - \mu) = a_t \quad (2.5)$$

Với $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$ được gọi là toán tử hồi quy (AR operator) bậc p .

Có nhiều phương pháp ước lượng các tham số của mô hình trong đó hay sử dụng nhất là phương pháp tối thiểu tổng bình phương các sai số hay bình phương cực tiểu. Tuy nhiên hiện nay các phần mềm thống kê như R, Eviews,... đã có tích hợp các module giúp tự động tính toán, ước lượng các tham số này. Do đó trong khuôn khổ của báo cáo sẽ không tìm hiểu sâu về các phương pháp ước lượng tham số của các mô hình. Nếu quan tâm bạn đọc có thể tham khảo trong các tài liệu [6, 9].

2.1.2 Mô hình trung bình động

Tương tự như mô hình hồi quy mô hình trung bình động (Move Average - MA) cũng là một mô hình cơ bản trong xác suất – thống kê. Mô hình này dự báo

Chương 2. Phương pháp dự báo dữ liệu chuỗi thời gian

giá trị của chuỗi thời gian dựa trên một hoặc nhiều giá trị ngẫu nhiên nhiễu trắng của chuỗi thời gian trước đó.

Tổng quát, một quá trình trung bình động (Moving average process) bậc q được biểu diễn như sau [9]:

$$z_t - \mu = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q} \quad (2.6)$$

Trong đó:

+ z_t là giá trị của chuỗi thời gian tại thời điểm t .

+ μ là giá trị trung bình của chuỗi thời gian.

+ $\theta_1, \theta_2, \dots, \theta_q$ là các tham số của mô hình.

+ a_t là nhiễu trắng tại thời điểm t . Đây là giá trị ngẫu nhiên có phân phối độc lập với giá trị trung bình là 0 và phương sai là σ_a^2

Với toán tử backward shift B được định nghĩa trong (2.2), phương trình (2.6) được biểu diễn dưới dạng:

$$\begin{aligned} z_t - \mu &= a_t - \theta_1 B a_t - \theta_2 B^2 a_t - \cdots - \theta_q B^q a_t \\ &= (1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q) a_t \\ &= \theta(B) a_t \end{aligned} \quad (2.7)$$

Với $\theta(q) = (1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q)$ được gọi là toán tử trung bình động (MA operator) bậc q .

2.1.3 Mô hình ARMA

Mô hình ARMA (Auto Regression Move Average) là mô hình kết hợp cả hai mô hình hồi quy (AR) và mô hình trung bình động (MA). Do đó mô hình ARMA mang đặc tính của hai mô hình hồi quy và trung bình động.

Tổng quát một quá trình ARMA (ARMA process) bậc p hồi quy và bậc q trung bình động được biểu diễn như sau [9]:

Chương 2. Phương pháp dự báo dữ liệu chuỗi thời gian

$$(z_t - \mu) - \phi_1(z_{t-1} - \mu) - \phi_2(z_{t-2} - \mu) - \cdots - \phi_p(z_{t-p} - \mu) \\ = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q} \quad (2.8)$$

Trong đó:

+ z_t là giá trị của chuỗi thời gian tại thời điểm t .

+ μ là giá trị trung bình của chuỗi thời gian.

+ $\phi_1, \phi_2, \dots, \phi_p$ và $\theta_1, \theta_2, \dots, \theta_q$ là các tham số của mô hình.

+ a_t là nhiễu trắng tại thời điểm t . Đây là giá trị ngẫu nhiên có phân phối độc lập với giá trị trung bình là 0 và phương sai là σ_a^2

Với toán tử backward shift B được định nghĩa trong (2.2), phương trình (2.8) được biểu diễn dưới dạng:

$$(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p)(z_t - \mu) \\ = (1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q)a_t \quad (2.9)$$

hay

$$\phi(B)(z_t - \mu) = \theta(B)a_t \quad (2.10)$$

Với $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$ được gọi là toán tử hồi quy (AR operator) bậc p và $\theta(q) = (1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q)$ được gọi là toán tử trung bình động (MA operator) bậc q .

2.2 Phương pháp máy học

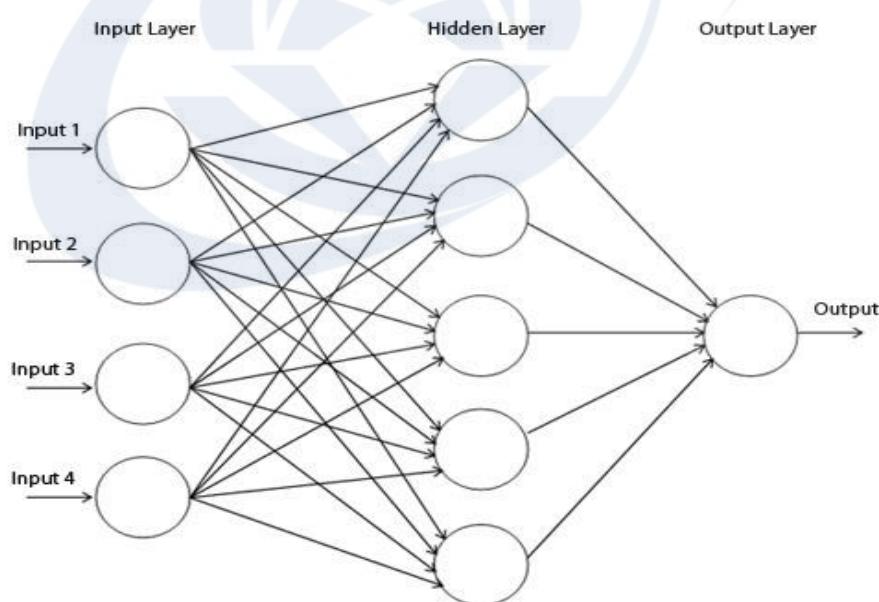
2.2.1 Phương pháp mạng neural

Mạng neural nhận tạo (Artificial Neural Network - ANN) hay gọi tắt là mạng neural là một trong những phương pháp máy học quan trọng được nghiên cứu, ứng dụng trong nhiều lĩnh vực của khoa học máy tính như nhận diện mẫu (pattern recognition), xử lý ảnh (image processing), xử lý văn bản,... Bên cạnh các lĩnh vực kể trên, trong những năm gần đây mạng neural bắt đầu được nghiên cứu, ứng dụng trong khai thác dữ liệu mà tiêu biểu là ứng dụng trong dự báo dữ liệu chuỗi thời gian. Do mạng neural là một trong những phương pháp cơ bản của khoa học máy tính và đã được ứng dụng trong nhiều lĩnh vực nên trong báo cáo này sẽ không trình bày lại các khái niệm cơ bản, cách hoạt động và ứng dụng

Chương 2. Phương pháp dự báo dữ liệu chuỗi thời gian

của mạng neural trong từng lĩnh vực cụ thể. Nếu độc giả quan tâm có thể tham khảo các tài liệu chuyên về mạng neural. Trong báo cáo này chỉ tập trung trình bày về mạng neural ứng dụng trong khai thác dữ liệu, đặc biệt là ứng dụng trong lĩnh vực dự báo dữ liệu chuỗi thời gian.

Trong lĩnh vực dự báo dữ liệu chuỗi thời gian, mạng neural thường được ứng dụng để dự báo các bài toán, vấn đề về thương mại và tài chính như dự báo giá sản phẩm, dự báo tỉ giá tiền tệ, xếp hạng tín dụng, dự báo giá chứng khoán, dự báo ngân hàng phá sản,... Ưu điểm lớn nhất của mạng neural giúp nó phù hợp trong lĩnh vực khai thác dữ liệu nói chung và dự báo dữ liệu chuỗi thời gian nói riêng đến từ cách tiếp cận của mô hình. Hướng tiếp cận của mô hình mạng neural là hướng tiếp cận theo hướng dữ liệu, tức là sẽ không có một khuôn mẫu chung nào cho mô hình, thay vào đó mô hình mạng neural sẽ được học và hoàn thiện từ dữ liệu. Cách tiếp cận này phù hợp với các tập dữ liệu không tuyến tính (nonlinear data sets), nơi mà không có một mô hình, lý thuyết thống kê nào có thể mô hình hóa được.



Hình 2.1. Mạng neural truyền thẳng 3 lớp

Nguồn: *Cấu trúc mạng neural* [30]

Về cơ bản, một mạng neural truyền thẳng (feedforward neural network) 3 lớp (input layer, hidden layer, output layer) được biểu diễn như sau [21]:

Chương 2. Phương pháp dự báo dữ liệu chuỗi thời gian

$$y_t = \alpha_0 + \sum_{j=1}^q \alpha_j g \left(\beta_{0j} + \sum_{i=1}^p \beta_{ij} y_{t-i} \right) + \varepsilon_t \quad (2.11.1)$$

Trong đó:

+ y_t là giá trị output của mạng neural.

+ $y_{t-1}, y_{t-2}, \dots, y_{t-p}$ là các giá trị input của mạng neural.

+ α_j với $j = 0, 1, 2, \dots, q$ và β_{ij} với $i = 0, 1, 2, \dots, p; j = 0, 1, 2, \dots, q$ là trọng số của các kết nối giữa các nerval.

+ p là số lượng các nodes ở lớp input.

+ q là số lượng các nodes ở lớp hidden.

+ ε_t là giá trị ngẫu nhiên tại thời điểm t .

+ Hàm chuyển đổi $g(x)$ của lớp hidden được định nghĩa như sau:

$$g(x) = \frac{1}{1 + \exp(-x)} \quad (2.11.2)$$

Nếu xem các giá trị $y_{t-1}, y_{t-2}, \dots, y_{t-p}$ như là các giá trị chuỗi thời gian trong quá khứ, thì y_t được xem như giá trị chuỗi thời gian được dự báo. Khi đó mạng neural có thể dự báo được giá trị của chuỗi thời gian từ những dữ liệu của chuỗi thời gian trong quá khứ. Dĩ nhiên để có thể dự báo chính xác giá trị của chuỗi thời gian, mạng neural cần phải trải qua một quá trình học, hay quá trình điều chỉnh các trọng số của mô hình để phù hợp với dữ liệu chuỗi thời gian đang xét.

Một cách tổng quát, mạng neural được áp dụng trong dự báo dữ liệu chuỗi thời gian được biểu diễn dưới dạng sau [21]:

$$y_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-p}, w) + \varepsilon_t \quad (2.12)$$

Trong đó:

+ w là vector tất cả các trọng số của mạng neural.

+ f được xem như là một hàm phi tuyến tính được xác định bởi cấu trúc mạng neural và các trọng số kết nối.

2.2.2 Phương pháp thuật giải di truyền

Cũng giống như mạng neural, phương pháp thuật giải di truyền (Genetic algorithm) cũng là một trong những phương pháp máy học quan trọng được nghiên cứu, ứng dụng rộng rãi trong nhiều lĩnh vực của khoa học máy tính. Ý tưởng của thuật giải di truyền là mô phỏng theo quá trình di truyền và tiến hóa của sinh vật với các đặc điểm: chọn lọc, lai tạo và đột biến. Cũng giống như mạng neural trong báo cáo này chỉ tập trung trình bày về thuật giải di truyền ứng dụng trong khai thác dữ liệu và dự báo dữ liệu chuỗi thời gian. Nếu độc giả quan tâm về thuật giải di truyền có thể tham khảo các tài liệu về thuật giải di truyền.

Thuật giải di truyền trong dự báo dữ liệu chuỗi thời gian được miêu tả bằng đoạn mã giả như sau [8]:

```

Generic Algorithm
// For forecasting time series
// input: N: số lần lặp
// M: số cá thể khởi tạo ban đầu
// Xmin, Xmax: chặn dưới và chặn trên cho giá trị
// của các cá thể khi khởi tạo ban đầu
// output: Xbest cá thể hay giá trị dự báo tốt nhất
1. P[i] = Xmin + (Xmax - Xmin).rand();
F[] = Ø;
count = 0;
Xbest = Xmax;
// Khởi tạo tập các cá thể
// P[] là mảng lưu các cá thể
// F[] là mảng lưu độ phù hợp của từng cá thể
// rand() là hàm sinh số ngẫu nhiên
// count là biến đếm số lần lặp
2. do
{
    for (int i = 0; i < P.len; i++)
    {
        F[i] = |ActualValue - P[i]|;
        // tính độ phù hợp
    }

    // lai tạo
    parent1, parent2 = ChooseParent(P[], F[]);
    // chọn cặp bố mẹ có độ phù hợp bé nhất
}

```

Chương 2. Phương pháp dự báo dữ liệu chuỗi thời gian

```
child1, child2 = Crossing(parent1, parent2);
P.add(child1, child2); // bổ sung 2 cá thể mới

// đột biến
Mutation(P[]);
// chọn 1 cá thể và tạo đột biến
Xbest = Best(P[], F[]);
// chọn cá thể có độ phù hợp bé nhất
}while(count < N)
3. Return Xbest
```

2.3 Phương pháp logic mờ

2.3.1 Phương pháp chuỗi thời gian mờ

Logic mờ (fuzzy logic) được giới thiệu lần đầu tiên vào năm 1965 bởi GS.

Lotfi Zadeh tại Đại học Barkeley, California. Từ đó logic mờ được ứng dụng rộng rãi trong nhiều lĩnh vực khoa học kỹ thuật như điều khiển tự động hóa,...

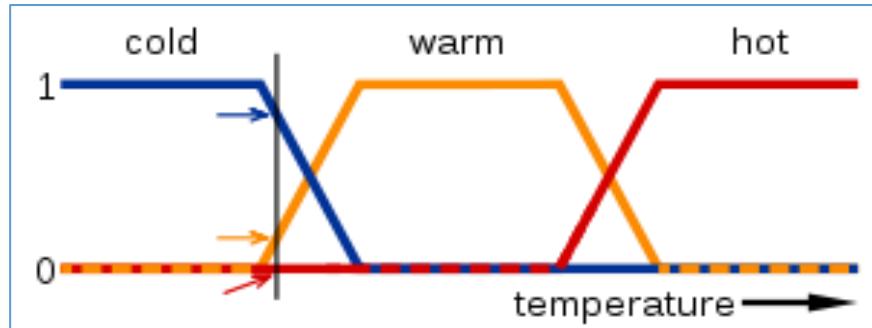
Khoa học máy tính cũng là một trong những lĩnh vực ứng dụng lý thuyết logic mờ để giải quyết các vấn đề, một trong số đó là ứng dụng logic mờ trong vấn đề dự báo dữ liệu chuỗi thời gian.

Logic mờ là mở rộng của lý thuyết logic cổ điển. Trong logic cổ điển, mỗi biến logic chỉ mang hai giá trị 0 hoặc 1 (đúng hoặc sai, true or false). Tuy nhiên trong logic mờ, mỗi biến logic nhận giá trị là một số thực nằm trong đoạn từ 0 đến 1 và giá trị này được biểu diễn thông qua một hàm số gọi là hàm thành phần (membership function). Hình 2.2 mô tả về 3 hàm thành phần đại diện cho 3 tập logic mờ là cold, warm, hot. Tại mỗi điểm nhiệt độ, giá trị của 3 hàm thành phần này là khác nhau. Ví dụ tại điểm nhiệt độ x được biểu thị bằng đường thẳng màu xám, tại đây nhiệt độ vào khoảng 10°C , khi đó giá trị của các hàm thành phần được biểu diễn như sau:

$$+ f_{cold}(x) = 0.8 \quad (\text{rất lạnh})$$

$$+ f_{warm}(x) = 0.1 \quad (\text{ít ám})$$

$$+ f_{hot}(x) = 0 \quad (\text{không nóng})$$



Hình 2.2. Logic mờ

Nhờ có logic mờ mà các khái niệm đo lường không tường minh trong thực tế như *rất, hơi, vừa, ít, ...* được biểu diễn một cách cụ thể trong toán học. Do trong báo cáo này chỉ tập trung trình bày phương pháp ứng dụng logic mờ trong dự báo dữ liệu chuỗi thời gian nên phần giới thiệu về logic mờ chỉ dừng lại ở các khái niệm cơ bản nhất. Các phần chi tiết và chuyên sâu hơn về logic mờ như các phép toán logic mờ, phương pháp mờ hóa và giải mờ,... bạn đọc có thể tham khảo thêm các tài liệu chuyên về logic mờ.

Để bắt đầu tìm hiểu về phương pháp ứng dụng logic mờ vào dự báo dữ liệu chuỗi thời gian trước tiên cần tìm hiểu về định nghĩa chuỗi thời gian mờ như sau [17, 18]:

Định nghĩa 2.1:

$Y(t)$ ($t = 0, 1, 2, \dots$) là một chuỗi thời gian, $f_i(t)$ ($i = 1, 2, \dots$) là các tập mờ, khi đó $F(t)$ là tập hợp các $f_1(t), f_2(t), \dots$ được gọi là chuỗi thời gian mờ định nghĩa dựa trên $Y(t)$ ($t = 0, 1, 2, \dots$).

Từ định nghĩa trên có thể xem $F(t)$ như là một biến ký tự (linguistic variable) và $f_i(t)$ ($i = 1, 2, \dots$) là các giá trị có thể của $F(t)$. Tại mỗi thời điểm t khác nhau, giá trị của $F(t)$ là khác nhau. Điểm khác nhau cơ bản giữa chuỗi thời gian thường và chuỗi thời gian mờ là giá trị của chuỗi thời gian, đối với chuỗi thời gian thường thì giá trị này là số thực, trong khi đối với chuỗi thời gian mờ nó là một tập mờ [17, 18].

Chương 2. Phương pháp dự báo dữ liệu chuỗi thời gian

Cũng giống các phương pháp dự báo chuỗi thời gian khác, phương pháp dự báo chuỗi thời gian mờ được xây dựng dựa trên ý tưởng tìm kiếm mối quan hệ giữa giá trị của chuỗi thời gian tại thời điểm t với giá trị của chuỗi thời gian tại các thời điểm trước đó. Từ ý tưởng này, quan hệ mờ được định nghĩa như sau [17]:

Định nghĩa 2.2:

Nếu $\forall f_j(t) \in F(t), j \in J. \exists f_i(t-1) \in F(t-1), i \in I$ sao cho $f_j(t) = f_i(t-1)^o R_{ij}(t, t-1)$ với ‘ o ’ là phép max-min (max-min operator) thì $F(t)$ được sinh ra hay có quan hệ mờ với $F(t-1)$. Ký hiệu $F(t-1) \rightarrow F(t)$. Với I và J lần lượt là tập các chỉ số của $F(t-1)$ và $F(t)$. $R_{ij}(t, t-1)$ là quan hệ mờ.

Nếu ký hiệu $R(t, t-1) = \bigcup_{ij} R_{ij}(t, t-1)$ thì quan hệ mờ giữa $F(t-1)$ và $F(t)$ được biểu diễn lại dưới dạng [17]:

$$F(t) = F(t-1)^o R(t, t-1) \quad (2.13)$$

Từ định nghĩa quan hệ mờ như trên, chuỗi thời gian mờ bất biến được định nghĩa như sau (invariant fuzzy time series) như sau [17]:

Định nghĩa 2.3:

Với $R(t, t-1)$ là quan hệ mờ của $F(t)$. Nếu $\forall t, R(t, t-1)$ là độc lập với t , tức là $\forall t, R(t, t-1) = R(t-1, t-2)$ thì $F(t)$ được gọi là chuỗi thời gian mờ bất biến. Ngược lại $F(t)$ được gọi là chuỗi thời gian mờ thay đổi (variant fuzzy time series).

Các bước dự báo dữ liệu chuỗi thời gian mờ (áp dụng cho chuỗi thời gian mờ bất biến) [18]:

+ Bước 1: Định nghĩa tập vũ trụ U (universes set) cho các tập mờ dựa trên dữ liệu chuỗi thời gian thu thập được. Thường thì tập vũ trụ U được định nghĩa là $U = [D_{min} - D_1, D_{max} + D_2]$, với D_{min}, D_{max} lần lượt là giá trị bé nhất và giá trị lớn nhất của tập dữ liệu. D_1, D_2 là 2 số dương tùy chọn.

+ Bước 2: Chia tập vũ trụ thành các tập con có chiều dài tương đương nhau.

Chương 2. Phương pháp dự báo dữ liệu chuỗi thời gian

- + Bước 3: Định nghĩa các tập mờ trên tập vũ trụ U , sử dụng các biến ngôn ngữ để ký hiệu các tập mờ. Xác định các hàm thành phần cho mỗi tập mờ dựa trên các tập con của tập vũ trụ U được định nghĩa trong bước 2.
- + Bước 4: Mờ hóa dữ liệu chuỗi thời gian, sử dụng các tập mờ định nghĩa trong bước 3.
- + Bước 5: Xác định các quan hệ mờ dựa trên dữ liệu chuỗi thời gian đã được mờ hóa.
- + Bước 6: Tính giá trị dự báo dựa trên các giá trị của chuỗi thời gian mờ trước đó và các quan hệ mờ đã được định nghĩa.
- + Bước 7: Giải mờ, chuyển giá trị dự báo từ giá trị mờ thành giá trị thực.

2.4 Phương pháp kết hợp

2.4.1 Kết hợp ARIMA và mạng neural

Cả mô hình ARIMA và mô hình mạng neural đều là những mô hình phù hợp để dự báo dữ liệu chuỗi thời gian. Tuy nhiên mỗi mô hình lại chỉ phù hợp với một số dạng dữ liệu đặc thù, như mô hình ARIMA phù hợp với dự báo dữ liệu chuỗi thời gian dạng tuyến tính, còn mô hình mạng neural lại phù hợp với dự báo dữ liệu chuỗi thời gian dạng phi tuyến tính. Do đó mà mô hình kết hợp giữa ARIMA và mạng neural có thể giúp tăng độ chính xác của dự báo trong thực tế.

Ý tưởng của mô hình này dựa trên việc xem xét dữ liệu chuỗi thời gian là sự kết hợp giữa hai thành phần tuyến tính và phi tuyến tính và hai thành phần này được ước lượng thông qua dữ liệu [21].

$$y_t = L_t + N_t \quad (2.14)$$

Trong đó:

- + y_t là giá trị của chuỗi thời gian
- + L_t là thành phần tuyến tính (linear component)
- + N_t là thành phần phi tuyến tính (nonlinear component)

Chương 2. Phương pháp dự báo dữ liệu chuỗi thời gian

Để dự báo giá trị của chuỗi thời gian, đầu tiên mô hình kết hợp ARIMA và mạng neural sử dụng mô hình ARIMA để dự báo cho thành phần tuyến tính. Khi đó, giá trị còn lại sẽ được dự báo bằng mạng neural. Gọi e_t là giá trị còn lại sau khi sử dụng mô hình ARIMA để dự báo [21]. Khi đó:

$$e_t = y_t - \hat{L}_t \quad (2.15)$$

Trong đó:

+ \hat{L}_t là giá trị dự báo cho thành phần tuyến tính tại thời điểm t

Mô hình mạng neural được dùng để dự báo giá trị còn lại e_t sau khi dự báo bằng mô hình ARIMA. Khi đó giá trị còn lại e_t sẽ được dự báo như sau [21]:

$$e_t = f(e_{t-1}, e_{t-2}, \dots, e_{t-n}) + \varepsilon_t \quad (2.16)$$

Trong đó:

+ f là một hàm phi tuyến được xác định bằng mạng neural

+ ε_t là giá trị ngẫu nhiên tại thời điểm t .

Ký hiệu \hat{N}_t là giá trị dự báo cho thành phần phi tuyến tính trong phương trình (2.16). Khi đó giá trị dự báo tại thời điểm t là [21]:

$$\hat{y}_t = \hat{L}_t + \hat{N}_t \quad (2.17)$$

Tổng kết, mô hình kết hợp ARIMA và mạng neural thực hiện hai bước để dự báo giá trị của chuỗi thời gian.

+ Bước 1: Dự báo thành phần tuyến tính của chuỗi thời gian bằng mô hình ARIMA.

+ Bước 2: Dự báo thành phần phi tuyến tính của chuỗi thời gian bằng mô hình mạng neural.

2.4.2 Mô hình ARIMA mờ

Đối với mô hình ARIMA thường thì dữ liệu chuỗi thời gian dùng để xây dựng mô hình phải có tối thiểu 50 điểm dữ liệu và mô hình ARIMA chỉ dự đoán tốt đối với các chuỗi thời gian có từ 100 điểm dữ liệu trở lên [27]. Tuy nhiên,

Chương 2. Phương pháp dự báo dữ liệu chuỗi thời gian

trong thực tế có những trường hợp do môi trường không chắc chắn hoặc do dữ liệu bị thay đổi liên tục dẫn đến những tình huống phải dự báo với ít dữ liệu quá khứ, ảnh hưởng đến kết quả dự báo của mô hình ARIMA, đây là một nhược điểm của mô hình ARIMA. Chính vì vậy mà mô hình ARIMA mờ được đề xuất để cải thiện kết quả dự báo trong trường hợp ít dữ liệu.

ARIMA mờ (Fuzzy Auto Regression Integrated Move Average - FARIMA) là phương pháp kết hợp mô hình ARIMA và mô hình hồi quy mờ (fuzzy regression). Ý tưởng chính của mô hình ARIMA mờ là thay vì sử dụng mô hình ARIMA với các tham số của mô hình như $\phi_1, \phi_2, \dots, \phi_p$ và $\theta_1, \theta_2, \dots, \theta_q$ là các giá trị số thực, mô hình ARIMA mờ sử dụng hồi quy mờ để mờ hóa các tham số của mô hình ARIMA thành các tham số mờ $\widetilde{\phi}_1, \widetilde{\phi}_2, \dots, \widetilde{\phi}_p$ và $\widetilde{\theta}_1, \widetilde{\theta}_2, \dots, \widetilde{\theta}_q$, từ đó giảm yêu cầu về số lượng các điểm dữ liệu của chuỗi thời gian dùng để xây dựng mô hình.

Mô hình ARIMA mờ khi được áp dụng trong dự báo dữ liệu chuỗi thời gian thường thực hiện qua 3 giai đoạn sau [27]:

+ *Giai đoạn 1:* Xác định mô hình ARIMA(p, d, q) dựa trên dữ liệu chuỗi thời gian. Kết quả của giai đoạn này là các tham số của mô hình, $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_{p+q}^*)$ và giá trị nhiễu trắng a_t . Kết quả của giai đoạn này sẽ là một phần trong dữ liệu đầu vào của giai đoạn 2.

+ *Giai đoạn 2:* Mờ hóa các tham số và xác định mô hình ARIMA mờ như sau:

$$\begin{aligned}\widetilde{W}_t = & \langle \alpha_1, c_1 \rangle W_{t-1} + \dots + \langle \alpha_p, c_p \rangle W_{t-p} + a_t - \langle \alpha_{p+1}, c_{p+1} \rangle a_{t-1} \\ & - \dots - \langle \alpha_{p+q}, c_{p+q} \rangle a_{t-q}\end{aligned}\tag{2.17}$$

Trong đó:

$$+ W_t = (1 - B)^d (Z_t - \mu)$$

+ α_i là trung tâm của số mờ

+ c_i là độ rộng xung quanh giá trị trung tâm của số mờ.

Chương 2. Phương pháp dự báo dữ liệu chuỗi thời gian

+ *Giai đoạn 3:* Tối ưu hóa giá trị dự báo bằng cách xóa đi các giá trị nằm ngoài biên dự báo của mô hình ARIMA mờ.

Tổng kết, mô hình FARIMA có thể dùng để thay thế mô hình ARIMA để dự báo giá trị của chuỗi thời gian trong trường hợp dữ liệu dùng để xây dựng mô hình tương đối hạn chế. Tuy nhiên, trong trường hợp có đầy đủ dữ liệu để xây dựng mô hình thì mô hình ARIMA vẫn vượt trội hơn mô hình FARIMA về kết quả dự báo cũng như chi phí để xây dựng mô hình. Vì vậy mà trong từng trường hợp cụ thể có thể cân nhắc để chọn mô hình phù hợp nhất cho bài toán dự báo dữ liệu chuỗi thời gian.



Chương 3. MÔ HÌNH KẾT HỢP ARIMA VÀ SUPPORT VECTOR MACHINE

Cả mô hình ARIMA và thuật giải Support Vector Machine (SVM) đều là những mô hình, phương pháp nổi bật trong lĩnh vực dự báo dữ liệu chuỗi thời gian. Mỗi mô hình đều mang những đặc điểm riêng biệt phù hợp với từng loại hình dữ liệu khác nhau. Trong chương này sẽ trình bày chi tiết về hai mô hình dự báo dữ liệu chuỗi thời gian là ARIMA và SVM, cũng như mô hình kết hợp ARIMA và SVM.

3.1 Mô hình ARIMA

Mô hình ARIMA là một mô hình khá “nổi tiếng” trong số các mô hình dự báo dữ liệu chuỗi thời gian. Hầu như các báo cáo khoa học trong lĩnh vực dự báo dữ liệu chuỗi thời gian đều ít nhiều đề cập đến mô hình này. Trong mục này sẽ trình bày về các tính chất quan trọng của chuỗi thời gian liên quan đến mô hình ARIMA và giới thiệu mô hình ARIMA trong dự báo dữ liệu chuỗi thời gian.

3.1.1 Tính dừng của chuỗi thời gian

Tính dừng (stationarity) của một quá trình ngẫu nhiên (hay chuỗi thời gian) có thể được xem như một mẫu thống kê cân bằng, tức là các tính chất thống kê của quá trình ngẫu nhiên đó không phải là một hàm theo thời gian, cụ thể hơn là giá trị trung bình và phương sai của quá trình ngẫu nhiên độc lập với yếu tố thời gian [9]. Bên cạnh đó, chuỗi thời gian dừng phải thỏa mãn điều kiện hiệp phương sai giữa hai thời đoạn chỉ phụ thuộc vào khoảng cách hay độ trễ về thời gian giữa hai thời đoạn này chứ không phụ thuộc vào thời điểm thực tế mà hiệp phương sai được tính [3]. Trong trường hợp ngược lại, chuỗi thời gian được xem như không có tính dừng (nonstationarity). Một cách tổng quát, chuỗi thời gian dừng được định nghĩa như sau [6]:

Định nghĩa 3.1:

Chuỗi thời gian X_t ($t = 1, 2, \dots$) là chuỗi thời gian dừng nếu

(i) $\mu_X(t)$ là độc lập với thời gian t

và

(ii) $\gamma_X(t + h, t)$ là độc lập với thời gian t với mỗi h

Trong đó:

+ h là một số nguyên được gọi là độ trễ

+ $\mu_X(t)$ là giá trị trung bình của chuỗi thời gian

+ $\gamma_X(r, s) = Cov(X_r, X_s) = E[(X_r - \mu_X(r))(X_s - \mu_X(s))]$ được gọi là hiệp phương sai của chuỗi thời gian. r và s là hai số nguyên bất kỳ.

Tính dừng là một tính chất quan trọng của chuỗi thời gian. Mô hình ARMA được đề cập trong chương 2 chỉ có thể dự báo trên những chuỗi thời gian dừng. Đối với những chuỗi thời gian không dừng trước khi được dự báo cần biến đổi chuỗi thời gian về dạng chuỗi thời gian dừng. Cách đơn giản thường dùng để biến đổi chuỗi thời gian về dạng chuỗi thời gian dừng là lấy sai phân. Thường thì sau một hoặc hai lần lấy sai phân chuỗi thời gian sẽ về dạng chuỗi thời gian dừng. Chi tiết về phương pháp biến đổi chuỗi thời gian về dạng chuỗi thời gian dừng sẽ được trình bày trong mục 3.1.5.

Tính dừng của một chuỗi thời gian có thể nhận biết dựa trên đồ thị của chuỗi thời gian, đồ thị của hàm tự tương quan mẫu hay kiểm định Dickey – Fuller [1].

+ Dựa vào đồ thị của chuỗi thời gian: Một cách trực quan chuỗi thời gian có tính dừng nếu như đồ thị của chuỗi thời gian cho thấy giá trị trung bình và phương sai của chuỗi thời gian không thay đổi theo thời gian. Hình 3.1 biểu diễn đồ thị của chuỗi thời gian không dừng, hình 3.2 biểu diễn chuỗi đồ thị của chuỗi thời gian dừng sau khi lấy sai phân một lần.

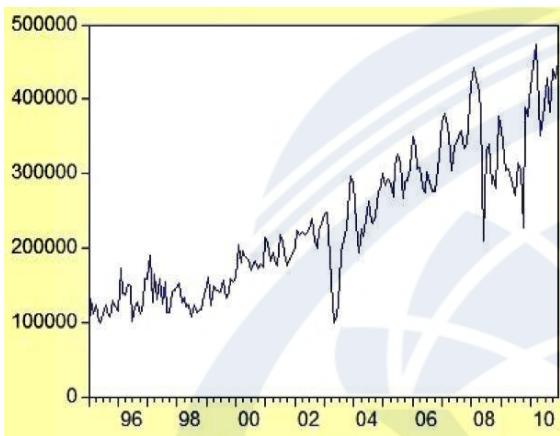
+ Dựa vào hàm tự tương quan mẫu (SAC – Sample Auto Correlation): Nếu đồ thị của hàm tự tương quan mẫu của một chuỗi thời gian giảm nhanh và tắt dần về 0 thì chuỗi có tính dừng. Hàm tự tương quan mẫu được định nghĩa như sau:

$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)} \quad (3.1)$$

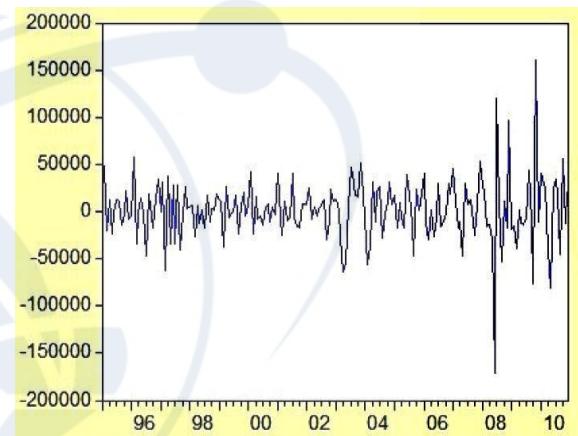
Trong đó:

$$\hat{\gamma}(h) = \frac{\sum_{t=1}^{n-|h|} (X_{t+|h|} - \mu_X)(X_t - \mu_X)}{n} \quad (3.1.1)$$

+ Kiểm định Dickey – Fuller nhằm xác định xem chuỗi thời gian có phải là Bước Ngẫu Nhiên (Random Walk, nghĩa là $X_t = X_{t-1} + e_t$) hay không. Nếu chuỗi thời gian là một Bước Ngẫu Nhiên thì không có tính dừng.



Hình 3.1. Chuỗi thời gian không dừng



Hình 3.2. Chuỗi thời gian dừng

Nguồn: *Diễn biến lượng khách Quốc tế đến Việt Nam giai đoạn 1995 – 2010 (lượt)* [1]

Nguồn: *Diễn biến lượng khách Quốc tế đến Việt Nam giai đoạn 1995 – 2010 (lượt)* [1]

3.1.2 Tính mùa của chuỗi thời gian

Tính mùa (seasonal) như đã đề cập trong chương 1 là một trong những tính chất của chuỗi thời gian. Tính mùa là sự thay đổi của chuỗi thời gian theo một khoảng thời gian trong năm. Có nhiều yếu tố gây ra sự thay đổi chuỗi thời gian theo mùa như yếu tố thời tiết, thói quen truyền thống,... Ví dụ máy lạnh, tủ lạnh thường được tiêu thụ nhiều vào mùa hè và giảm vào mùa đông hay doanh số bán hàng thường tăng mạnh vào những dịp lễ tết cuối năm.

Để xác định chuỗi thời gian có tính mùa hay không có thể dựa vào đồ thị của chuỗi thời gian hay đồ thị của hàm tự tương quan mẫu. Đối với đồ thị của

Chương 3. Mô hình kết hợp ARIMA và Support Vector Machine

chuỗi thời gian, nếu cứ sau một năm đồ thị của chuỗi thời gian có hình dạng tương tự như thời điểm này năm trước thì nhiều khả năng chuỗi thời gian đó có tính mùa. Đối với đồ thị hàm tự tương quan mẫu, nếu cứ sau m thời đoạn đồ thị hàm tự tương quan mẫu lại có giá trị cao thì chuỗi thời gian đó có khả năng có tính mùa.

Chuỗi thời gian có tồn tại tính mùa sẽ không có tính dừng. Phương pháp đơn giản nhất để khử tính mùa là lấy sai phân thứ m . Nếu chuỗi thời gian X_t có tính mùa với chu kỳ m thời đoạn thì chuỗi thời gian $Z_t = X_t - X_{t-m}$ sẽ không còn tính mùa. Đây cũng là ý tưởng của mô hình SARIMA (Seasonal Autoregressive Integrated Moving Average), một biến thể của mô hình ARIMA trong trường hợp dữ liệu chuỗi thời gian có tính mùa.

Trong thực tế giá trị m thường được xác định dựa trên dữ liệu. Giá trị m thường nhận các giá trị như $m = 12$ đối với các chuỗi thời gian có tính mùa với chu kỳ một năm hoặc $m = 4$ đối với các chuỗi thời gian có tính mùa với chu kỳ một quý.

3.1.3 Hàm tự tương quan và hàm tự tương quan riêng phần

Cho chuỗi thời gian dừng X_t , hàm hiệp phương sai γ_k của chuỗi thời gian X_t tại độ trễ k được định nghĩa như sau:

$$\gamma_k = \gamma(t, t+k) = cov(t, t+k) = E[(X_t - \mu_X)(X_{t+k} - \mu_X)] \quad (3.2)$$

Khi $k = 0$ thì giá trị của hàm hiệp phương sai tại độ trễ 0 chính là phương sai σ_X^2 của chuỗi thời gian X_t . $\gamma_0 = \sigma_X^2$

Hàm tự tương quan (autocorrelation function - ACF) ρ_k của chuỗi thời gian dừng X_t đo lường sự phụ thuộc tuyến tính giữa các cặp quan sát $X(t)$ và $X(t+k)$, ứng với từng độ trễ $k = 1, 2, \dots$. Với mỗi độ trễ k hàm tự tương quan tại độ trễ k được xác định thông qua độ lệch giữa các biến ngẫu nhiên $X(t)$ và $X(t+k)$ so với giá trị trung bình của chuỗi thời gian và được chuẩn hóa qua phương sai của chuỗi thời gian đó. Hàm tự tương quan tại các độ trễ k khác nhau sẽ có giá trị khác nhau.

$$\rho_k = \frac{\gamma_k}{\gamma_0} = \frac{E[(X_t - \mu_X)(X_{t+k} - \mu_X)]}{\sigma_X^2} \quad (3.3)$$

Về mặt lý thuyết, chuỗi dừng khi tất cả các $\rho_k = 0$ hay chỉ vài ρ_k khác 0.

Khi hàm tự tương quan giảm đột ngột, có nghĩa ρ_k rất lớn ở độ trễ $k = 1, 2$ và giảm nhanh về không ở các độ trễ tiếp theo, những ρ_k này được xem là những “đỉnh” và có nhiều ý nghĩa về mặt thống kê. Hầu hết các hàm tự tương quan sẽ giảm đột ngột sau độ trễ $k = 1, 2$. Hàm tự tương quan của chuỗi thời gian không dừng không giảm đột ngột mà trái lại giảm nhanh nhưng đều: không có đỉnh, chiều hướng này được gọi là “tắt dần”.

Song song với việc xác định hàm tự tương quan giữa các cặp quan sát $X(t)$ và $X(t + k)$, hàm tự tương quan từng phần (partial autocorrelation function - PACF) α_k tại các độ trễ $k = 1, 2, \dots$ cũng được xác định nhằm đo lường sự phụ thuộc tuyến tính giữa giá trị quan sát $X(t)$ và các giá trị trung gian trong khoảng giữa $X(t)$ và $X(t + k)$. Hàm tự tương quan từng phần tại các độ trễ k khác nhau sẽ có giá trị khác nhau. Hàm tự tương quan từng phần được định nghĩa như sau [6]:

$$\begin{aligned} \alpha_0 &= 1 \\ \text{và} \end{aligned} \quad (3.4)$$

$$\alpha_k = \phi_{kk}, k > 1$$

Trong đó:

+ ϕ_{kk} là thành phần cuối cùng của vector ϕ_k

$$\phi_k = \Gamma_k^{-1} \gamma_k \quad (3.4.1)$$

$$\Gamma_k = [\gamma_{i-j}]_{i,j=1}^k \quad (3.4.2)$$

$$\gamma_k = [\gamma_1, \gamma_2, \dots, \gamma_k]^T \quad (3.4.3)$$

Tóm lại, hàm tự tương quan ACF và hàm tự tương quan từng phần PACF của chuỗi thời gian có các đặc tính khác nhau. Hàm tự tương quan ACF đo mức độ phụ thuộc tuyến tính giữa các cặp quan sát trong khi hàm tự tương quan từng phần đo mức độ phụ thuộc tuyến tính từng phần.

3.1.4 Giới thiệu mô hình

Mô hình ARIMA là một mô hình thường được dùng để dự báo đối với dữ liệu tuyến tính, vì mô hình này là sự kết hợp của hai mô hình hồi quy (AR) và mô hình trung bình động (MA), cả hai mô hình này đều mô hình hóa cho sự biến đổi tuyến tính của dữ liệu (mô hình hồi quy mô hình hóa cho mối quan hệ tuyến tính giữa giá trị của chuỗi thời gian tại từng thời điểm so với giá trị trung bình của chuỗi thời gian, còn mô hình trung bình động mô hình hóa cho mối quan hệ tuyến tính giữa giá trị hiện tại của chuỗi thời gian với các giá trị ngẫu nhiên có phân phối chuẩn tại những thời điểm trong quá khứ) [4]. Do đó mà mô hình ARIMA được xem là mô hình phù hợp với dữ liệu tuyến tính.

Mô hình ARMA được giới thiệu trong chương 2 là mô hình dự báo dữ liệu chuỗi thời gian với giả thiết chuỗi thời gian có tính dừng. Tuy nhiên trong thực tế chuỗi thời gian thường không có tính dừng, tức là kết hợp (integrated). Do đó thường thì trước khi được dự báo, chuỗi thời gian không dừng phải được biến đổi về chuỗi thời gian dừng. Đây cũng là ý tưởng chính của mô hình ARIMA.

Mô hình tự hồi quy kết hợp trung bình động (Auto Regression Integrated Move Average - ARIMA) là mô hình cải tiến của mô hình ARMA áp dụng trong trường hợp chuỗi thời gian không có tính dừng. Một cách tổng quát, mô hình ARIMA với các tham số p, d, q được biểu diễn dưới dạng $\text{ARIMA}(p,d,q)$.

Trong đó:

+ p là bậc hồi quy được định nghĩa tương tự như mô hình ARMA.

+ q là bậc trung bình động được định nghĩa tương tự như mô hình ARMA.

+ d là số lần chuỗi thời gian phải được tính sai phân cho đến khi chuỗi thời gian có tính dừng.

Ví dụ: $\text{ARIMA}(2,1,3)$ là mô hình ARIMA có bậc hồi quy bằng 2, bậc trung bình động là 3 và được tính sai phân 1 lần.

Nếu chuỗi thời gian có tính dừng thì mô hình $\text{ARIMA}(p,d,q)$ sẽ trở thành $\text{ARIMA}(p,0,q) = \text{ARMA}(p,q)$. Nếu chuỗi thời gian được biểu diễn bởi mô hình

Chương 3. Mô hình kết hợp ARIMA và Support Vector Machine

ARIMA dưới dạng ARIMA($p, 0, 0$) thì chuỗi thời gian đó dừng và có AR(p) thuần túy. Nếu chuỗi thời gian được biểu diễn bởi mô hình ARIMA dưới dạng ARIMA($0, 0, q$) thì chuỗi thời gian đó dừng và có MA(q) thuần túy.

Vấn đề đặt ra là khi xem xét dự báo cho một chuỗi thời gian là làm thế nào để xác định chuỗi thời gian đó tuân theo quá trình nào, chuỗi thời gian có thể là một quá trình AR thuần túy (với bậc hồi quy p) hay một quá trình MA thuần túy (với bậc trung bình động q) hay một quá trình ARMA (với bậc hồi quy p và bậc trung bình động q) hay một quá trình ARIMA (với bậc hồi quy p và bậc trung bình động q và được tính sai phân d lần). Phương pháp Box – Jenkins (BJ) được đề xuất để giải quyết vấn đề này.

Phương pháp Box-Jenkins (BJ)

Phương pháp Box-Jenkins được đặt theo tên của hai nhà toán học, người đề xuất ra phương pháp này, đó là George Box và Gwilym Jenkins. Phương pháp Box-Jenkins xác định các bước cần thực hiện để xây dựng và dự báo dữ liệu chuỗi thời gian bằng mô hình ARIMA. Phương pháp này trải qua bốn bước:

+ **Bước 1: Nhận dạng mô hình.** Tức là thao tác để xác định các tham số p , d , q của mô hình ARIMA.

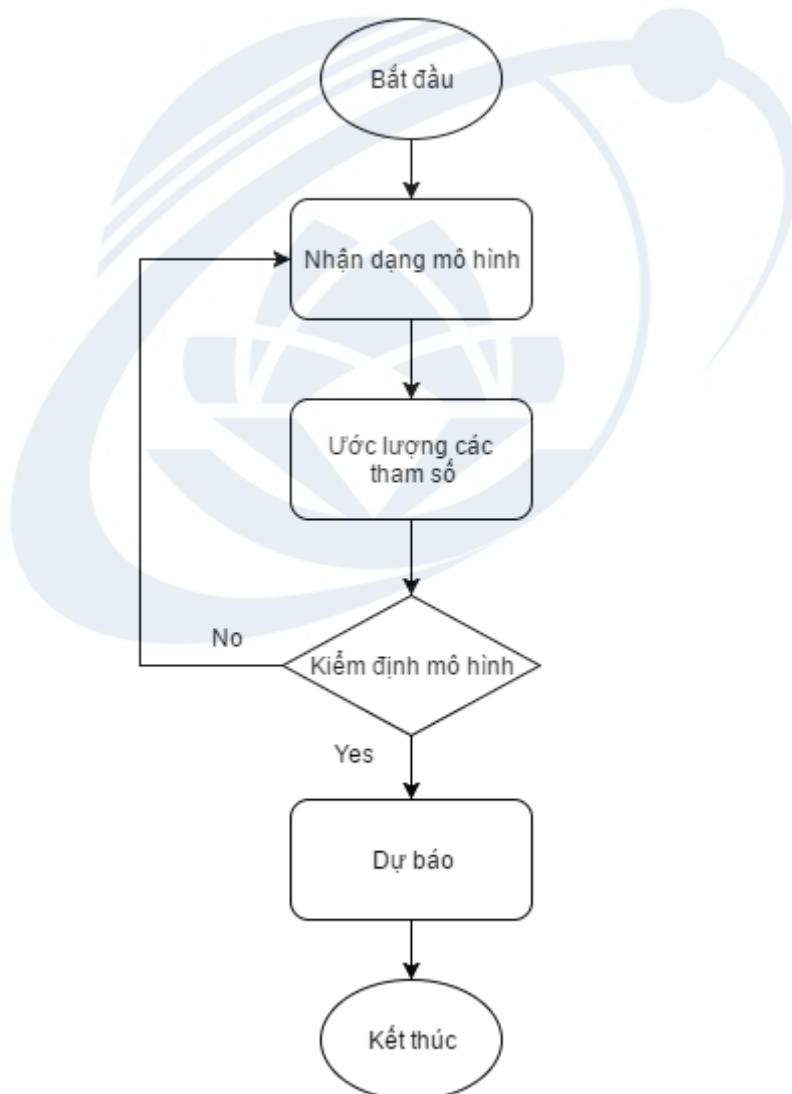
+ **Bước 2: Ước lượng các tham số.** Sau khi xác định các tham số p , q của mô hình ARIMA. Bước tiếp theo là ước lượng các số hạng tự hồi quy và các số hạng trung bình động. Thường thì các số hạng này được ước lượng bằng phương pháp bình phương tối thiểu. Tuy nhiên, hiện tại đã có nhiều phần mềm thống kê hỗ trợ giúp ước lượng các số hạng này.

+ **Bước 3: Kiểm định mô hình.** Sau khi đã lựa chọn một mô hình ARIMA cụ thể và ước lượng các tham số của nó, phương pháp Box – Jenkins yêu cầu kiểm định và đánh giá mô hình được lựa chọn có phù hợp với dữ liệu hay không, vì có thể có một mô hình ARIMA khác với các tham số p , d , q khác phù hợp hơn với dữ liệu.

Chương 3. Mô hình kết hợp ARIMA và Support Vector Machine

+ **Bước 4: Dự báo.** Sau khi chọn được một mô hình ARIMA phù hợp với dữ liệu. Mô hình đó được áp dụng vào dự báo giá trị tương lai của chuỗi thời gian dựa trên các giá trị hiện tại và trong quá khứ.

Mục tiêu của phương pháp Box-Jenkins là xác định và ước lượng một mô hình thống kê có thể mô hình hóa cho dữ liệu chuỗi thời gian mẫu dùng để xây dựng mô hình. Do đó phương pháp Box-Jenkins yêu cầu dữ liệu phải có tính dừng trước khi dùng để dự báo. Điều kiện ràng buộc này để các đặc điểm của mô hình sau khi được xác định không thay đổi theo thời gian, từ đó làm cơ sở cho việc dự báo giá trị tương lai.



Hình 3.3. Sơ đồ mô phỏng phương pháp Box-Jenkins

3.1.5 Nhận dạng mô hình

Bước này thực hiện để xác định các tham số p, d, q của mô hình ARIMA. Các tham số này được xác định bởi các phương pháp khác nhau.

Xác định tham số d : Tham số d của mô hình ARIMA biểu thị cho số lần tính sai phân của chuỗi thời gian cho đến khi chuỗi thời gian đó dừng. Phương pháp xác định chuỗi thời gian có phải là chuỗi thời gian dừng hay không được trình bày trong mục 3.1.1.

+ Trong trường hợp chuỗi thời gian dừng: Dĩ nhiên chuỗi thời gian sẽ không cần tính sai phân để biến đổi về chuỗi thời gian dừng nữa. Khi đó $d = 0$.

+ Ngược lại, trong trường hợp chuỗi thời gian không dừng: Khi đó chuỗi thời gian cần được tính sai phân cho đến khi được biến đổi về chuỗi thời gian dừng. Giả sử, chuỗi thời gian ban đầu X_t không dừng. Khi đó chuỗi thời gian $Y_t = X_t - X_{t-1}$ là chuỗi thời gian sau khi được tính sai phân 1 lần. Nếu Y_t không dừng thì chuỗi thời gian Y_t tiếp tục được dùng để tính sai phân lần 2 và chuỗi thời gian $Z_t = Y_t - Y_{t-1}$ là chuỗi thời gian sau khi được tính sai phân 2 lần. Quá trình này được lặp đi lặp lại cho đến khi chuỗi thời gian có tính dừng và d là số lần tính sai phân như vậy. Thường thì sau khi tính sai phân từ 1 đến 2 lần thì chuỗi thời gian không dừng ban đầu sẽ được biến đổi về dạng chuỗi thời gian dừng.

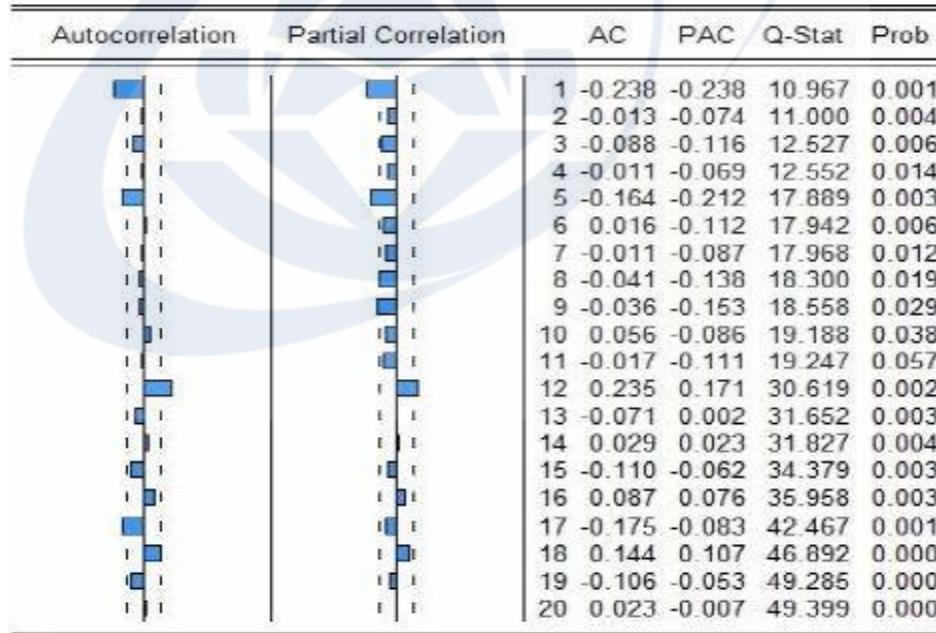
Xác định tham số p, q : Tham số p, q lần lượt là tham số dùng để xác định số lượng các số hạng hồi quy và các số hạng trung bình động. Để xác định giá trị của các tham số p, q phụ thuộc vào đồ thị của hàm tự tương quan (ACF) và hàm tự tương quan riêng phần (PACF). Chọn các giá trị của p tại các độ trễ mà tại đó giá trị của hàm PACF khác 0 về mặt thống kê. Tương tự chọn các giá trị của q tại các độ trễ mà tại đó giá trị của hàm ACF khác 0 về mặt thống kê. Các dạng lý thuyết của ACF và PACF được trình bày trong bảng sau:

Bảng 3.1. Các dạng lý thuyết của ACF và PACF

Loại mô hình	Dạng đồ thị ACF	Dạng đồ thị PACF
AR(p)	Giảm dần	Có đỉnh ở p và giảm về 0 sau p
MA(q)	Có đỉnh ở q và giảm về 0 sau q	Giảm dần
ARMA(p, q)	Giảm dần	Giảm dần

Chú thích: Bảng tóm tắt các dạng lý thuyết của ACF và PACF.

Hình 3.4 là ví dụ về đồ thị của hàm tự tương quan và hàm tự tương quan riêng phần. Ở các độ trễ 1, 5, 8, 9 và 12 giá trị của hàm PACF khác 0 theo nghĩa thống kê và giá trị liền sau đó bằng 0 theo nghĩa thống kê nên giá trị của p có thể nhận là 1, 5, 8, 9 và 12. Tương tự ở các độ trễ 1, 5, 12 và 17 giá trị của hàm ACF khác 0 theo nghĩa thống kê và giá trị liền sau đó bằng 0 theo nghĩa thống kê nên giá trị của q có thể nhận là 1, 5, 12 và 17.



Hình 3.4. Đồ thị hàm tự tương quan và hàm tự tương quan riêng phần

Nguồn: *Tương quan chuỗi biến động lượng khách du lịch quốc tế đến Việt Nam [1]*

3.1.6 Ước lượng các tham số

Sau khi nhận dạng xong các giá trị thích hợp của tham số p và q , bước tiếp theo là ước lượng các số hạng tự hồi quy và các số hạng trung bình động. Thường thì các số hạng này được ước lượng bằng phương pháp bình phương tối thiểu.

Phương pháp bình phương tối thiểu là phương pháp thường được sử dụng để ước lượng các tham số của phương trình hồi quy. Ý tưởng chính của phương pháp này là xác định các tham số của phương trình hồi quy sao cho khoảng cách giữa đồ thị của phương trình hồi quy đang cần xấp xỉ và đồ thị của phương trình hồi quy được tạo bởi các tham số là nhỏ nhất có thể. Khoảng cách này thường được tính bằng bình phương hiệu giữa hai giá trị là giá trị của phương trình hồi quy cần xấp xỉ và giá trị của phương trình hồi quy được tạo bởi các tham số. Chính vì vậy mà phương pháp này thường được gọi là phương pháp bình phương tối thiểu. Giá trị của các tham số hồi quy sao cho phương trình hồi quy đó xấp xỉ tốt nhất cho phương trình hồi quy đang xét, khi đó các giá trị này được xem như là tham số của phương trình hồi quy.

Ngày nay thao tác ước lượng các tham số thường được tính toán tự động bằng các phần mềm thống kê, nên trong phần này sẽ không đi sâu tìm hiểu về phương pháp ước lượng các tham số này. Bạn đọc quan tâm đến các phương pháp ước lượng tham số có thể tham khảo thêm ở các tài liệu chuyên về xác suất thống kê.

3.1.7 Kiểm định mô hình

Sau khi đã xác định được số lượng các tham số cũng như giá trị của từng tham số. Bước tiếp theo là kiểm tra mô hình được chọn có phù hợp với dữ liệu chuỗi thời gian dùng để xây dựng mô hình hay không. Để xác định một mô hình ARIMA có phù hợp với dữ liệu chuỗi thời gian hay không cần xác định phần còn lại hay phần chênh lệch giữa giá trị dự báo và giá trị thực tế có phải là chuỗi các giá trị ngẫu nhiên (hay còn là nhiễu trắng) hay không.

$$\varepsilon_t = X_t - \widehat{X}_t \quad (3.5)$$

Trong đó:

+ ε_t là nhiễu trắng

+ X_t là giá trị thực tế

+ \widehat{X}_t là giá trị dự báo

Để xác định ε_t có phải là chuỗi các giá trị ngẫu nhiên hay không cần quan sát đồ thị của hàm tự tương quan ACF và tự tương quan riêng phần PACF của nó. Nếu đồ thị của hai hàm này không có các đỉnh khác 0 về mặt thống kê thì xem như ε_t là các giá trị ngẫu nhiên thuận túy.

Trong trường hợp có nhiều hơn một mô hình ARIMA phù hợp với dữ liệu chuỗi thời gian. Khi đó cần chọn một mô hình ARIMA phù hợp nhất với dữ liệu chuỗi thời gian để dự báo. Sử dụng các tiêu chuẩn BIC (Bayesian information criterion), tiêu chuẩn AIC (Akaike info criterion) hay ước lượng sai số chuẩn (Standard error of estimate - SEE) để chọn một mô hình ARIMA phù hợp nhất với dữ liệu. Mô hình ARIMA nào có các giá trị này bé nhất thì mô hình ARIMA đó được chọn để làm mô hình cho dự báo.

3.1.8 Dự báo

Sau khi chọn được một mô hình ARIMA phù hợp nhất với dữ liệu chuỗi thời gian. Giá trị dự báo của chuỗi thời gian được tính toán dựa trên mô hình ARIMA và giá trị của chuỗi thời gian đó.

Đối với chuỗi thời gian X_t có tính dừng, giá trị dự báo \widehat{X}_t tại thời điểm t cũng chính là giá trị dự báo của chuỗi thời gian tại thời điểm t . Ngược lại trong trường hợp chuỗi thời gian X_t không có tính dừng và giả sử chuỗi thời gian Y_t là chuỗi thời gian X_t sau khi tính sai phân 1 lần, Y_t có tính dừng. Khi đó giá trị dự báo \widehat{Y}_t tại thời điểm t chưa phải là giá trị dự báo của chuỗi thời gian X_t ban đầu, mà giá trị dự báo của chuỗi thời gian ban đầu $\widehat{X}_t = X_{t-1} + \widehat{Y}_t$

Một trong số các lý do về tính phổ biến của phương pháp dự báo sử dụng mô hình ARIMA là thành công của nó trong dự báo dữ liệu chuỗi thời gian.

Chương 3. Mô hình kết hợp ARIMA và Support Vector Machine

Trong một số trường hợp dự báo thu được từ phương pháp này có tính tin cậy cao hơn so với các dự báo thu được từ các phương pháp lập mô hình kinh tế lượng truyền thống khác, đặc biệt là đối với các dự báo ngắn hạn. Tuy nhiên do dữ liệu luôn biến động nên mô hình ARIMA cần được cập nhật thường xuyên để đảm bảo tính chính xác của kết quả dự báo.

3.2 Support Vector Machine

Support Vector Machine (SVM) là một thuật giải quan trọng và được biết đến nhiều trong lĩnh vực máy học. Không giống như mô hình ARIMA, thuật giải SVM không chỉ được ứng dụng riêng trong lĩnh vực dự báo dữ liệu chuỗi thời gian, mà SVM còn được ứng dụng rộng rãi trong rất nhiều lĩnh vực khác của khoa học máy tính như trong các bài toán về nhận diện hay trong các bài toán về phân lớp, gom cụm dữ liệu,... Trong mục này sẽ giới thiệu về thuật giải SVM và ứng dụng của nó trong lĩnh vực dự báo dữ liệu chuỗi thời gian.

3.2.1 Giới thiệu

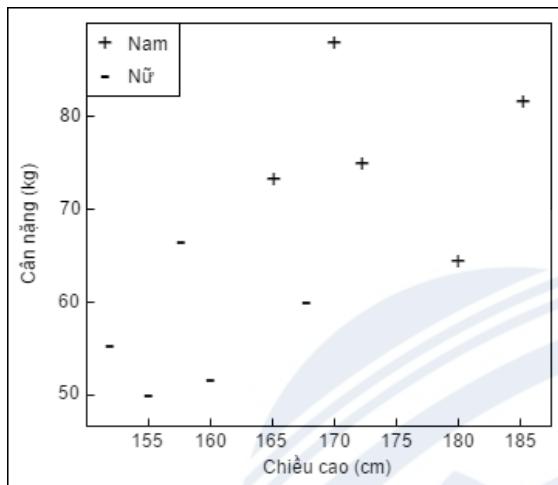
Thuật giải SVM ban đầu được phát minh bởi Nhà toán học Liên Xô, Vladimir N. Vapnik và Alexey Ya. Chervonenkis vào năm 1963. Đến năm 1992, Bernhard E. Boser, Isabelle M. Guyon và Vladimir N. Vapnik đề xuất phương pháp tạo các phân lớp không tuyến tính bằng cách áp dụng kernel trick để làm tối đa khoảng cách từ các phân lớp đến siêu phẳng (maximum-margin hyperplanes).

Ban đầu thuật giải SVM được đề xuất để giải quyết bài toán phân lớp, do đó mục tiêu của thuật giải SVM là tìm một siêu phẳng (hyperlane) tối ưu để phân lớp dữ liệu sao cho tối đa khoảng cách (margin) giữa các lớp dữ liệu. Do SVM được xây dựng và hoàn thiện từ dữ liệu nên thuật giải SVM được xem như là một thuật giải học có giám sát (supervised learning algorithm) [28]. Trong mục tiêu của thuật giải SVM có hai khái niệm cần làm rõ đó là siêu phẳng (hyperlane) và khoảng cách (margin). Để tìm hiểu hai khái niệm này ta xem xét ví dụ sau.

Hình 3.5 là một ví dụ về một bài toán phân lớp đơn giản giữa 2 lớp (Men và Women), phân lớp này dựa trên chiều cao (cm) và cân nặng (kg) của một

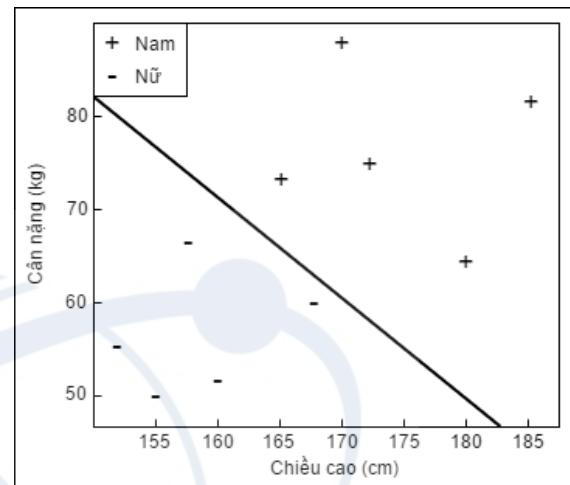
Chương 3. Mô hình kết hợp ARIMA và Support Vector Machine

người để xác định. Vấn đề đặt ra cho bài toán phân lớp này là giả sử cho biết chiều cao và cân nặng của một người, hỏi người đó là Men hay Women. Để trả lời cho câu hỏi đó, chúng ta cần tìm một đường thẳng sao cho có thể phân chia rõ ràng 2 lớp Men và Women, tức là mỗi lớp nằm trọn trên một phía của đường thẳng này, hình 3.6 là ví dụ về một đường thẳng phân lớp.



Hình 3.5. Bài toán phân lớp

Nguồn: *Ví dụ bài toán phân lớp* [23]



Hình 3.6. Đường thẳng phân lớp

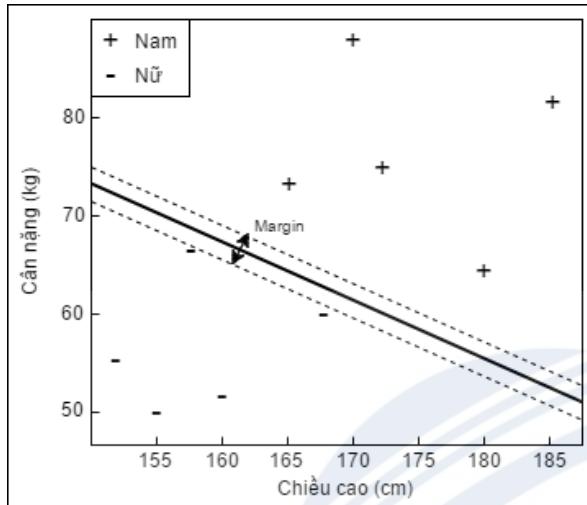
Nguồn: *Ví dụ đường thẳng phân lớp* [23]

Ví dụ trên là một ví dụ đơn giản cho bài toán phân lớp trong không gian hai chiều (cân nặng, chiều cao). Trong thực tế để phân lớp cần xét trên nhiều yếu tố khác nhau của đối tượng, do đó mà số chiều trong bài toán phân lớp có thể lớn hơn hai chiều. Chính vì vậy mà khái niệm siêu phẳng (hyperlane) được đề xuất để mô tả cho một “mặt phẳng” giúp phân chia các lớp đối tượng. Siêu phẳng trong không gian một chiều được xem như là một điểm, trong không gian hai chiều là một đường thẳng, trong không gian ba chiều là một mặt phẳng, từ không gian bốn chiều trở lên ta gọi chung đó là một siêu phẳng.

Khoảng cách (margin) là phần không gian có độ lớn bằng hai lần khoảng cách từ siêu phẳng đến điểm dữ liệu gần siêu phẳng nhất. Hay nói cách khác là không có một điểm dữ liệu nào nằm trong margin. Hình 3.7 mô tả về margin của một đường thẳng phân lớp, có thể thấy trong hình này margin là phần không gian

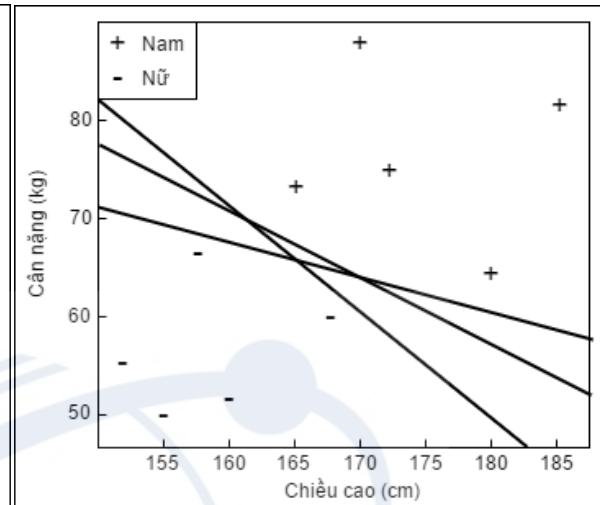
Chương 3. Mô hình kết hợp ARIMA và Support Vector Machine

bị giới hạn bởi hai đường thẳng và trong margin không có bất kỳ một điểm dữ liệu nào.



Hình 3.7. Khoảng cách trong phân lớp

Nguồn: Ví dụ khoảng cách phân lớp [23]



Hình 3.8. Các đường thẳng phân lớp

Nguồn: Ví dụ các đường thẳng phân lớp [23]

Hình 3.8 là ví dụ về một số đường thẳng phân lớp, có thể thấy đối với một bài toán phân lớp dữ liệu có thể có nhiều đường thẳng (siêu phẳng) để phân lớp dữ liệu, do đó mà mục tiêu của thuật giải SVM là tìm một siêu phẳng tối ưu để phân lớp dữ liệu. Siêu phẳng tối ưu (optimal hyperlane) được định nghĩa là một siêu phẳng có margin lớn nhất. Vậy làm thế nào để tìm được một siêu phẳng có margin lớn nhất. Để trả lời cho câu hỏi đó ta cần tìm hiểu về cách tính độ rộng của margin.

3.2.2 Độ rộng của margin

Trước khi tính độ rộng của margin, ta xem xét biểu diễn toán học của một siêu phẳng. Tổng quát, siêu phẳng được biểu diễn dưới dạng sau [28]:

$$w^T x = 0 \quad (3.6)$$

Trong đó:

+ w, x là hai vector bất kỳ

Chương 3. Mô hình kết hợp ARIMA và Support Vector Machine

Ta biết rằng trong không gian 2 chiều, một đường thẳng được biểu diễn dưới dạng $y = ax + b$. Mà siêu phẳng trong không gian 2 chiều chính là một đường thẳng, vậy làm thế nào để biểu diễn phương trình đường thẳng dưới dạng phương trình siêu phẳng. Ta có phương trình đường thẳng

$$y = ax + b \quad (3.7)$$

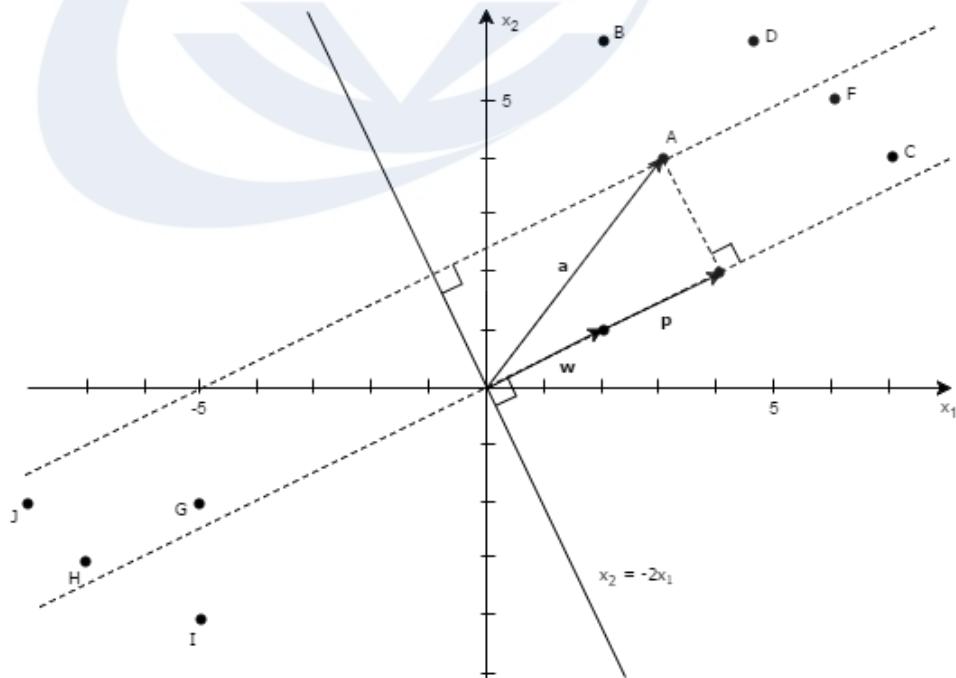
Suy ra: $y - ax - b = 0 \quad (3.7.1)$

Cho $w = \begin{pmatrix} -b \\ -a \\ 1 \end{pmatrix}$ và $x = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$, khi đó phương trình siêu phẳng:

$$w^T x = -b \times 1 + (-a) \times x + 1 \times y$$

Suy ra: $w^T x = y - ax - b \quad (3.8)$

Mặc dù cách biểu diễn phương trình đường thẳng và phương trình siêu phẳng là khác nhau, tuy nhiên về ý nghĩa là như nhau. Có hai lý do mà thuật giải SVM chọn biểu diễn một siêu phẳng dưới dạng (3.6). Thứ nhất cách biểu diễn này phù hợp và dễ dàng tính toán trong không gian nhiều chiều. Thứ hai \vec{w} luôn là một biểu diễn đơn giản cho siêu phẳng.



Hình 3.9. Ví dụ về tính độ rộng của margin

Nguồn: *Ví dụ về tính độ rộng của margin* [23]

Để tính độ rộng của margin ta xét một ví dụ cụ thể sau, hình 3.9 minh họa cho ví dụ. Trong không gian 2 chiều, cho một tập dữ liệu với mỗi điểm dữ liệu được xác định bởi 2 yếu tố (x_1, x_2) . Một siêu phẳng phân chia tập dữ liệu này

thành 2 lớp với $w = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}$ và $x = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}$, vì $w_0 = 0$ nên ta có thể biểu diễn lại

$w = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ và $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, khi đó phương trình của siêu phẳng:

$$w^T x = 2x_1 + x_2 = 0$$

$$\text{Suy ra: } x_2 = -2x_1 \quad (3.9)$$

Điểm dữ liệu $A(3, 4)$ là điểm dữ liệu gần nhất với siêu phẳng, khoảng cách từ A đến siêu phẳng được tính như sau.

Kẻ đường thẳng đi qua \vec{w} , khoảng cách từ điểm A đến siêu phẳng bằng độ lớn của \vec{p} , với \vec{p} nằm trên đường thẳng đi qua \vec{w} và có điểm gốc là gốc tọa độ, đỉnh là hình chiếu của điểm A trên đường thẳng đó. Gọi θ là góc được tạo bởi \vec{a} và \vec{p} , khi đó:

$$\cos \theta = \frac{\|\vec{p}\|}{\|\vec{a}\|}$$

$$\text{Suy ra: } \|\vec{p}\| = \|\vec{a}\| \cos \theta \quad (3.10)$$

Mặt khác, vì \vec{w} và \vec{p} cùng nằm trên một đường thẳng nên θ cũng là góc tạo bởi \vec{w} và \vec{a} , xét tích vô hướng của \vec{w} và \vec{a}

$$\vec{w} \cdot \vec{a} = \|\vec{w}\| \|\vec{a}\| \cos \theta$$

$$\text{Suy ra: } \cos \theta = \frac{\vec{w} \cdot \vec{a}}{\|\vec{w}\| \|\vec{a}\|} \quad (3.11)$$

Thay (3.11) vào (3.10) ta có:

$$\|\vec{p}\| = \|\vec{a}\| \frac{\vec{w} \cdot \vec{a}}{\|\vec{w}\| \|\vec{a}\|} = \frac{\vec{w} \cdot \vec{a}}{\|\vec{w}\|} \quad (3.11)$$

Gọi \vec{u} là vector đơn vị của \vec{w} , khi đó theo tính chất của vector đơn vị thì:

$$\vec{u} = \frac{\vec{w}}{\|\vec{w}\|} \quad (3.12)$$

Thay (3.12) vào (3.11) ta có:

$$\|\vec{p}\| = \vec{u} \cdot \vec{a} \quad (3.13)$$

Trong ví dụ này, $\vec{a} = (3, 4); \vec{w} = (2, 1); \|\vec{w}\| = \sqrt{2^2 + 1} = \sqrt{5}$ và $\vec{u} = \frac{\vec{w}}{\|\vec{w}\|} = \left(\frac{2}{\sqrt{5}}, \frac{1}{\sqrt{5}}\right)$, do đó:

$$\|\vec{p}\| = \vec{u} \cdot \vec{a} = 3 \times \frac{2}{\sqrt{5}} + 4 \times \frac{1}{\sqrt{5}} = 2\sqrt{5}$$

Vì độ rộng margin gấp 2 lần khoảng cách từ điểm dữ liệu gần nhất đến siêu phẳng nên trong trường hợp này độ rộng margin của siêu phẳng là $4\sqrt{5}$.

Trong ví dụ trên siêu phẳng $x_2 = -2x_1$ là một siêu phẳng phân lớp dữ liệu tốt, nhưng không phải là một siêu phẳng tối ưu, siêu phẳng có margin lớn nhất. Như trong hình 3.10, siêu phẳng bên dưới siêu phẳng $x_2 = -2x_1$ với margin M_2 mới là siêu phẳng tối ưu. Vấn đề đặt ra cho thuật giải SVM là làm thế nào để tìm được một siêu phẳng tối ưu cho phân lớp dữ liệu.

3.2.3 Tìm kiếm siêu phẳng tối ưu

Vấn đề tìm kiếm một siêu phẳng tối ưu tương đương với việc tìm kiếm một siêu phẳng với margin lớn nhất. Để tìm kiếm một siêu phẳng có margin lớn nhất cần tìm 2 siêu phẳng có vai trò như “biên” của margin sao cho khoảng cách giữa 2 siêu phẳng đó là lớn nhất. Dĩ nhiên giữa 2 “biên” này sẽ không có điểm dữ liệu nào. Trong hình 3.10, siêu phẳng đi qua điểm G và siêu phẳng đi qua điểm A và B lần lượt là 2 “biên” của siêu phẳng tối ưu.

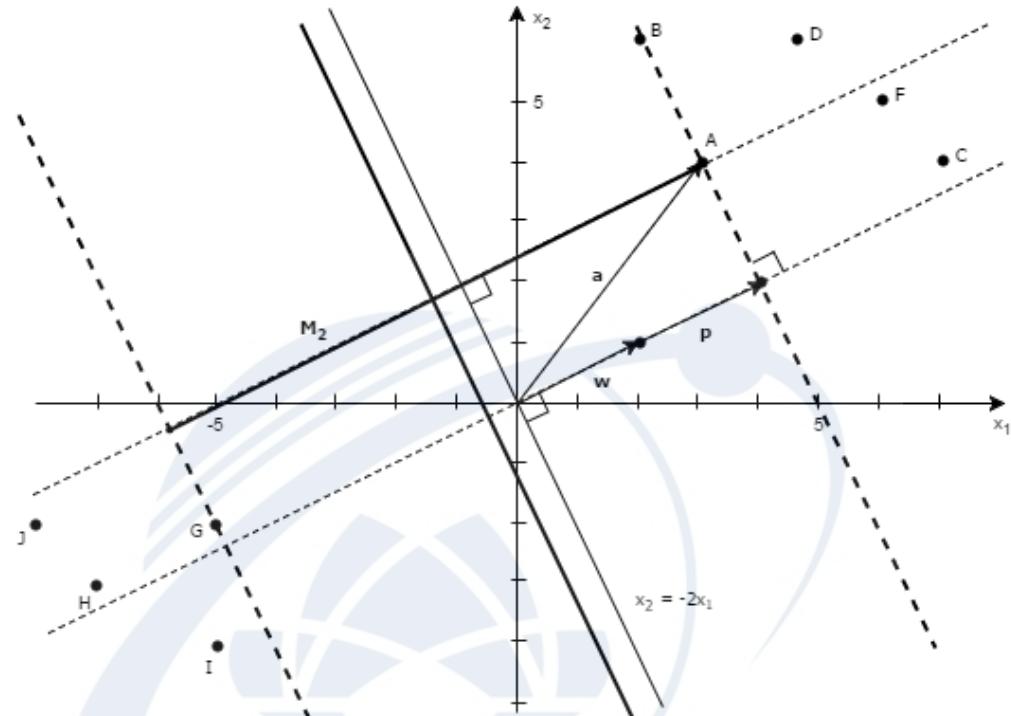
Các bước tìm kiếm siêu phẳng tối ưu:

+ Bước 1: *Mô hình hóa dữ liệu và phân lớp*

Một cách tổng quát, giả sử tập dữ liệu cần được phân lớp là \mathcal{D} bao gồm n cặp giá trị (x_i, y_i) , trong đó x_i là vector p chiều, mỗi chiều đại diện cho một thuộc

tính của đối tượng. y_i là giá trị phân lớp của đối tượng, giả sử có 2 phân lớp là +1 và -1. Khi đó \mathcal{D} được định nghĩa như sau [28]:

$$\mathcal{D} = \{(x_i, y_i) \mid x_i \in \mathbb{R}^p, y_i \in \{-1, 1\}, i = \overline{1, n}\}$$



Hình 3.10. Siêu phẳng tối ưu

Nguồn: *Siêu phẳng tối ưu* [23]

+ Bước 2: *Tìm “biên” của margin*

Margin bị giới hạn bởi 2 “biên”, “biên” của margin thực chất là một siêu phẳng, do đó để tìm margin lớn nhất ta sẽ tìm 2 siêu phẳng sao cho không có đối tượng dữ liệu nào nằm bên trong chúng và khoảng cách giữa 2 siêu phẳng này là lớn nhất.

Trong không gian 2 chiều, một siêu phẳng H_0 có thể biểu diễn dưới dạng

$$\vec{w} \cdot \vec{x} + b = 0 \quad (3.14)$$

Trong đó $\vec{w} = (-a, 1)$, $\vec{x} = (x, y)$, b là một số bất kỳ gọi là bias. Khi đó 2 “biên” là 2 siêu phẳng H_1, H_2 sao cho:

$$H_1: \vec{w} \cdot \vec{x} + b = 1 \quad (3.15)$$

$$H_2: \vec{w} \cdot \vec{x} + b = -1 \quad (3.16)$$

Một siêu phẳng bất kỳ thỏa điều kiện để trở thành “biên” của margin khi và chỉ khi siêu phẳng đó thỏa 1 trong 2 ràng buộc sau với mọi đối tượng x_i

$$\vec{w} \cdot \vec{x}_i + b \geq 1 \text{ với những } x_i \text{ thuộc phân lớp 1} \quad (3.17)$$

hoặc

$$\vec{w} \cdot \vec{x}_i + b \leq -1 \text{ với những } x_i \text{ thuộc phân lớp -1} \quad (3.18)$$

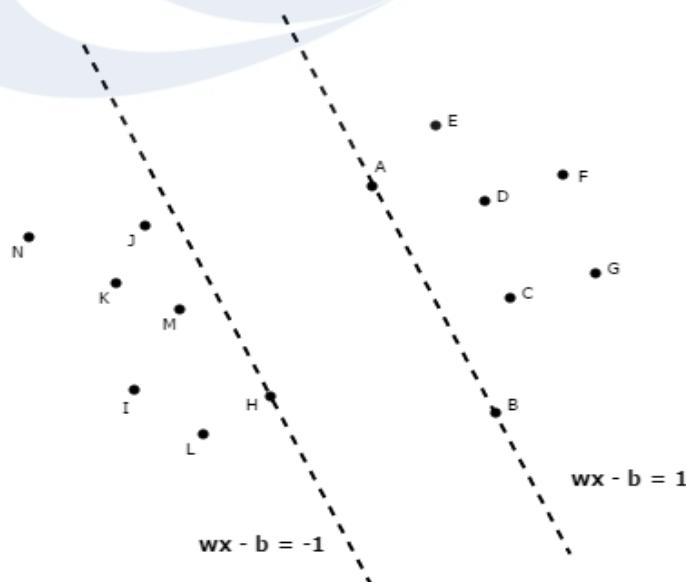
Xét bất phương trình (3.18), mọi đối tượng x_i thỏa bất phương trình này đều thuộc vào lớp -1, tức là $y_i = -1$. Nhân 2 vế của bất phương trình (3.18) với y_i ta có bất phương trình sau:

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 \text{ với những } x_i \text{ thuộc phân lớp -1} \quad (3.19)$$

Xét bất phương trình (3.17), mọi đối tượng x_i thỏa bất phương trình này đều thuộc vào lớp 1, tức là $y_i = 1$. Nhân 2 vế của bất phương trình (3.17) với y_i ta có bất phương trình sau:

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 \text{ với những } x_i \text{ thuộc phân lớp 1} \quad (3.20)$$

Vì (3.19) và (3.20) là như nhau nên có thể tổng hợp hai phương trình này lại thành:



Hình 3.11. Hai biên của margin

Nguồn: *Hai biên của margin* [23]

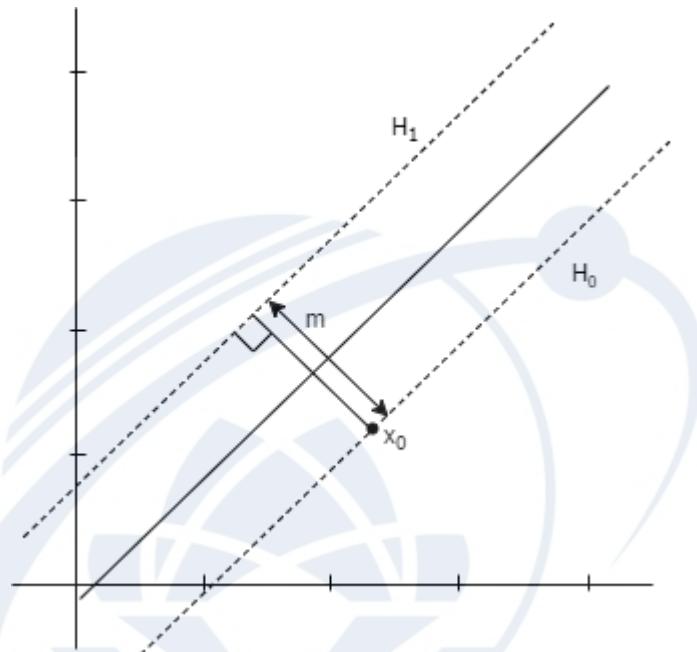
$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1, \forall i \ 1 \leq i \leq n \quad (3.21)$$

+ Bước 3: *Tối đa khoảng cách giữa hai “biên”*

Cho hai siêu phẳng H_0, H_1 với phương trình:

$$H_0: \vec{w} \cdot \vec{x} + b = -1 \quad (3.22)$$

$$H_1: \vec{w} \cdot \vec{x} + b = 1 \quad (3.23)$$



Hình 3.12. Khoảng cách giữa hai siêu phẳng

Nguồn: *Khoảng cách giữa hai siêu phẳng* [23]

Với x_0 là một điểm nằm trên siêu phẳng H_0 , khi đó khoảng cách giữa hai siêu phẳng H_0, H_1 là khoảng cách từ x_0 đến H_1 , khoảng cách này ký hiệu là m như trong hình 3.12.

Để tìm giá trị của m , trước tiên ta xét \vec{w} , vì $H_1: \vec{w} \cdot \vec{x} + b = 1$ nên \vec{w} vuông góc với H_1 (hình 3.13a). Với \vec{u} là vector đơn vị của \vec{w} , $\|\vec{u}\| = 1$ (hình 3.13b), gọi $\vec{k} = m\vec{u} = m \frac{\vec{w}}{\|\vec{w}\|}$ khi đó $\|\vec{k}\| = m\|\vec{u}\| = m$ và \vec{k} cũng vuông góc với H_1 vì \vec{k} có cùng hướng với \vec{u} (hình 3.13c).

Do ta chỉ quan tâm đến độ lớn của \vec{k} nên để đơn giản cho việc tính toán ta chuyển \vec{k} về dạng như trong hình 3.13d, với gốc là điểm x_0 nằm trên siêu phẳng H_0 và đỉnh là điểm z_0 nằm trên siêu phẳng H_1 . Khi đó:

$$\vec{z}_0 = \vec{x}_0 + \vec{k} \quad (3.24)$$

Vì điểm z_0 nằm trên siêu phẳng H_1 nên

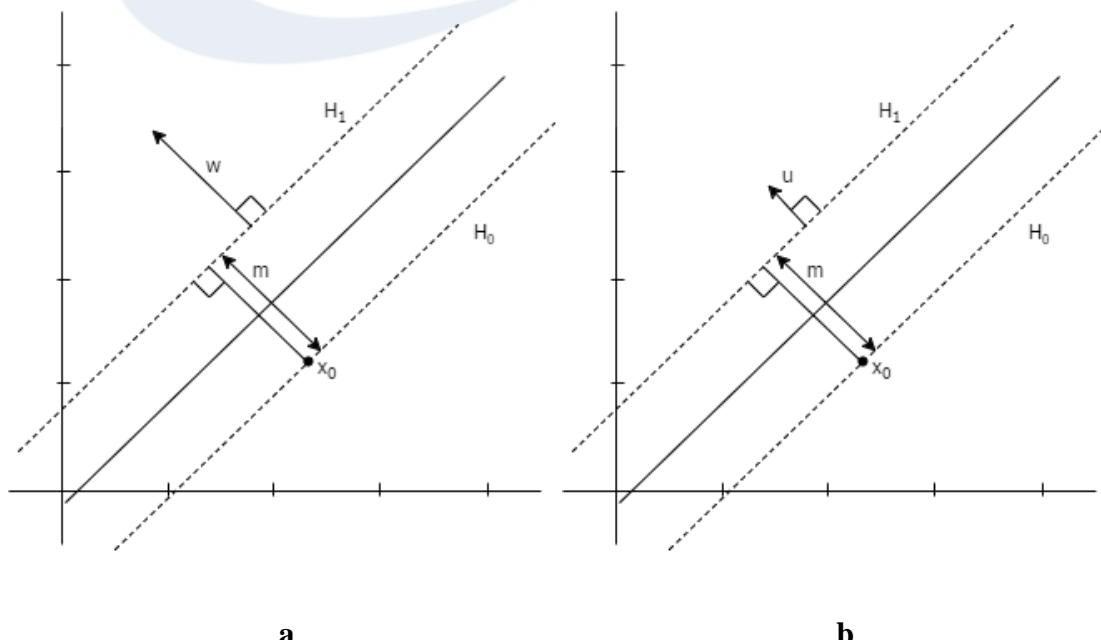
$$\vec{w} \cdot \vec{z}_0 + b = 1 \quad (3.25)$$

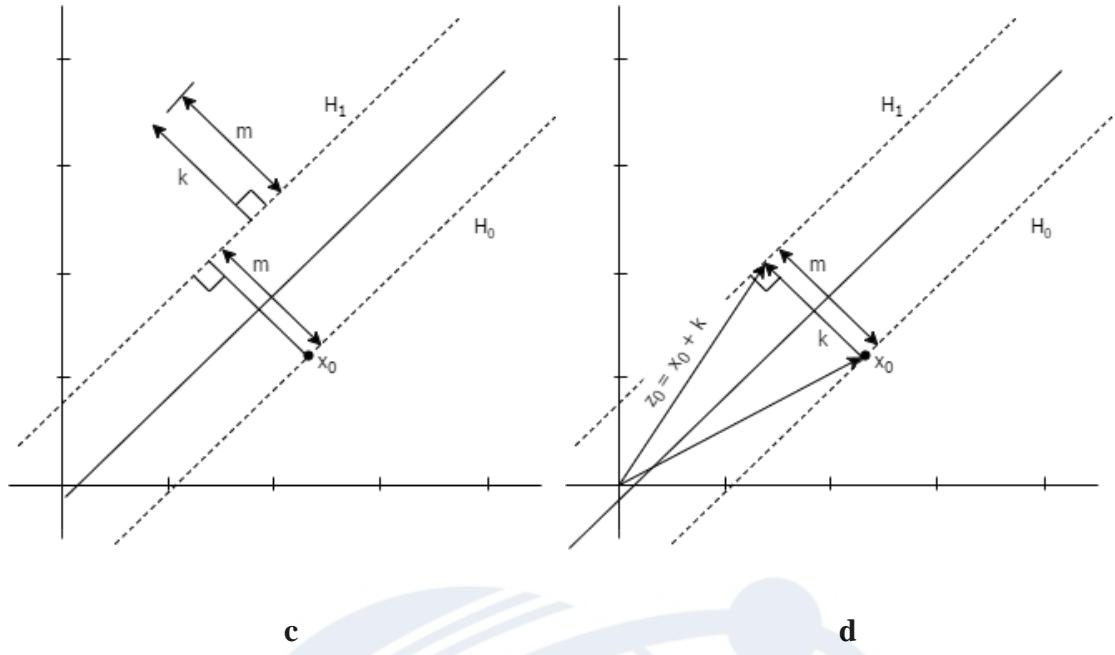
Thay (3.24) vào (3.25) ta có:

$$\begin{aligned} & \vec{w} \cdot (\vec{x}_0 + \vec{k}) + b = 1 \\ \Leftrightarrow & \vec{w} \cdot \left(\vec{x}_0 + m \frac{\vec{w}}{\|\vec{w}\|} \right) + b = 1 \\ \Leftrightarrow & \vec{w} \cdot \vec{x}_0 + m \frac{\|\vec{w}\|^2}{\|\vec{w}\|} + b = 1 \\ \Leftrightarrow & \vec{w} \cdot \vec{x}_0 + m \|\vec{w}\| + b = 1 \\ \Leftrightarrow & \vec{w} \cdot \vec{x}_0 + b = 1 - m \|\vec{w}\| \end{aligned} \quad (3.26)$$

Do x_0 nằm trên siêu phẳng H_0 nên $\vec{w} \cdot \vec{x}_0 + b = -1$, thay vào phương trình (3.26) ta có:

$$\begin{aligned} -1 &= 1 - m \|\vec{w}\| \\ m \|\vec{w}\| &= 2 \\ m &= \frac{2}{\|\vec{w}\|} \end{aligned} \quad (3.27)$$





Hình 3.13.

- a) \vec{w} của siêu phẳng
- b) \vec{u} vector đơn vị của \vec{w}
- c) \vec{k}
- d) \vec{k} sau khi chuyển đổi

Nguồn: Ví dụ tính khoảng cách giữa hai siêu phẳng [28]

Từ phương trình (3.27) ta có thể thấy giá trị m càng lớn khi và chỉ khi độ lớn của \vec{w} càng bé. Vậy để chọn được một siêu phẳng tối ưu có margin lớn nhất ta sẽ chọn siêu phẳng có \vec{w} với độ lớn bé nhất. Vấn đề tìm siêu phẳng tối ưu được biểu diễn dưới dạng bài toán tối ưu như sau:

$$\begin{aligned} & \text{Maximize } \frac{1}{\|\vec{w}\|} \\ & \text{subject to } y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 \\ & \quad (\forall i \ 1 \leq i \leq n) \end{aligned} \tag{3.28}$$

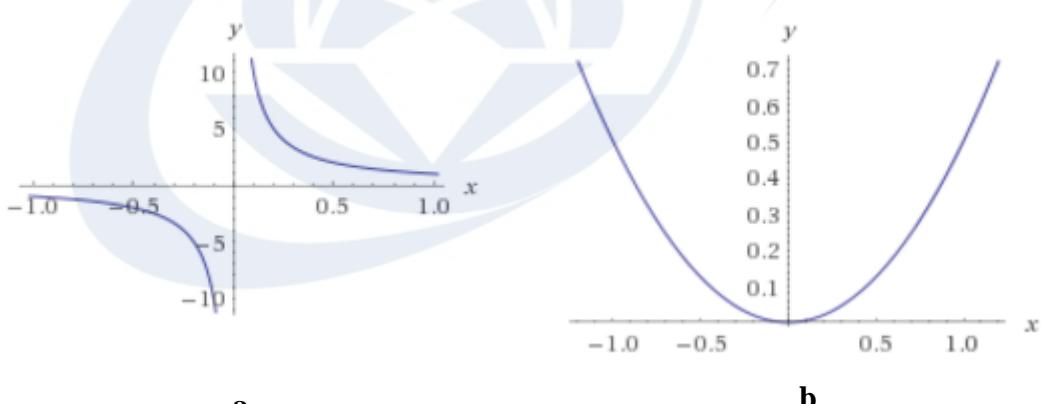
Chú ý, trong bài toán tối ưu kết quả $\max \frac{1}{\|\vec{w}\|}$ và $\max \frac{2}{\|\vec{w}\|}$ là tương đương với nhau, nên ta có thể sử dụng hàm $\frac{1}{\|\vec{w}\|}$ để thay thế cho hàm $\frac{2}{\|\vec{w}\|}$ trong bài toán (3.28). Hình 3.14a là đồ thị của hàm số $y = \frac{1}{x}$, vì m chỉ nhận giá trị lớn

Chương 3. Mô hình kết hợp ARIMA và Support Vector Machine

hơn hoặc bằng 0 nên ta chỉ xem xét phần đồ thị của hàm số $y = \frac{1}{x}$ ở góc phần tư thứ I, từ đó thị này có thể thấy giá trị của hàm số này *maximize* khi giá trị của x gần về 0. Mặt khác, hình 3.14b là đồ thị của hàm $y = \frac{x^2}{2}$, hàm này *minimize* khi $x = 0$. Do đó, trong bài toán tối ưu kết quả *maximize* $\frac{1}{\|\vec{w}\|}$ và *minimize* $y = \frac{x^2}{2}$ là tương đương với nhau. Từ nhận xét trên bài toán tối ưu (3.28) tương đương với bài toán tối ưu sau:

$$\begin{aligned} & \text{Minimize } \frac{\|\vec{w}\|^2}{2} \\ & \text{subject to } y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 \\ & \quad (\forall i \ 1 \leq i \leq n) \end{aligned} \tag{3.29}$$

Lý do ta chọn hàm $y = \frac{x^2}{2}$ thay thế cho hàm $y = \frac{1}{x}$ trong bài toán tối ưu vì hàm $y = \frac{x^2}{2}$ là một hàm lồi và là một hàm liên tục, bên cạnh đó việc tìm *minimize* của hàm này cũng dễ dàng hơn.



Hình 3.14.

- a) Đồ thị hàm $y = \frac{1}{x}$
- b) Đồ thị hàm $y = \frac{x^2}{2}$

Sau khi mô hình hóa vấn đề tìm kiếm một siêu phẳng tối ưu dưới dạng một bài toán tối ưu của một hàm số, việc tiếp theo ta cần phải thực hiện đó là giải bài

toán tối ưu này để tìm các giá trị thỏa mãn các ràng buộc. Trong toán học tối ưu, để giải một bài toán tối ưu có thể áp dụng một số phương pháp như tìm cực trị của hàm số bằng đạo hàm cấp 1 và tính giá trị đạo hàm cấp 2 tại các cực trị để xác định đó là cực tiểu hoặc cực đại, sau đó so sánh các giá trị của hàm số tại cực tiểu hoặc cực đại này lại với nhau để xác định giá trị nhỏ nhất hoặc lớn nhất của hàm số. Tuy nhiên, phương pháp này chỉ có thể áp dụng tốt trong trường hợp bài toán tối ưu không bị giới hạn bởi các ràng buộc (constraints), cụ thể là giá trị của các biến không bị giới hạn bởi các điều kiện. Trong trường hợp bài toán tối ưu bị giới hạn bởi các ràng buộc như trong bài toán tối ưu (3.29) ta cần sử dụng một phương pháp khác để giải bài toán này, phương pháp đó được gọi là Lagrange multipliers. Phần tiếp theo sẽ trình bày về cách áp dụng phương pháp Lagrange multipliers trong giải bài toán tối ưu để tìm kết quả cho bài toán tối ưu (3.29).

3.2.4 Phương pháp Lagrange multipliers

Phương pháp Lagrange multipliers được đặt theo tên của Nhà toán học người Ý, Joseph Louis Lagrange, người đề xuất ra phương pháp này. Trong toán học tối ưu, phương pháp Lagrange multipliers là một chiến lược tìm kiếm giá trị lớn nhất và giá trị bé nhất cục bộ của một hàm số sao cho thỏa mãn các điều kiện ràng buộc.

Tổng quát, một bài toán tối ưu được trình bày như sau:

$$\begin{aligned} & \text{Maximize / Minimize } f(x_1, x_2, \dots, x_n) \\ & \text{subject to } g(x_1, x_2, \dots, x_n) = c \end{aligned} \tag{3.30}$$

Trong đó:

+ $f(x_1, x_2, \dots, x_n): \mathbb{R}^n \rightarrow \mathbb{R}$ là hàm số cần tìm giá trị lớn nhất hoặc giá trị bé nhất.

+ $g(x_1, x_2, \dots, x_n): \mathbb{R}^n \rightarrow \mathbb{R}$ là ràng buộc (constraint)

+ f và g là các hàm liên tục và khả vi từng phần (có đạo hàm riêng).

Khi đó hàm Lagrange (Lagrange function hoặc Lagrangian) được định nghĩa là:

$$\mathcal{L}(x_1, x_2, \dots, x_n, \lambda) = f(x_1, x_2, \dots, x_n) - \lambda \cdot [g(x_1, x_2, \dots, x_n) - c] \quad (3.31)$$

Trong đó λ được gọi là Lagrange multiplier, nếu tồn tại điểm $(x_1^0, x_2^0, \dots, x_n^0, \lambda_0)$ sao cho đạo hàm riêng của \mathcal{L} tại điểm $(x_1^0, x_2^0, \dots, x_n^0, \lambda_0)$ bằng 0 thì điểm đó được gọi là điểm dừng (stationary point) của hàm Lagrange. Một điểm bất kỳ có giá trị lớn nhất hoặc giá trị bé nhất của hàm f là một điểm dừng, tuy nhiên không phải mọi điểm dừng đều là điểm mà tại đó hàm f có giá trị lớn nhất hoặc bé nhất. Do đó, giải phương trình đạo hàm riêng Lagrange cho ta điều kiện cần để xác định một điểm có phải là một cực trị hay không, để xác định đâu là điểm có giá trị lớn nhất hoặc giá trị bé nhất của hàm f ta phải tính giá trị của hàm này tại các điểm cực trị đó.

Đạo hàm riêng của hàm Lagrange được ký hiệu là $\nabla \mathcal{L}(x_1, x_2, \dots, x_n, \lambda) = \nabla f(x_1, x_2, \dots, x_n) - \lambda \cdot \nabla g(x_1, x_2, \dots, x_n)$. Để hiểu hơn về phương pháp Lagrange multipliers ta xét một ví dụ về bài toán tối ưu đơn giản sau:

$$\text{Maximize } f(x, y) = 2 - x^2 - 2y^2$$

$$\text{subject to } g(x, y) = x + y - 1 = 0$$

Để giải bài toán tối ưu này ta sử dụng phương pháp Lagrange multipliers, hàm Lagrange được biểu diễn như sau [15]:

$$\begin{aligned} \mathcal{L}(x, y, \lambda) &= f(x, y) - \lambda \cdot g(x, y) \\ &= 2 - x^2 - 2y^2 - \lambda(x + y - 1) \end{aligned} \quad (3.32)$$

Phương trình đạo hàm riêng của hàm Lagrange

$$\nabla \mathcal{L}(x, y, \lambda) = \nabla f(x, y) - \lambda \cdot \nabla g(x, y) = 0 \quad (3.33)$$

Các đạo hàm riêng của hàm Lagrange

$$\frac{\partial}{\partial x} \mathcal{L}(x, y, \lambda) = -2x - \lambda = 0 \quad (3.34.1)$$

$$\frac{\partial}{\partial y} \mathcal{L}(x, y, \lambda) = -4y - \lambda = 0 \quad (3.34.2)$$

$$\frac{\partial}{\partial \lambda} \mathcal{L}(x, y, \lambda) = x + y - 1 = 0 \quad (3.34.3)$$

Giải hệ phương trình (3.34.1), (3.34.2) và (3.34.3) ta tìm được $x = \frac{2}{3}$; $y = \frac{1}{3}$; $\lambda = -\frac{4}{3}$ và giá trị của $f\left(\frac{2}{3}; \frac{1}{3}\right) = \frac{4}{3}$. Đến đây ta chỉ biết được $f\left(\frac{2}{3}; \frac{1}{3}\right) = \frac{4}{3}$ là một cực trị của hàm f . Để kiểm tra xem $\frac{4}{3}$ có phải là giá trị lớn nhất của hàm f hay không, ta sẽ tính giá trị của hàm f tại một điểm bất kỳ thỏa hàm g , sau đó so sánh kết quả với cực trị mà ta tìm được. $g(0; 1) = 0$; $f(0; 1) = 0$. Vậy $f\left(\frac{2}{3}; \frac{1}{3}\right) = \frac{4}{3}$ là giá trị lớn nhất của hàm f .

Multiple Constraints (Đa ràng buộc): Trong bài toán tối ưu (3.30), chỉ có một hàm ràng buộc g . Tuy nhiên, trong thực tế một bài toán tối ưu có thể có nhiều hơn một hàm ràng buộc. Phương pháp Lagrange multipliers có thể áp dụng trong trường hợp bài toán tối ưu có nhiều hàm ràng buộc, với $\vec{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_N)$ khi đó hàm Lagrange được biểu diễn tổng quát như sau:

$$\begin{aligned} \mathcal{L}(x_1, x_2, \dots, x_n, \vec{\lambda}) \\ = f(x_1, x_2, \dots, x_n) - \sum_{i=1}^N \lambda_i \cdot [g_i(x_1, x_2, \dots, x_n) - c] \end{aligned} \quad (3.35)$$

Phương trình đạo hàm riêng của hàm Lagrange có dạng $\nabla \mathcal{L}(x_1, x_2, \dots, x_n, \vec{\lambda}) = \nabla f(x_1, x_2, \dots, x_n) - \sum_{i=1}^N \lambda_i \cdot \nabla g_i(x_1, x_2, \dots, x_n)$

Inequality Constraints (Ràng buộc dạng bất phương trình): Ngoài khả năng áp dụng trong trường hợp bài toán tối ưu có nhiều hàm ràng buộc, phương pháp Lagrange multipliers còn có thể áp dụng được trong trường hợp các hàm ràng buộc là các bất phương trình. Phương pháp giải bài toán tối ưu trong trường hợp này cũng tương tự như phương pháp giải bài toán tối ưu trong trường hợp các hàm ràng buộc là các phương trình, chỉ có một điểm lưu ý là giá trị của λ_i sau khi tìm được cần kiểm tra lại với các điều kiện sau:

- + Nếu $g_i(x_1, x_2, \dots, x_n) \geq 0$ thì $\lambda_i \geq 0$
- + Nếu $g_i(x_1, x_2, \dots, x_n) \leq 0$ thì $\lambda_i \leq 0$
- + Nếu $g_i(x_1, x_2, \dots, x_n) = 0$ thì λ_i không bị ràng buộc

Phương pháp Lagrange multipliers trong SVM: Quay trở lại với bài toán tìm siêu phẳng tối ưu có margin lớn nhất trong thuật giải SVM. Như kết quả của mục trước, để tìm siêu phẳng tối ưu ta phải giải bài toán tối ưu (3.29) sau

$$\begin{aligned} & \text{Minimize}_{\vec{w}} \frac{\|\vec{w}\|^2}{2} \\ & \text{subject to } y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 \\ & (\forall i \ 1 \leq i \leq n) \end{aligned}$$

Áp dụng phương pháp Lagrange multipliers cho bài toán tối ưu trên ta có hàm Lagrange sau:

$$\mathcal{L}(\vec{w}, b, \vec{\lambda}) = \frac{1}{2} \vec{w} \cdot \vec{w} - \sum_{i=1}^n \lambda_i [y_i(\vec{w} \cdot \vec{x}_i + b) - 1] \quad (3.36)$$

Các đạo hàm riêng của hàm Lagrange

$$\begin{aligned} \frac{\partial}{\partial \vec{w}} \mathcal{L}(\vec{w}, b, \vec{\lambda}) &= \vec{w} - \sum_{i=1}^n \lambda_i y_i \vec{x}_i = 0 \\ \Rightarrow \vec{w} &= \sum_{i=1}^n \lambda_i y_i \vec{x}_i \end{aligned} \quad (3.37.1)$$

$$\frac{\partial}{\partial b} \mathcal{L}(\vec{w}, b, \vec{\lambda}) = \sum_{i=1}^n \lambda_i y_i = 0 \quad (3.37.2)$$

Thay (3.37.1) và (3.37.2) vào (3.36) ta có

$$\begin{aligned} \mathcal{L}(\vec{w}, b, \vec{\lambda}) &= \frac{1}{2} \vec{w} \cdot \vec{w} - \sum_{i=1}^n \lambda_i y_i \vec{w} \cdot \vec{x}_i - \sum_{i=1}^n \lambda_i y_i b + \sum_{i=1}^n \lambda_i \\ \mathcal{L}(\vec{w}, b, \vec{\lambda}) &= \frac{1}{2} \vec{w} \cdot \vec{w} - \vec{w} \sum_{i=1}^n \lambda_i y_i \vec{x}_i - \sum_{i=1}^n \lambda_i y_i b + \sum_{i=1}^n \lambda_i \\ \mathcal{L}(\vec{w}, b, \vec{\lambda}) &= \frac{1}{2} \vec{w} \cdot \vec{w} - \vec{w} \cdot \vec{w} + \sum_{i=1}^n \lambda_i \\ \mathcal{L}(\vec{w}, b, \vec{\lambda}) &= -\frac{1}{2} \vec{w} \cdot \vec{w} + \sum_{i=1}^n \lambda_i \end{aligned}$$

Chương 3. Mô hình kết hợp ARIMA và Support Vector Machine

$$\mathcal{L}(\vec{\lambda}) = \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \lambda_i y_i \vec{x}_i \sum_{j=1}^n \lambda_j y_j \vec{x}_j \quad (3.38)$$

Theo lý thuyết về vấn đề kép (dual problem), nếu biết \vec{w} ta có thể biết $\vec{\lambda}$ và ngược lại nếu biết $\vec{\lambda}$ ta có thể biết \vec{w} . Vì vậy mà bài toán tối ưu (3.29) tương đương với bài toán tối ưu sau:

$$\begin{aligned} & \text{Maximize } \mathcal{L}(\vec{\lambda}) \\ & \text{subject to } \lambda_i \geq 0 \text{ and } \sum_{i=1}^n \lambda_i y_i = 0 \end{aligned} \quad (3.39)$$

Phương pháp Quadratic Programming (QC) thường được dùng để giải bài toán tối ưu trên. Giả sử $\vec{\lambda}^0 = (\lambda_1^0, \lambda_2^0, \dots, \lambda_N^0)$ là kết quả của bài toán tối ưu này, khi đó giá trị của \vec{w}_0 được tính như sau:

$$\vec{w}_0 = \sum_{i=1}^n \lambda_i^0 y_i \vec{x}_i \quad (3.40)$$

Và giá trị của b_0 được suy ra từ điều kiện Karush-Kuhn-Tucker (KKT):

$$\lambda_i^0 [y_i (\vec{w}_0 \cdot \vec{x}_i + b_0) - 1] = 0 \quad (3.41)$$

Lưu ý, chỉ những điểm dữ liệu (\vec{x}_i, y_i) nằm trên siêu phẳng “biên” mới có giá trị $\lambda_i^0 \neq 0$, những điểm này được gọi là support vectors. Sau khi đã xác định được \vec{w}_0 và b_0 , để xác định một đối tượng \vec{x}_k nào đó thuộc vào phân lớp nào, đơn giản ta chỉ cần tính giá trị $\vec{w}_0 \cdot \vec{x}_k + b_0$ và so sánh kết quả, nếu kết quả lớn hơn hoặc bằng 1 thì \vec{x}_k thuộc vào phân lớp 1. Ngược lại, nếu kết quả bé hơn hoặc bằng -1 thì \vec{x}_k thuộc phân lớp -1.

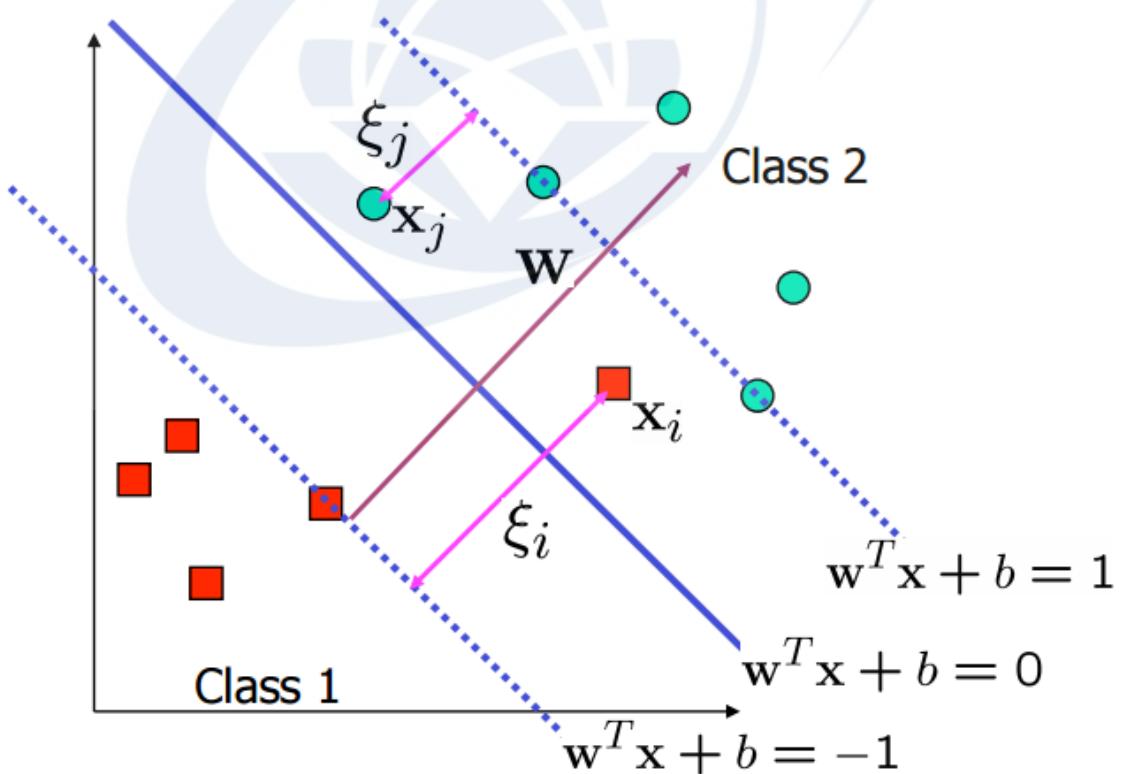
Trên đây là ý tưởng và các bước để thực hiện phân lớp dữ liệu của thuật giải SVM. Tuy nhiên dữ liệu trong thực tế thường bị tác động của nhiều yếu tố, dẫn đến dữ liệu thường xuyên bị nhiễu và không được phân lớp một cách tuyến tính (non-linear separate). Vì vậy mà thuật giải SVM có thêm hai cải tiến là Soft margin và Kernel để thích nghi với các yếu tố của dữ liệu.

3.2.5 Soft Margin và Kernel

Soft Margin: Hình 3.15 là một ví dụ về trường hợp phân lớp dữ liệu trong đó có 2 điểm dữ liệu nhiễu là x_i và x_j . Trong trường hợp này nếu xem hai điểm dữ liệu nhiễu này là các điểm dữ liệu bình thường và áp dụng thuật giải SVM sẽ dẫn đến kết quả là không tìm được một siêu phẳng tối ưu nào để phân lớp dữ liệu. Vì vậy mà thuật giải SVM được cải tiến cho trường hợp phân lớp dữ liệu bị nhiễu như sau:

$$\begin{aligned} & \text{Minimize } \frac{\|\vec{w}\|^2}{2} + C \sum_{i=1}^n \xi_i \\ & \text{subject to } y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \\ & (\forall i \ 1 \leq i \leq n) \end{aligned} \quad (3.42)$$

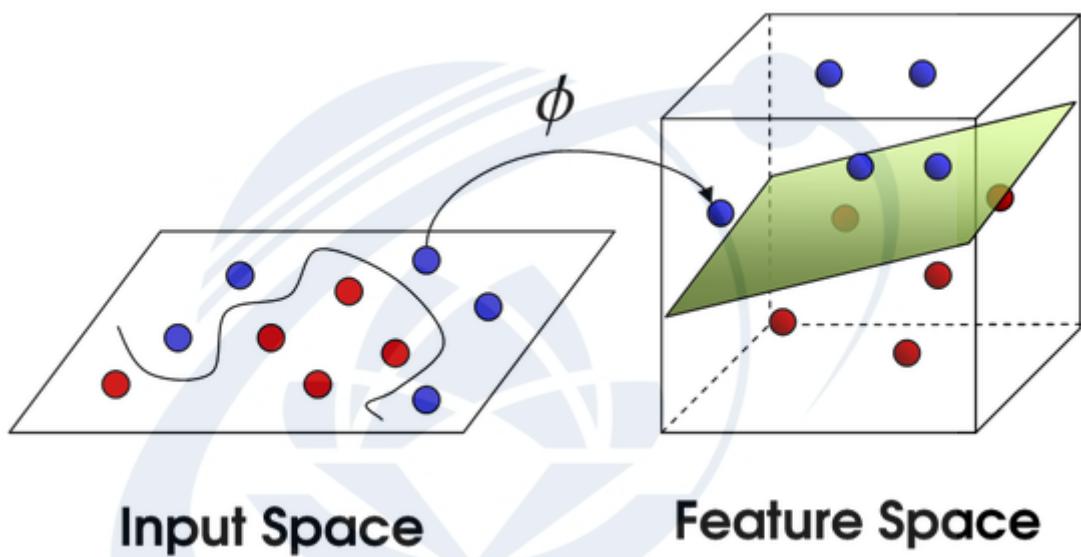
Trong đó C là một hằng số được dùng để tinh chỉnh vấn đề overfitting. Bài toán tối ưu này được giải theo cách tương tự như bài toán tối ưu (3.29). Margin trong trường hợp này được gọi là Soft Margin.



Hình 3.15. Ví dụ về Soft Margin

Nguồn: *Soft Margin* [31]

Kernel: Trong các ví dụ về bài toán phân lớp bằng thuật giải SVM từ đầu mục 3.2 đến bây giờ, dữ liệu dùng để xây dựng mô hình thường đã có tính phân lớp một cách tuyến tính (linear separate), tức là các lớp đối tượng thường được phân bố thành các cụm riêng biệt, không đan xen với nhau. Tuy nhiên, trong thực tế các đối tượng thuộc các phân lớp khác nhau thường có phân bố đan xen vào nhau như trong hình 3.16a. Điều này dẫn đến việc ta không thể tìm được một siêu phẳng nào để phân lớp dữ liệu. Do đó vào năm 1992 Vladimir N. Vapnik và các cộng sự của mình đã cải tiến thuật giải SVM bằng cách bổ sung thêm một



Hình 3.16. Ví dụ về Kernel

- a) Dữ liệu trong không gian ban đầu
- b) Dữ liệu sau khi được biến đổi vào không gian nhiều chiều hơn

Nguồn: *Kernel* [26]

khái niệm mới gọi là Kernel dùng để hỗ trợ phân lớp dữ liệu trong trường hợp dữ liệu phân lớp không tuyến tính.

Thông thường ta sẽ dùng một hàm $\phi: \mathbb{R}^n \rightarrow \mathbb{R}^m, n < m$ để biến đổi dữ liệu từ không gian ban đầu vào không gian nhiều chiều hơn sao cho dữ liệu có tính phân lớp. Tuy nhiên việc biến đổi và tính toán trong không gian nhiều chiều là khó khăn hơn trong không gian ít chiều. Do đó mà thuật giải SVM

Chương 3. Mô hình kết hợp ARIMA và Support Vector Machine

không sử dụng hàm ϕ để biến đổi dữ liệu vào không gian nhiều chiều, mà thay vào đó SVM sử dụng một khái niệm gọi là Kernel K.

$$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j) \quad (3.43)$$

Xét một ví dụ đơn giản, giả sử ta có 2 điểm dữ liệu trong không gian \mathbb{R}^2 là $\vec{x}(x_1, x_2)$ và $\vec{x}'(x'_1, x'_2)$ và một hàm $\phi(\vec{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$. Khi đó:

$$\phi(\vec{x}) \cdot \phi(\vec{x}') = (1 + x_1x'_1 + x_2x'_2) \quad (3.44)$$

Giả sử ta định nghĩa một kernel như sau:

$$K(\vec{x}, \vec{x}') = (1 + x_1x'_1 + x_2x'_2) \quad (3.45)$$

Từ (3.44) và (3.45) có thể thấy việc định nghĩa một hàm ϕ và biến đổi dữ liệu vào không gian nhiều chiều hơn về mặt tính toán là không cần thiết. Đơn giản ta chỉ cần sử dụng một kernel với kết quả tính toán tương đương. Do đó mà chúng ta chỉ cần quan tâm đến kernel mà không cần quan tâm đến việc làm thế nào để tìm một hàm biến đổi dữ liệu từ không gian ban đầu vào không gian nhiều chiều hơn. Nhưng dù sao đi nữa hàm ϕ vẫn cho ta một hình dung tốt về ý tưởng biến đổi dữ liệu không tuyến tính thành dữ liệu tuyến tính trong thuật giải SVM.

Một số kernel thường dùng:

Polynomial kernel với bậc d

$$K(\vec{x}, \vec{x}') = (\vec{x}^T \vec{x}' + 1)^d \quad (3.46)$$

Radial kernel hay Gaussian kernel với tham số σ

$$K(\vec{x}, \vec{x}') = \exp\left(-\|\vec{x} - \vec{x}'\|^2 / (2\sigma^2)\right) \quad (3.47)$$

Sigmoid kernel với hai tham số κ và θ

$$K(\vec{x}, \vec{x}') = \tanh(\kappa \vec{x}^T \vec{x}' + \theta) \quad (3.48)$$

Với kernel $K(\vec{x}_i, \vec{x}_j)$, bài toán tìm một siêu phẳng của thuật giải SVM được biểu diễn lại như sau:

$$\text{Maximize } \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i,j=1}^n \lambda_i \lambda_j y_i y_j K(\vec{x}_i, \vec{x}_j) \quad (3.49)$$

subject to $\lambda_i \geq 0$ and $\sum_{i=1}^n \lambda_i y_i = 0$

Thuật giải SVM thường được giới thiệu như là một phương pháp tốt để phân lớp dữ liệu. Nhưng ý tưởng và phương pháp của thuật giải SVM về vấn đề tìm kiếm một siêu phẳng tối ưu có thể áp dụng được trong nhiều lĩnh vực khác. Một trong số đó là lĩnh vực dự báo dữ liệu chuỗi thời gian với phương pháp Support Vector Regression (SVR). Phản tiếp theo của báo cáo sẽ trình bày về SVR dựa trên ý tưởng của thuật giải SVM và ứng dụng SVR vào dự báo dữ liệu chuỗi thời gian.

3.2.6 Support Vector Machine trong dự báo chuỗi thời gian

Support Vector Machine trong ước lượng hồi quy (SVM for regression estimation) còn được gọi là phương pháp Support Vector Regression (SVR). SVM trong phân lớp dữ liệu và SVR trong ước lượng hồi quy là hai phương pháp giải quyết hai vấn đề hoàn toàn khác nhau. Tuy nhiên, ý tưởng về việc sử dụng support vectors và kernel để tìm kiếm một lời giải tối ưu cho hai vấn đề này là như nhau. Chính vì vậy mà phương pháp SVR được xem như là một ứng dụng của SVM trong bài toán ước lượng hồi quy.

Để ứng dụng SVR vào dự báo dữ liệu chuỗi thời gian trước tiên ta xem xét nhận xét sau. Chuỗi thời gian $x(t), t = 1, 2, \dots, n$. Vector $x_t = (x(t), x(t - \tau), \dots, x(t - (d - 1)\tau))$ với τ là thời gian trễ (time delay) và d là số chiều của vector. Nếu giá trị của d đủ lớn thì vector x_t được xem như một vector mô tả trạng thái của chuỗi thời gian tại thời điểm t . Nếu ta tìm được một hàm f để mô hình hóa cho chuỗi thời gian này thì giá trị của chuỗi thời gian tại điểm $t + 1$ sẽ được xác định bằng công thức $x(t + 1) = f(x_t)$ [29].

Tổng quát, cho tập dữ liệu với các điểm dữ liệu $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ với $x_i \in \mathbb{R}^m, y_i \in \mathbb{R}$ là ngẫu nhiên và độc lập, được sinh ra từ một hàm chưa biết. Khi đó SVR xấp xỉ hàm này bằng một hàm có dạng như sau [7]:

$$f(\vec{x}) = \vec{w} \cdot \vec{x} + b \quad (3.50)$$

Chương 3. Mô hình kết hợp ARIMA và Support Vector Machine

Mục tiêu của phương pháp SVR là tìm kiếm một hàm f có dạng như trên sao cho độ lệch giữa giá trị của hàm f và giá trị thực tế tại các thời điểm không vượt quá độ lệch ε và f là “phẳng” nhất có thể (as flat as possible). Nói cách khác, SVR không quan tâm những độ lệch bé hơn ε và cũng không chấp nhận những độ lệch lớn hơn ε [16]. Giá trị của \vec{w} và b được xác định bằng cách minimize hàm rủi ro (risk function) sau [7]:

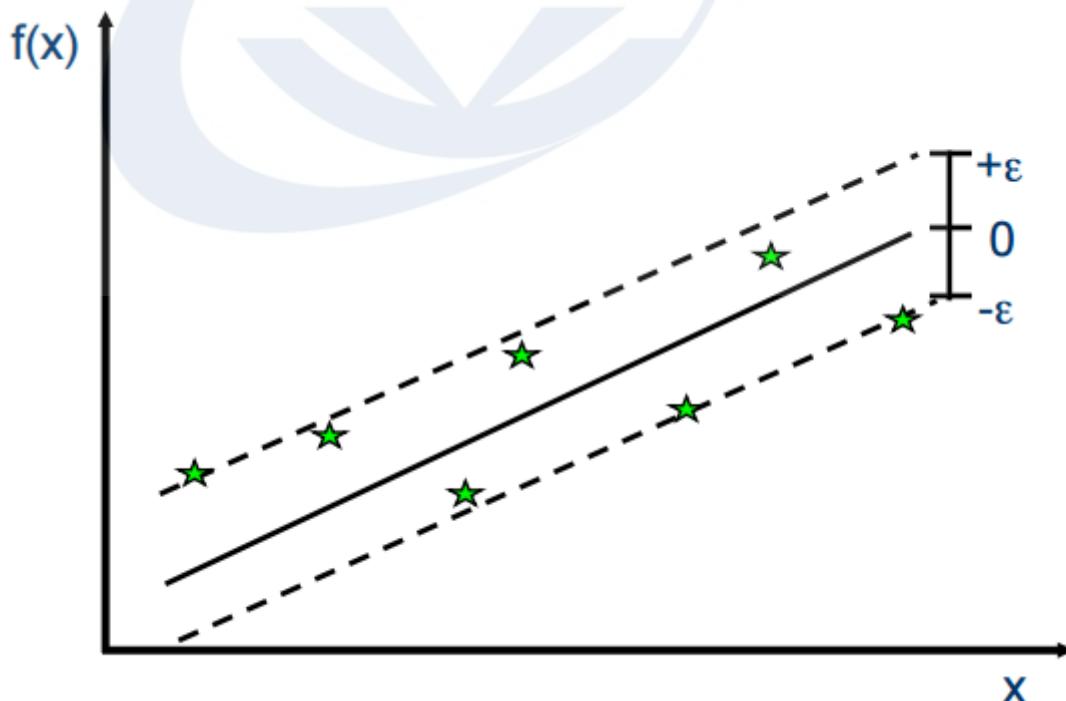
$$\text{minimize } \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^n L_\varepsilon(y_i, f(\vec{x}_i)) \quad (3.51)$$

Trong đó:

+ Minimize $\|\vec{w}\|^2$ sẽ làm cho hàm f “phẳng” nhất có thể.

$$+ L_\varepsilon(y_i, f(\vec{x}_i)) = \begin{cases} |y_i - f(\vec{x}_i)| - \varepsilon, & |y_i - f(\vec{x}_i)| > \varepsilon \\ 0, & |y_i - f(\vec{x}_i)| \leq \varepsilon \end{cases}$$

+ C là một hằng số dùng để điều chỉnh giữa “độ phẳng” (flatness) của hàm f và số lượng các điểm dữ liệu có độ lệch lớn hơn ε . Hàm f càng phẳng thì số lượng các điểm dữ liệu có độ lệch lớn hơn ε càng tăng và ngược lại.



Hình 3.17. SVR trong ước lượng hồi quy

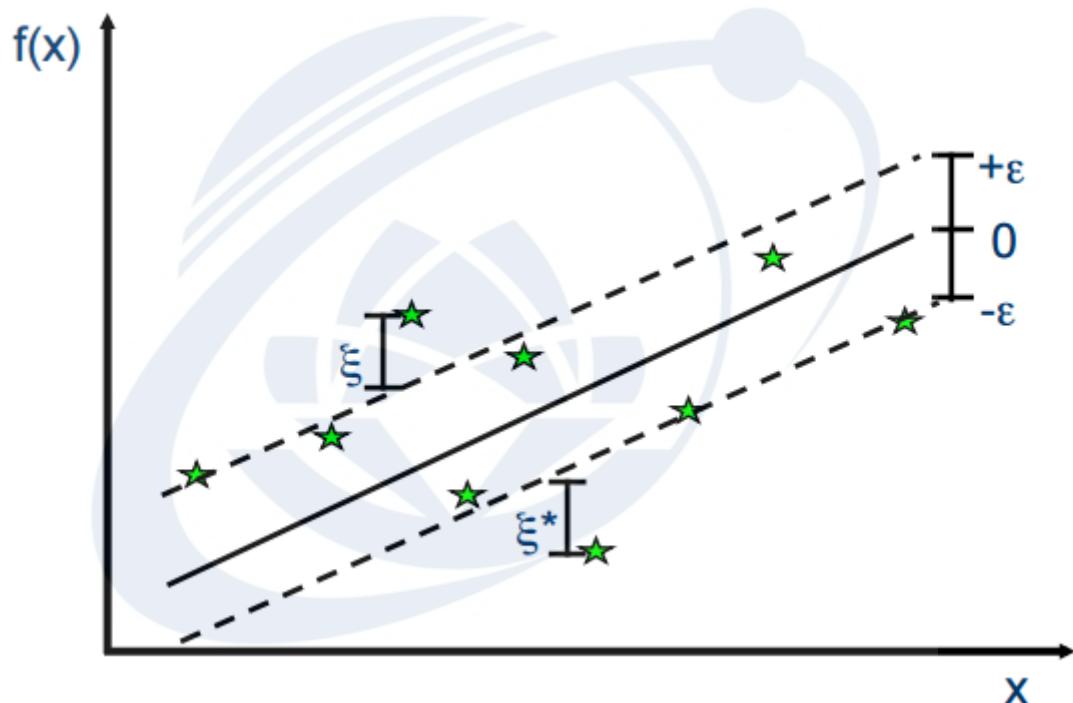
Nguồn: *Support Vector Regression* [27]

Chương 3. Mô hình kết hợp ARIMA và Support Vector Machine

Gọi $\xi^{(*)} = |y_i - f(x_i)| - \varepsilon$, khi đó vấn đề minimize hàm rủi ro trong (3.51) sẽ tương đương với bài toán tối ưu sau:

$$\begin{aligned} & \text{Minimize } \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ & \text{subject to } y_i - \vec{w} \cdot \vec{x}_i - b \leq \varepsilon + \xi_i \\ & \quad \vec{w} \cdot \vec{x}_i + b - y_i \leq \varepsilon + \xi_i^* \\ & \quad \xi_i^{(*)} \geq 0 \end{aligned} \quad (3.52)$$

Lưu ý: $\xi_i^{(*)}$ là ký hiệu chung để chỉ ξ_i hoặc ξ_i^* . Có thể thấy $\xi_i^{(*)}$ trong SVR có nét tương đồng với ξ_i trong soft margin của SVM.



Hình 3.18. SVR trong ước lượng hồi quy

Nguồn: *Support Vector Regression* [27]

Tương tự như thuật giải SVM, để giải bài toán tối ưu này ta sử dụng phương pháp Lagrange mutiplier. Khi đó hàm Lagrange có dạng như sau:

$$\begin{aligned} L = & \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) - \sum_{i=1}^n \lambda_i (\varepsilon + \xi_i - y_i + \vec{w} \cdot \vec{x}_i + b) \\ & - \sum_{i=1}^n \lambda_i^* (\varepsilon + \xi_i^* + y_i - \vec{w} \cdot \vec{x}_i - b) - \sum_{i=1}^n \eta_i \xi_i - \sum_{i=1}^n \eta_i^* \xi_i^* \end{aligned} \quad (3.53)$$

Chương 3. Mô hình kết hợp ARIMA và Support Vector Machine

Trong đó $\lambda_i, \lambda_i^*, \eta_i, \eta_i^* \geq 0$ là các Lagrange mutiplier. Các đạo hàm riêng của hàm Lagrange:

$$\frac{\partial}{\partial b} L = \sum_{i=1}^n (\lambda_i^* - \lambda_i) = 0 \quad (3.54)$$

$$\frac{\partial}{\partial w} L = w - \sum_{i=1}^n (\lambda_i - \lambda_i^*) \vec{x}_i = 0 \quad (3.55)$$

$$\frac{\partial}{\partial \xi_i^{(*)}} L = C - \lambda_i^{(*)} - \eta_i^{(*)} = 0 \quad (3.56)$$

Thay (3.54), (3.55) và (3.56) vào (3.53) và theo lý thuyết về vấn đề kép (dual problem) ta có bài toán tối ưu tương đương với bài toán tối ưu (3.52) như sau:

$$\begin{aligned} & \text{Maximize} \quad \sum_{i=1}^n y_i (\lambda_i - \lambda_i^*) - \varepsilon \sum_{i=1}^n y_i (\lambda_i + \lambda_i^*) - \frac{1}{2} \sum_{i=1}^n (\lambda_i - \lambda_i^*) \sum_{j=1}^n (\lambda_j - \lambda_j^*) (\vec{x}_i \cdot \vec{x}_j) \\ & \text{subject to} \quad \sum_{i=1}^n y_i (\lambda_i - \lambda_i^*) = 0 \\ & \quad 0 \leq \lambda_i, \lambda_i^* \leq C \end{aligned} \quad (3.57)$$

Tương tự như trong thuật giải SVM, phương pháp Quadratic Programming (QC) cũng được áp dụng để giải bài toán tối ưu trên. Sau khi xác định được các giá trị của λ_i, λ_i^* , giá trị của \vec{w} và b được tính bằng các công thức sau:

$$\vec{w} = \sum_{i=1}^n (\lambda_i - \lambda_i^*) \vec{x}_i \quad (3.58)$$

Áp dụng điều kiện Karush-Kuhn-Tucker (KKT) ta có:

$$b = y_i - \vec{w} \cdot \vec{x}_i - \varepsilon, \forall \lambda_i \in (0, C) \quad (3.59.1)$$

$$b = y_i - \vec{w} \cdot \vec{x}_i + \varepsilon, \forall \lambda_i^* \in (0, C) \quad (3.59.2)$$

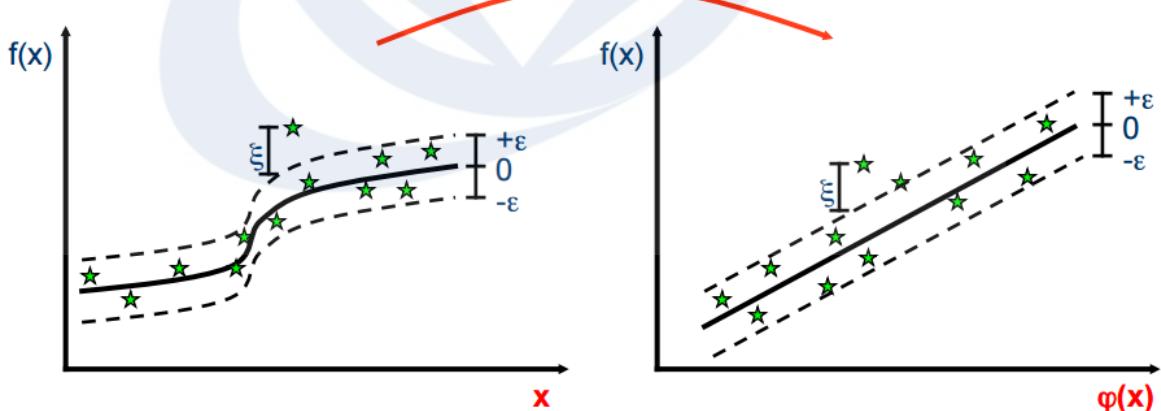
Vậy hàm f trong (3.50) được xác định như sau:

$$f(\vec{x}) = \sum_{i=1}^n (\lambda_i - \lambda_i^*) (\vec{x}_i \cdot \vec{x}) + b \quad (3.60)$$

Trong thuật giải SVM dữ liệu cần phân lớp thường không được phân chia một cách tuyến tính, tức là các đối tượng dữ liệu của các lớp khác nhau lại phân bố đan xen vào nhau, điều đó gây khó khăn cho thuật giải SVM trong việc tìm kiếm một siêu phẳng để phân lớp dữ liệu. Do đó mà trong thuật giải SVM sử dụng các kernel để biến đổi dữ liệu lên các không gian nhiều chiều hơn sao cho dữ liệu được phân chia một cách tuyến tính trong không gian đó và có thể tìm kiếm được một siêu phẳng tối ưu để phân lớp dữ liệu. Trong bài toán ước lượng hồi quy bằng phương pháp SVR cũng vậy. Thực tế dữ liệu hồi quy thường không tuyến tính, do đó để tìm kiếm một hàm f tối ưu mô hình hóa dữ liệu, phương pháp SVR sử dụng các kernel để biến đổi dữ liệu.

Sử dụng kernel $K(\vec{x}_i, \vec{x})$ để biến đổi dữ liệu, khi đó hàm f được biểu diễn lại dưới dạng sau:

$$f(\vec{x}) = \sum_{i=1}^n (\lambda_i - \lambda_i^*) K(\vec{x}_i, \vec{x}) + b \quad (3.61)$$



Hình 3.19. Biến đổi dữ liệu từ không tuyến tính thành tuyến tính

Nguồn: *Support Vector Regression* [27]

Mặc dù các phương pháp ARIMA, SVR hay mạng neural đều là các phương pháp nổi bật và cho kết quả tốt trong lĩnh vực dự báo dữ liệu chuỗi thời gian, nhưng trong thực tế dữ liệu thường bị ảnh hưởng bởi nhiều yếu tố khác nhau dẫn đến kết quả dự báo của những mô hình riêng biệt đôi khi không được như mong

Chương 3. Mô hình kết hợp ARIMA và Support Vector Machine

muốn. Do đó nhu cầu cần thiết là tìm kiếm một phương pháp có thể kết hợp các ưu điểm và khắc phục các nhược điểm của từng mô hình. Một trong số các mô hình kết hợp đó là mô hình kết hợp ARIMA và Support Vector Machine. Phần tiếp theo của báo cáo sẽ trình về mô hình kết hợp ARIMA và Support Vector Machine trong lĩnh vực dự báo dữ liệu chuỗi thời gian.

3.3 Mô hình kết hợp ARIMA và Support Vector Machine

3.3.1 Giới thiệu

Như đã trình bày trong chương 1, chuỗi thời gian trong thực tế thường bị ảnh hưởng hoặc tác động bởi nhiều yếu tố khác nhau, các yếu tố này thường được phân chia làm hai nhóm dựa trên nguồn gốc xuất hiện của chúng.

Nhóm thứ nhất là nhóm các yếu tố phát sinh từ “bên trong” chuỗi thời gian, nhóm yếu tố này là nhóm yếu tố chính thường xuyên chi phối sự thay đổi của chuỗi thời gian. Ví dụ chuỗi thời gian là tốc độ dòng chảy của một con sông, khi đó các yếu tố như lượng mưa, độ dốc của địa hình hay tốc độ gió là các yếu tố chính ảnh hưởng lên chuỗi thời gian này. Hay chuỗi thời gian là giá cổ phiếu của một công ty, khi đó tình hình kinh doanh của công ty, quy luật cung cầu là những yếu tố tác động chính lên giá cổ phiếu của công ty đó. Các yếu tố trong nhóm này quy định đặc tính tuyến tính của chuỗi thời gian.

Nhóm các yếu tố thứ hai là các yếu tố “bên ngoài” ảnh hưởng lên chuỗi thời gian. Đây là các yếu tố ngẫu nhiên, tuy không thường xuyên ảnh hưởng lên chuỗi thời gian nhưng những ảnh hưởng của nó thường gây tác động lớn lên chuỗi thời gian. Ví dụ chuỗi thời gian là giá vàng trên thế giới thường bị ảnh hưởng lớn bởi các yếu tố ngẫu nhiên như thiên tai, chiến tranh hay các sự kiện chính trị. Các yếu tố trong nhóm này quy định đặc tính phi tuyến tính của chuỗi thời gian.

Chính vì các đặc tính tuyến tính và phi tuyến tính này của chuỗi thời gian mà kết quả dự báo của các mô hình riêng biệt đôi khi không được như mong đợi. Lý do là bởi các mô hình dự báo riêng biệt thường chỉ phù hợp để dự báo cho một số thành phần của chuỗi thời gian. Ví dụ mô hình ARIMA phù hợp để dự báo cho thành phần tuyến tính của chuỗi thời gian, trong khi thành phần phi tuyến

tính mô hình ARIMA thường bỏ qua, không dự báo được. Ngược lại, các mô hình máy học như SVM hay mạng neural lại thích hợp để dự báo cho thành phần phi tuyến tính của chuỗi thời gian hơn là thành phần tuyến tính. Vì vậy mà việc cần thiết là tìm cách kết hợp các mô hình dự báo riêng biệt này lại với nhau sao cho có thể phát huy các ưu điểm cũng như khắc phục được các nhược điểm của từng mô hình.

Mô hình kết hợp ARIMA và Support Vector Machine là một trong những mô hình tiếp cận theo hướng trên. Mô hình này sử dụng mô hình ARIMA để dự báo cho thành phần tuyến tính của chuỗi thời gian, đồng thời sử dụng phương pháp SVM trong ước lượng hồi quy để dự báo cho thành phần phi tuyến tính của chuỗi thời gian. Sau đó kết quả dự báo của hai mô hình này sẽ được kết hợp lại với nhau để cho kết quả dự báo sau cùng.

3.3.2 Nội dung

Chuỗi thời gian bao gồm hai thành phần tuyến tính (linear) và không tuyến tính (nonlinear). Do đó một chuỗi thời gian x_t có thể được mô hình hóa như sau:

$$x_t = L_t + N_t \quad (3.62)$$

Trong đó: L_t đại diện cho thành phần tuyến tính (Linear) của chuỗi thời gian, N_t đại diện cho thành phần phi tuyến tính (Nonlinear) của chuỗi thời gian. Cả hai thành phần này đều được ước lượng từ dữ liệu.

Đầu tiên, mô hình ARIMA được sử dụng để ước lượng cho thành phần tuyến tính L_t của chuỗi thời gian. Giả sử \hat{L}_t là kết quả dự báo của mô hình ARIMA. Khi đó thành phần còn lại (residuals) e_t của chuỗi thời gian sau khi lấy kết quả thực tế trừ kết quả dự báo được xác định như sau:

$$e_t = x_t - \hat{L}_t \quad (3.63)$$

Thành phần còn lại e_t chứa trong nó thành phần phi tuyến tính của chuỗi thời gian. Do đó bước tiếp theo phương pháp SVM trong ước lượng hồi quy được sử dụng để dự báo thành phần phi tuyến tính N_t này dựa trên các e_t . Giả sử

Chương 3. Mô hình kết hợp ARIMA và Support Vector Machine

phương pháp SVM tìm được một hàm f tối ưu có thể mô hình hóa cho thành phần phi tuyến tính của chuỗi thời gian. Khi đó:

$$e_t = f(e_{t-1}, e_{t-2}, \dots, e_{t-n}) + \varepsilon_t \quad (3.64)$$

Trong đó: ε_t là một giá trị lỗi ngẫu nhiên tại thời điểm t .

Sau cùng, kết quả dự báo của mô hình là tổng hợp kết quả dự báo của thành phần tuyến tính \hat{L}_t bằng mô hình ARIMA và kết quả dự báo của thành phần phi tuyến tính \hat{N}_t bằng phương pháp SVM trong ước lượng hồi quy.

$$\hat{x}_t = \hat{L}_t + \hat{N}_t \quad (3.65)$$

3.3.3 Một số kết quả tham khảo và đánh giá

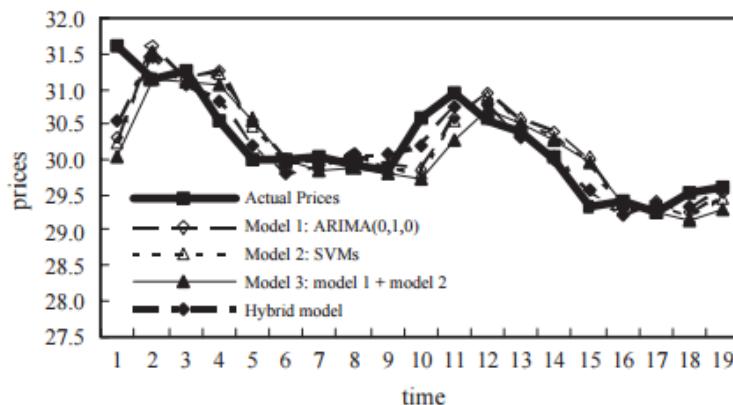
Mô hình kết hợp ARIMA và Support Vector Machine đã được nghiên cứu và ứng dụng trong một số lĩnh vực như dự báo chứng khoán, dự báo năng lượng, dự báo trong trồng trọt và chăn nuôi,... Sau đây là kết quả dự báo của mô hình kết hợp này trong một số nghiên cứu đã được công bố.

Dự báo chứng khoán

Bảng 3.2. So sánh kết quả dự báo giá cổ phiếu Công ty Eastman Kodak

Loại mô hình	MAE	MSE	MAPE	RMSE
ARIMA	0.3495	0.2257	1.1494	0.4751
SVM	0.3466	0.2247	1.1433	0.4740
ARIMA + SVM	0.3499	0.2138	1.1534	0.4624
Hybrid	0.2303	0.1000	0.7598	0.3162

Chú thích: Bảng so sánh được trích dẫn từ [14].



Hình 3.20. Biểu đồ so sánh kết quả dự báo giá cổ phiếu Công ty Eastman Kodak

Nguồn: *Biểu đồ so sánh kết quả dự báo giá cổ phiếu Công ty Eastman Kodak* [14]

Trong kết quả nghiên cứu này, các tác giả đã cài đặt 4 mô hình dự báo: Mô hình 1 là mô hình ARIMA, mô hình 2 là mô hình SVM, mô hình 3 là mô hình kết hợp ARIMA và SVM với các tham số mặc định, mô hình 4 cũng là mô hình kết hợp ARIMA và SVM tuy nhiên các tham số đã được tác giả lựa chọn tối ưu để cho kết quả dự báo tốt nhất.

Dự báo năng lượng

Bảng 3.3. So sánh kết quả dự báo sản lượng điện cung cấp của Công ty Heilongjiang of China từ 12/04/1999 đến 31/05/1999

Loại mô hình	MAPE	RMSE
ARIMA	4.50%	43.49
SVM	4.00%	38.77
Hybrid	3.85%	35.72

Chú thích: Bảng so sánh được trích dẫn từ [13].

Dự báo trồng trọt và chăn nuôi

Bảng 3.4. So sánh kết quả dự báo sản lượng xuất khẩu hoa lan của Thái Lan từ 01/2007 đến 03/2011

Loại mô hình	MAE	MAPE	RMSE
ARIMA	256.4365	10.59%	365.1041
SVM	342.6776	14.08%	477.8260
Hybrid	250.1385	10.10%	357.9637

Chú thích: Bảng so sánh được trích dẫn từ [19].

Bảng 3.5. So sánh kết quả dự báo sản lượng xuất khẩu thịt heo của Thái Lan từ 01/2007 đến 03/2011

Loại mô hình	MAE	MAPE	RMSE
ARIMA	1,113.9550	11.49%	1,389.8392
SVM	1,326.0020	13.12%	1,728.3100
Hybrid	1,023.9810	10.70%	1,303.1289

Chú thích: Bảng so sánh được trích dẫn từ [19].

Dựa trên các kết quả nghiên cứu trên có thể thấy mô hình kết hợp ARIMA và Support Vector Machine cho kết quả dự báo tốt hơn so với các mô hình riêng biệt. Do đó có thể sử dụng mô hình kết hợp này để dự báo giá trị của chuỗi thời gian. Từ những ưu điểm và kết quả dự báo của mô hình kết hợp ARIMA và Support Vector Machine chương tiếp theo của báo cáo sẽ trình bày một ứng dụng cụ thể của mô hình này vào dự báo tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt.

Chương 4. DỰ BÁO TẠI CÔNG TY DỊCH VỤ TRỰC TUYẾN CỘNG ĐỒNG VIỆT

Chương này sẽ trình bày về một ứng dụng cụ thể của mô hình kết hợp ARIMA và Support Vector Machine trong dự báo dữ liệu chuỗi thời gian nhằm đánh giá kết quả dự báo của mô hình kết hợp này, đồng thời tìm kiếm một mô hình có thể đáp ứng được nhu cầu dự báo tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt.

4.1 Giới thiệu về công ty và bài toán dự báo

Công ty Dịch vụ Trực tuyến Cộng Đồng Việt được thành lập vào năm 2008. Công ty hoạt động trong lĩnh vực trung gian thanh toán và tích hợp hệ thống. Một trong những dịch vụ mà công ty cung cấp cho người dùng là tiện ích thanh toán hóa đơn.

Ngày nay, tại những thành phố lớn như Hà Nội, Hồ Chí Minh, Đà Nẵng,... những nhân viên công sở làm việc trong giờ hành chính thường gấp rắc rối với việc thanh toán các loại hóa đơn như hóa đơn tiền điện, tiền nước, tiền cước viễn thông, cước internet,... Lý do là những nhân viên thu tiền thường đến nhà họ thu vào giờ hành chính và họ thường không có mặt ở nhà vào giờ đó để chờ đóng các loại hóa đơn này, chính vì vậy mà họ thường chậm thanh toán dẫn đến việc bị cắt điện, cắt nước, cắt internet,... Mặc dù nhiều ngân hàng cũng đã cung cấp dịch vụ thu hộ nhưng ngân hàng cũng chỉ làm việc trong giờ hành chính, còn hình thức thanh toán online bằng eBanking của ngân hàng cũng tiềm ẩn nhiều rủi ro. Do đó mà Công ty Dịch vụ Trực tuyến Cộng Đồng Việt đã cung cấp cho người dùng dịch vụ thanh toán hóa đơn.

Với dịch vụ thanh toán hóa đơn của Công ty Dịch vụ Trực tuyến Cộng Đồng Việt người dùng sẽ không phải bận tâm với việc làm thế nào để thanh toán các loại hóa đơn như trước, mà thay vào đó người dùng chỉ đơn giản chọn một trong nhiều hình thức thanh toán cả online và offline do Công ty Dịch vụ Trực tuyến Cộng Đồng Việt cung cấp để thanh toán các loại hóa đơn của mình.

Chương 4. Dự báo tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt

Với hình thức online người dùng có thể thanh toán trực tuyến tại website thanh toán hóa đơn của Công ty Dịch vụ Trực tuyến Cộng Đồng Việt tại địa chỉ www.paybill.vn hoặc thanh toán qua ứng dụng thanh toán hóa đơn có thể download trực tiếp từ App Store hoặc Google Play.

Với hình thức offline người dùng càng tiện lợi hơn khi chỉ cần chưa đến 1 phút có thể thanh toán xong hóa đơn tại các siêu thị điện máy có liên kết với Công ty Dịch vụ Trực tuyến Cộng Đồng Việt như Nguyễn Kim, Thế Giới Di Động, Viễn Thông A, FPT Shop,... hay tại các cửa hàng tiện lợi như Circle K, Family Mart, Vinmart,... Ưu điểm của hình thức thanh toán này là nhanh, gọn và tiện lợi vì các cửa hàng này thường ở gần nhà và khu đông dân cư, bên cạnh đó các cửa hàng này cũng hoạt động 24/24 nên bất cứ khi nào người dùng cũng có thể đến để thanh toán các loại hóa đơn.

Bên cạnh việc mở rộng mạng lưới các cửa hàng, Công ty Dịch vụ Trực tuyến Cộng Đồng Việt cũng đã kết nối được với nhiều nhà cung cấp như Công ty Điện lực Hồ Chí Minh, Công ty Điện lực Hà Nội, các công ty cấp nước, truyền hình SCTV, FPT Telecom,... để giúp người dùng có thể thanh toán được nhiều loại hóa đơn của nhiều nhà cung cấp khác nhau.

Chính vì những tiện ích đó mà càng ngày càng có nhiều người dùng tin tưởng sử dụng dịch vụ thanh toán hóa đơn của Công ty Dịch vụ Trực tuyến Cộng Đồng Việt. Sự phát triển đó mang đến nhiều hy vọng nhưng cũng đặt ra cho Công ty Dịch vụ Trực tuyến Cộng Đồng Việt nhiều vấn đề cần phải giải quyết. Một trong những vấn đề đó là dự báo số lượng giao dịch trên ngày.

Số lượng giao dịch trên ngày chắc chắn luôn luôn biến động. Do đó để có thể phục vụ khách hàng tốt hơn cũng như nâng cao chất lượng dịch vụ của công ty việc cần thiết là phải dự báo được số lượng giao dịch trên ngày. Việc phân tích và dự báo số lượng giao dịch trên ngày mang lại nhiều lợi ích như:

+ Lợi ích về nhân sự: việc dự báo lượng giao dịch trên từng ngày sẽ giúp các nhà quản lý sắp xếp nhân sự một cách hợp lý để hỗ trợ tốt cho khách hàng

Chương 4. Dự báo tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt

đến giao dịch, hay bố trí thêm nhân viên trực tổng đài, đảm bảo chất lượng dịch vụ trong những ngày cao điểm.

+ Lợi ích về kinh doanh: kết quả dự báo giúp các nhà quản lý tham khảo để đưa ra các chiến lược kinh doanh và marketing hiệu quả.

+ Lợi ích về mặt kỹ thuật: giúp các nhà vận hành hệ thống có kế hoạch đảm bảo hệ thống ổn định và sắp xếp lịch bảo trì hệ thống hợp lý sao cho ít ảnh hưởng đến khách hàng nhất có thể.

Từ những lợi ích trên có thể thấy vấn đề dự báo số lượng giao dịch trên ngày là một vấn đề quan trọng cần phải giải quyết. Tương tự như các bài toán khai thác dữ liệu khác, để có kết quả dự báo chính xác, dữ liệu cần được tiền xử lý để biến đổi về dạng phù hợp với phương pháp dự báo. Phần tiếp theo sẽ trình bày về giai đoạn chuẩn bị và tiền xử lý dữ liệu.

4.2 Chuẩn bị và tiền xử lý dữ liệu

Giai đoạn chuẩn bị và xử lý dữ liệu ban đầu luôn là một giai đoạn quan trọng trong quy trình khai thác dữ liệu. Ứng với từng phương pháp khai thác dữ liệu khác nhau mà giai đoạn chuẩn bị và tiền xử lý dữ liệu được thực hiện khác nhau, nhưng nhìn chung mục tiêu của giai đoạn này là cố gắng loại bỏ các dữ liệu nhiễu, dữ liệu thừa, bổ sung các dữ liệu bị thiếu, biến đổi dữ liệu ban đầu về dạng phù hợp với phương pháp khai thác dữ liệu.



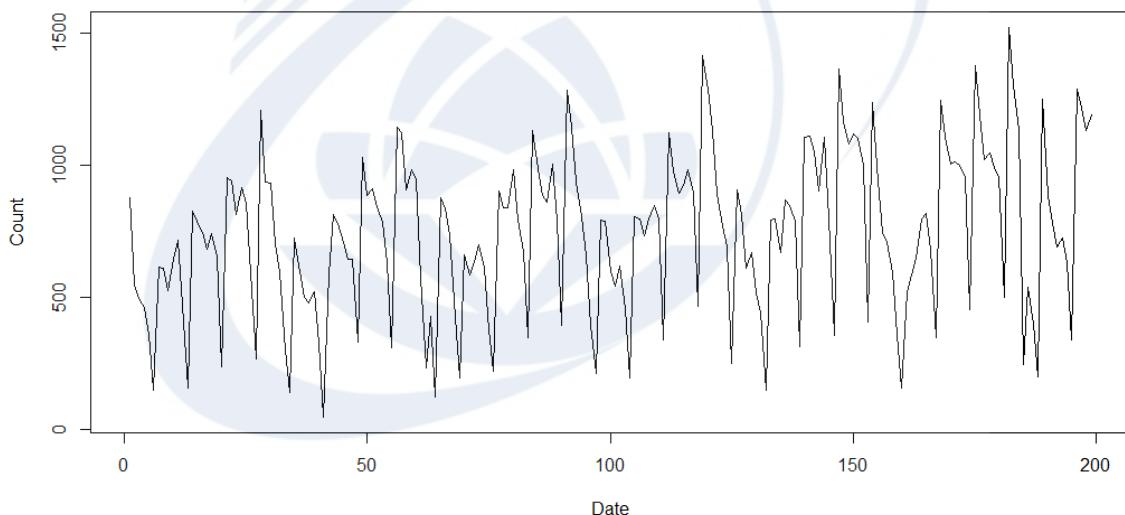
Hình 4.1. Quy trình khai thác dữ liệu

Đối với vấn đề dự báo số lượng giao dịch trên ngày tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt, dữ liệu sử dụng để xây dựng mô hình được thu thập từ dữ liệu giao dịch. Dữ liệu được thu thập nằm trong khoảng từ 01/07/2014 đến 15/01/2015. Để đảm bảo an toàn thông tin, dữ liệu dùng xây dựng mô hình chỉ được chọn từ tập dữ liệu giao dịch của một nhà cung cấp là Công ty Điện lực Hồ Chí Minh.

Chương 4. Dự báo tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt

Do mục tiêu là dự báo số lượng giao dịch trên ngày nên từ dữ liệu giao dịch ban đầu cần được biến đổi về dạng phù hợp bằng cách nhóm các giao dịch theo từng ngày và tính tổng số lượng giao dịch trên từng ngày. Bên cạnh đó những thông tin như họ tên, địa chỉ khách hàng, số tiền thanh toán,... là những thông tin không cần thiết cho việc dự báo được loại bỏ để đảm bảo bí mật thông tin của khách hàng. Ngoài ra những dữ liệu nhiễu như giao dịch lỗi, giao dịch bị hủy đều được loại bỏ để đảm bảo không ảnh hưởng đến kết quả dự báo của mô hình.

Tập dữ liệu để xây dựng mô hình được chia làm 2 phần. Phần thứ nhất dùng để huấn luyện (train) mô hình được chọn từ 01/07/2014 đến 31/12/2014, tổng số 184 ngày. Phần thứ hai dùng để kiểm tra (test) mô hình được chọn từ 01/01/2015 đến 15/01/2015, tổng số 15 ngày.



Hình 4.2. Biểu đồ số lượng giao dịch theo ngày từ 01/07/2014 đến 15/01/2015

Sau khi dữ liệu được xử lý và biến đổi về dạng phù hợp với mô hình dự báo. Phần tiếp theo báo cáo sẽ trình bày về các bước dự báo của mô hình kết hợp ARIMA và Support Vector Machine dựa trên tập dữ liệu này.

4.3 Dự báo

Mô hình kết hợp ARIMA và Support Vector Machine thực hiện hai bước dự báo. Bước đầu tiên sử dụng mô hình ARIMA để dự báo thành phần tuyến tính

Chương 4. Dự báo tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt

của chuỗi thời gian. Bước thứ hai sử dụng phương pháp SVM trong ước lượng hồi quy để dự báo thành phần phi tuyến tính của chuỗi thời gian.

4.3.1 Dự báo thành phần tuyến tính bằng mô hình ARIMA

Kiểm tra tính dừng của chuỗi thời gian: Như đã trình bày trong chương 3, mô hình ARIMA có thể dự báo cho cả chuỗi thời gian dừng và chuỗi thời gian không dừng. Trong trường hợp chuỗi thời gian dừng, tham số $d = 0$, khi đó mô hình ARIMA trở thành mô hình ARMA. Ngược lại ta có thể lấy sai phân của chuỗi thời gian để biến đổi nó về dạng chuỗi thời gian dừng. Vậy trước khi thực hiện dự báo ta cần kiểm tra tính dừng của chuỗi thời gian. Để xác định chuỗi thời gian có tính dừng hay không, có thể quan sát đồ thị của chuỗi thời gian, trong hình 4.2 có thể thấy giá trị trung bình của chuỗi thời gian dường như không có sự biến động lớn, do đó chuỗi thời gian này có thể có tính dừng. Để chắc chắn hơn, ta sử dụng kiểm định Dickey – Fuller để kiểm tra tính dừng của chuỗi thời gian.

Trong phần mềm thống kê R, có thể dễ dàng sử dụng kiểm định Dickey – Fuller cho một chuỗi thời gian bằng hàm `adf.test()` trong thư viện `tseries`. Chi tiết về hàm này có thể tham khảo [24].

Kiểm tra tính dừng của chuỗi thời gian bằng kiểm định Dickey – Fuller:

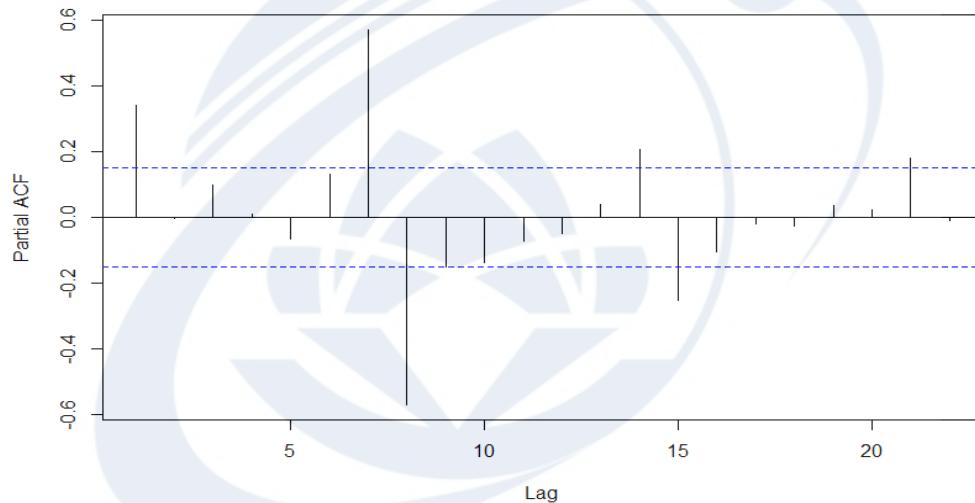
```
> library(tseries) # tải thư viện tseries  
> adf.test(ts, alternative = c("stationary"))      # ts  
là chuỗi thời gian  
  
Augmented Dickey-Fuller Test  
  
data: ts  
  
Dickey-Fuller = -5.1068, Lag order = 5, p-value = 0.01  
alternative hypothesis: stationary  
  
Warning message:  
In adf.test(timeseries, alternative = c("stationary")) :  
p-value smaller than printed p-value
```

Có thể thấy giá trị $p-value$ bé hơn 0.01 do đó không thể bác bỏ giả thiết *stationary* nên kết luận chuỗi thời gian có tính dừng.

Nhận dạng mô hình: Sau khi xác định chuỗi thời gian có tính dừng, có thể kết luận giá trị $d = 0$, tiếp theo là xác định các tham số còn lại p và q . Để xác định giá trị của các tham số p, q phụ thuộc vào đồ thị của hàm tự tương quan (ACF) và hàm tự tương quan riêng phần (PACF). Chọn các giá trị của p tại các độ trễ mà tại đó giá trị của hàm PACF khác 0 về mặt thống kê. Tương tự chọn các giá trị của q tại các độ trễ mà tại đó giá trị của hàm ACF khác 0 về mặt thống kê.

Sử dụng đồ thị hàm PACF để xác định giá trị của tham số p , trong phần mềm thống kê R đồ thị hàm PACF có thể được vẽ bằng hàm *pacf()*.

```
> pacf(trains) # trains là chuỗi thời gian
```

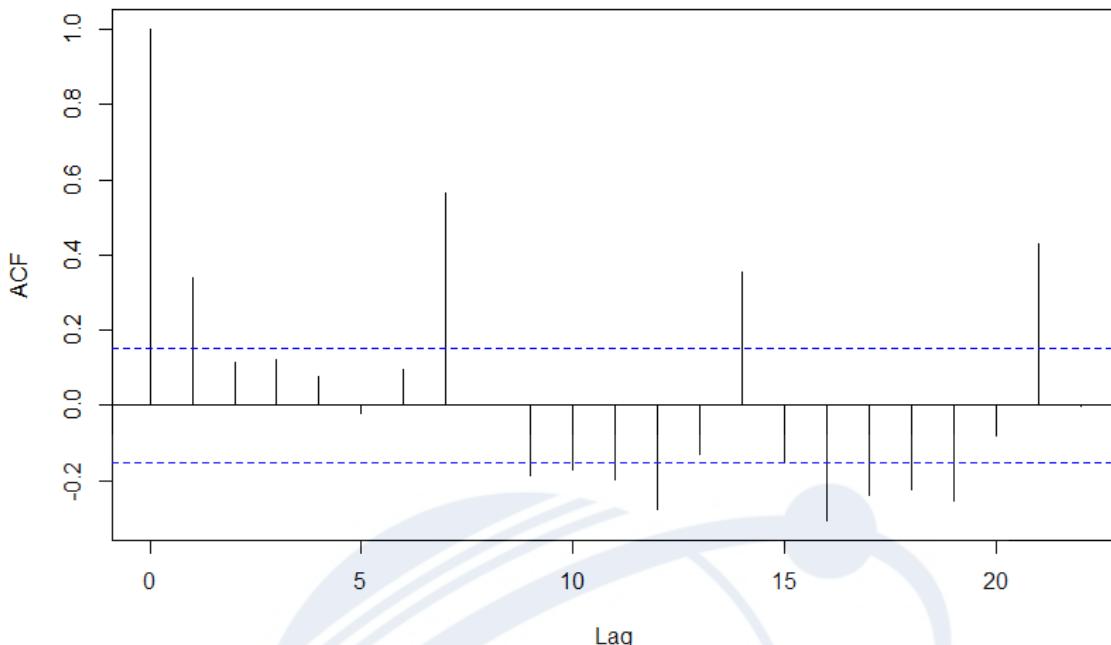


Hình 4.3. Đồ thị hàm PACF

Từ đồ thị của hàm PACF có thể thấy tại các độ trễ (Lag) 1, 7, 8, 14, 15 và 21 giá trị của hàm PACF khác 0 theo nghĩa thống kê, vậy các giá trị có thể của p là 1, 7, 8, 14, 15 và 21.

Tương tự, sử dụng đồ thị hàm ACF để xác định giá trị của tham số q , trong phần mềm thống kê R đồ thị hàm ACF có thể được vẽ bằng hàm *acf()*.

```
> acf(trains) # trains là chuỗi thời gian
```



Hình 4.4. Đồ thị hàm ACF

Từ đồ thị của hàm ACF có thể thấy tại các độ trễ (Lag) 0, 1, 7, 9, 10, 11, 12, 14, 16, 17, 18, 19 và 21 giá trị của hàm ACF khác 0 theo nghĩa thống kê, vậy các giá trị có thể của q là 0, 1, 7, 9, 10, 11, 12, 14, 16, 17, 18, 19 và 21.

Bằng cách tổ hợp các giá trị có thể có của p , d , q ta có thể xác định được các mô hình ARIMA. Số giá trị có thể có của p là 6, của q là 13 và d là 1, nên số mô hình ARIMA có thể có là 78 mô hình.

Ước lượng các tham số: Sau khi xác định các tham số p , d , q của mô hình ARIMA. Bước tiếp theo là ước lượng các số hạng tự hồi quy và các số hạng trung bình động. Trong phần mềm thống kê R các số hạng này được ước lượng tự động bằng hàm *arima()*. Ví dụ để xác định các số hạng của mô hình ARIMA(1, 0, 1) trong R thực hiện như sau:

```
> arima(trains, order = c(1,0,1)) # trains là chuỗi thời gian, order = c(1,0,1) là mô hình ARIMA(1, 0, 1)
```

Call:

```
arima(x = trains, order = c(1, 0, 1))
```

Coefficients:

Chương 4. Dự báo tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt

```

ar1      ma1      intercept
0.3384  0.0059    716.494
s.e.    0.3032  0.3321    31.504
sigma^2 estimated as 72914: log likelihood = -1186.01,
aic = 2380.03

```

Các số hạng $ar1 = 0.3384$ và $ma1 = 0.0059$ là lần lượt đại diện cho số hạng hồi quy và số hạng trung bình động của mô hình ARIMA(1, 0, 1). Tương tự như vậy ta áp dụng cho các mô hình ARIMA với các tham số p, q khác.

Kiểm định mô hình: Sau khi đã lựa chọn một mô hình ARIMA cụ thể và ước lượng các tham số của nó, phương pháp Box – Jenkins yêu cầu kiểm định và đánh giá mô hình được lựa chọn có phù hợp với dữ liệu hay không, vì có thể có một mô hình ARIMA khác với các tham số p, d, q khác phù hợp hơn với dữ liệu.

Các tiêu chuẩn BIC (Bayesian information criterion), tiêu chuẩn AIC (Akaike info criterion) hay ước lượng sai số chuẩn (Standard error of estimate - SEE) được dùng để lựa chọn một mô hình ARIMA phù hợp với dữ liệu chuỗi thời gian. Mô hình ARIMA nào có các giá trị này bé nhất thì mô hình ARIMA đó được chọn để làm mô hình cho dự báo. Các tiêu chuẩn dùng để đánh giá mô hình ARIMA [4]:

$$AIC(p) = n \ln \left(\frac{\hat{\sigma}_e^2}{n} \right) + 2p \quad (4.1)$$

$$BIC(p) = n \ln \left(\frac{\hat{\sigma}_e^2}{n} \right) + p + p \ln(n) \quad (4.2)$$

$$SEE = \sqrt{\frac{\hat{\sigma}_e^2}{n}} \quad (4.3)$$

Trong đó:

+ n là số điểm dữ liệu.

+ $\hat{\sigma}_e^2$ là tổng bình phương sai lệch giữa giá trị dự báo và giá trị thực tế.

+ p là số tham số của mô hình.

Chương 4. Dự báo tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt

Sau khi thử hợp các giá trị có thể của p , d , q , số mô hình ARIMA được thử nghiệm là 78 mô hình. Dùng phần mềm thống kê R để tính các giá trị BIC, AIC và SEE của các mô hình ARIMA, có 3 mô hình (được trình bày trong bảng 4.1) có các giá trị BIC, AIC và SEE bé nhất. Mô hình ARIMA phù hợp nhất với dữ liệu chuỗi thời gian sẽ được lựa chọn dựa trên kết quả dự báo của từng mô hình trong bước tiếp theo.

Bảng 4.1. Các giá trị tiêu chuẩn BIC, AIC và ước lượng sai số chuẩn SEE của các mô hình ARIMA

Mô hình	BIC	AIC	SEE
ARIMA(8, 0, 7)	2241.117	2187.909	315.205
ARIMA(14, 0, 9)	2252.569	2174.322	222.767
ARIMA(15, 0, 14)	2275.222	2178.196	161.310

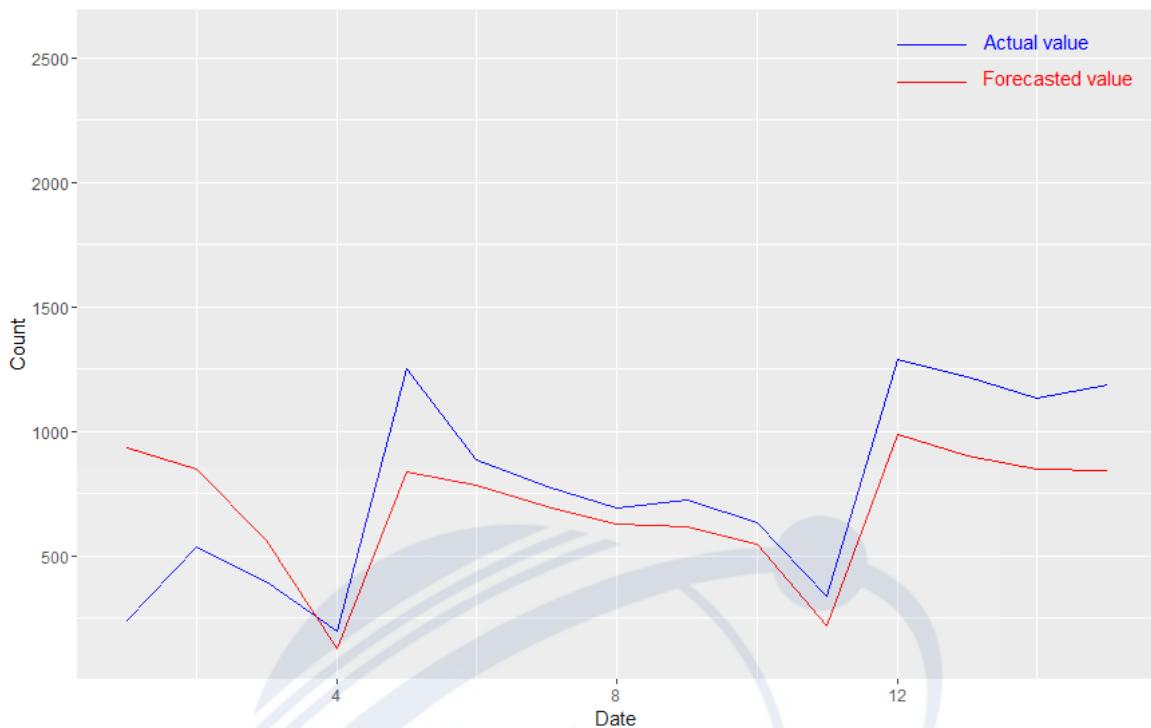
Chú thích: Các giá trị được tính bằng phần mềm thống kê R

Dự báo: Sau khi chọn được một mô hình ARIMA phù hợp nhất với dữ liệu chuỗi thời gian. Giá trị dự báo của chuỗi thời gian được tính toán dựa trên mô hình ARIMA và giá trị của chuỗi thời gian đó.

Trong phần mềm thống kê R giá trị dự báo của một mô hình ARIMA được thực hiện bằng hàm *predict()*, chi tiết về hàm này có thể tham khảo tại [25].

Ví dụ: dự báo 15 giá trị tiếp theo của chuỗi thời gian bằng mô hình ARIMA(1, 0, 1)

```
> arima_model <- arima(trains, order = c(1, 0, 1))
> pred <- predict(arima_model, n.ahead=15) # 15 giá trị
dự báo tiếp theo
```



Hình 4.5. Kết quả dự báo của mô hình ARIMA(21, 0, 19) bằng phần mềm thống kê R

Trong 3 mô hình ARIMA có các giá trị BIC, AIC và SEE bé nhất là ARIMA(8, 0, 7), ARIMA(14, 0, 9), ARIMA(15, 0, 14) thì mô hình ARIMA(15, 0, 14) cho kết quả dự báo tốt nhất dựa trên tiêu chí đánh giá của các độ đo. Chi tiết về các độ đo này sẽ được trình bày trong mục 4.4.1.

Bảng 4.2. Kết quả dự báo của các mô hình ARIMA

Mô hình	RMSE	MAE	MAPE
ARIMA(8, 0, 7)	293.440	262.099	0.258
ARIMA(14, 0, 9)	207.385	143.547	0.143
ARIMA(15, 0, 14)	150.172	107.276	0.096

Chú thích: Các giá trị được tính bằng phần mềm thống kê R

4.3.2 Dự báo thành phần phi tuyến tính bằng phương pháp SVM

Sau khi lựa chọn được một mô hình ARIMA phù hợp để dự báo thành phần tuyến tính của chuỗi thời gian. Bước thứ hai sử dụng phương pháp SVM trong ước lượng hồi quy để dự báo thành phần phi tuyến tính của chuỗi thời gian. Như đã đề cập trong chương 3, một chuỗi thời gian x_t thường bao gồm hai thành phần tuyến tính L_t và phi tuyến tính N_t .

$$x_t = L_t + N_t \quad (4.4)$$

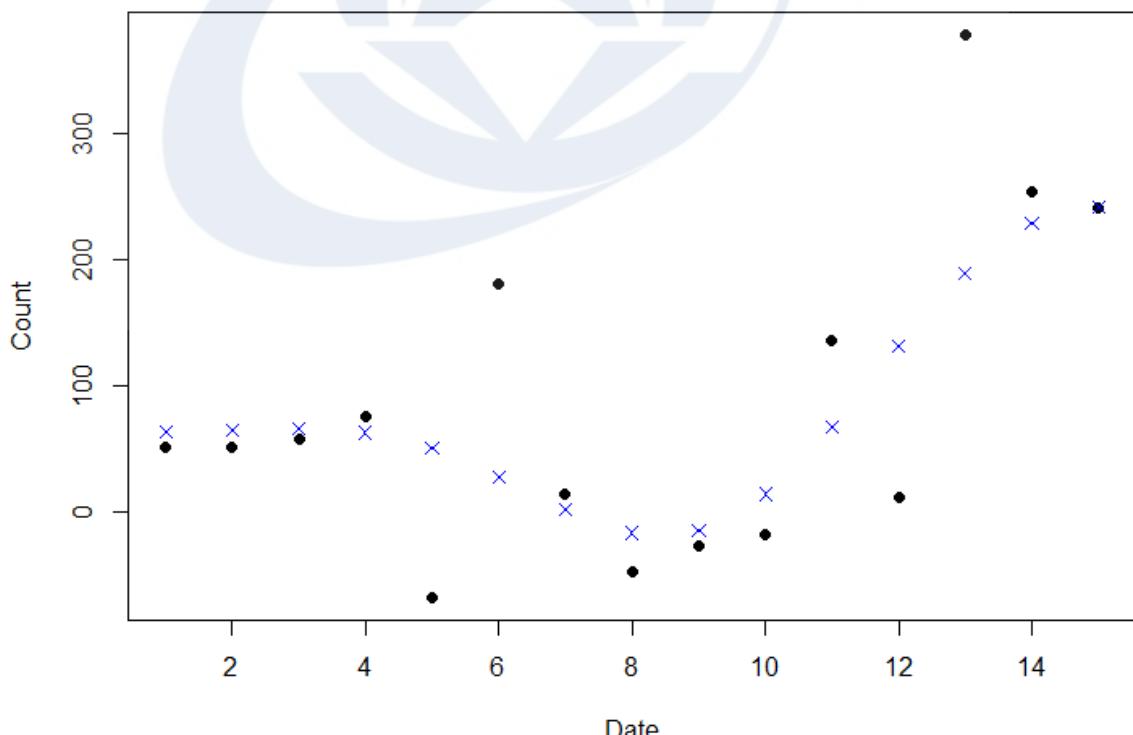
Nếu gọi \hat{L}_t là kết quả dự báo của mô hình ARIMA. Khi đó thành phần còn lại (residuals) e_t của chuỗi thời gian sau khi lấy kết quả thực tế trừ kết quả dự báo được xác định như sau:

$$e_t = x_t - \hat{L}_t \quad (4.5)$$

Thành phần còn lại e_t chứa trong nó thành phần phi tuyến tính N_t của chuỗi thời gian. Do đó phương pháp SVM trong ước lượng hồi quy được sử dụng để dự báo thành phần phi tuyến tính N_t dựa trên e_t .

Trong phần mềm thống kê R sử dụng hàm `svm()` của thư viện `e1071` để huấn luyện mô hình trước khi dự báo. Chi tiết về hàm `svm()` có thể tham khảo tại [23]

```
> model <- svm(Count ~ Date, train) # train là chuỗi thời gian
> pred <- predict(model)
```



Hình 4.6. Kết quả dự báo thành phần phi tuyến tính của chuỗi thời gian bằng phương pháp SVM

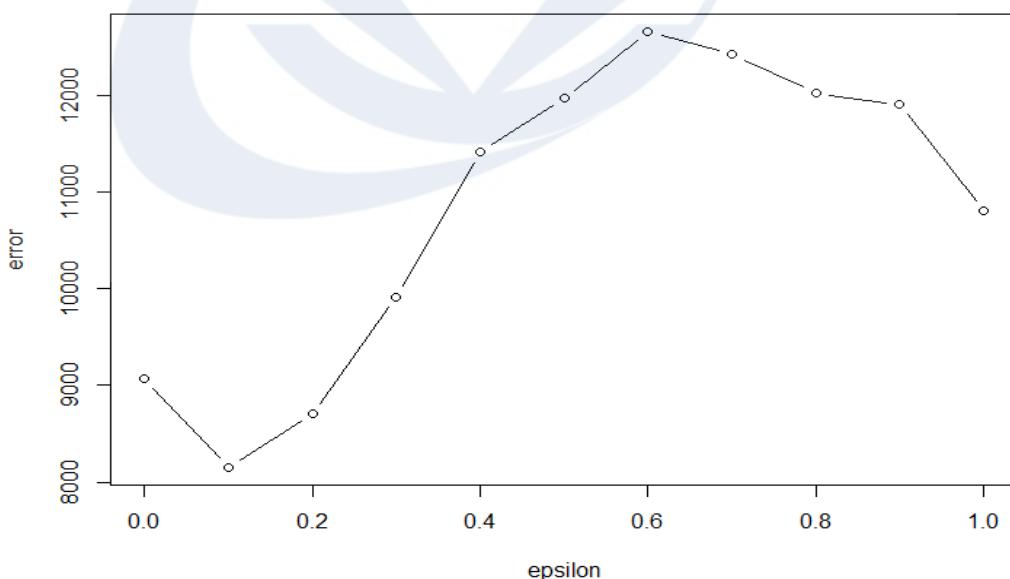
Chương 4. Dự báo tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt

Để tăng độ chính xác của kết quả dự báo, trong R hỗ trợ hàm *tune* của thư viện *e1071*. Chi tiết về hàm *tune()* có thể tham khảo tại [23], hàm này hỗ trợ nhiều tham số, trong đó có 2 tham số quan trọng là *epsilon* và *cost* giúp tăng độ chính xác của dự báo và điều chỉnh overfitting. Để chọn các giá trị thích hợp cho *epsilon* và *cost* ta sử dụng phương pháp grid search.

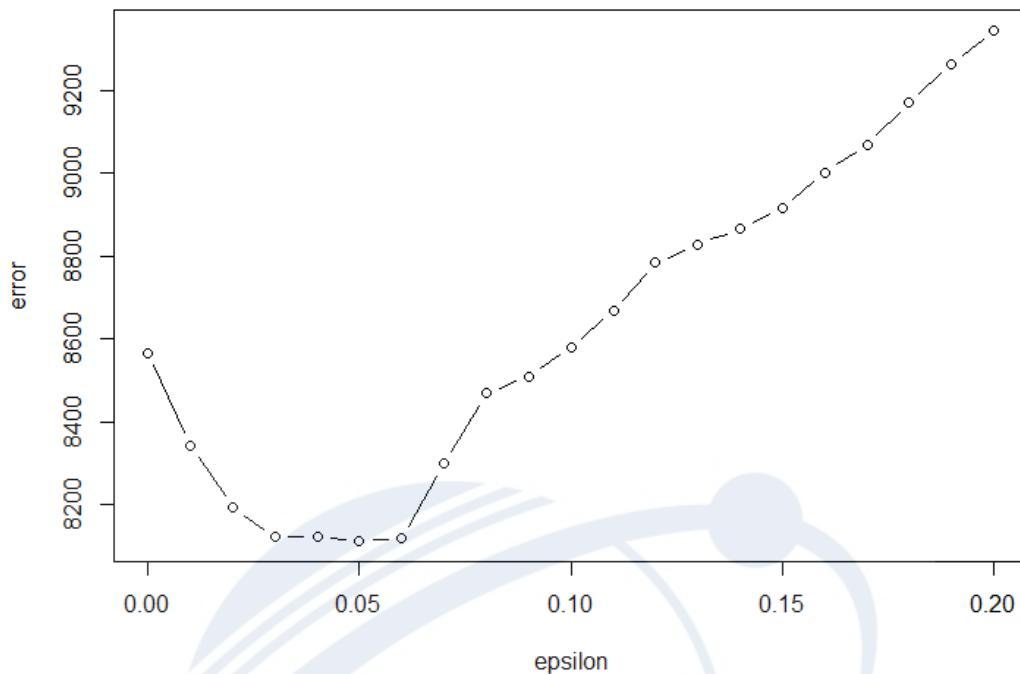
```
> tuneResult <- tune(svm, Count ~ Date, data = train,  
ranges = list(epsilon = seq(0,1,0.1)),cost=2^(2:9))  
  
> plot(tuneResult)
```

Đoạn chương trình trên khi thực hiện trong R cho kết quả như hình 4.7, có thể thấy giá trị của *epsilon* làm cho error nhỏ nhất nằm trong khoảng từ 0 đến 0.2. Do đó để chọn *epsilon* sao cho error là nhỏ nhất ta sẽ thu hẹp khoảng cách tìm kiếm từ 0 đến 0.2 với độ rộng là 0.01.

```
> tuneResult <- tune(svm, Count ~ Date, data = train,  
ranges = list(epsilon = seq(0,0.2,0.01)),cost=2^(2:9))  
  
> plot(tuneResult)
```



Hình 4.7. Kết quả khảo sát giá trị *epsilon* trong khoảng từ 0 đến 1 với độ rộng 0.1



Hình 4.8. Kết quả khảo sát giá trị *epsilon* trong khoảng từ 0 đến 0.2 với độ rộng 0.01

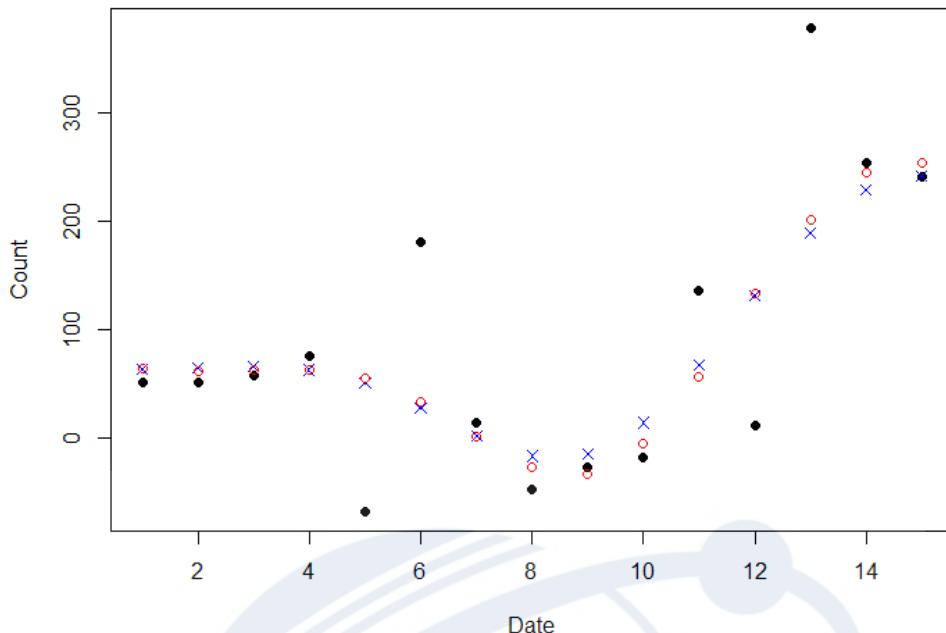
Từ kết quả khảo sát trên có thể thấy giá trị *epsilon* thích hợp nhất là 0.05. Khi đó mô hình tốt nhất để dự báo cho thành phần phi tuyến tính của chuỗi thời gian là:

```
> tuneResult <- tune(svm, Count ~ Date, data = train,
  ranges = list(epsilon = seq(0,0.2,0.01)),cost=2^(2:9))
> model <- tuneResult$best.model
> pred <- predict(model)
```

Bảng 4.3. Kết quả dự báo của các mô hình SVM

Mô hình	RMSE	MAE	MAPE
SVM	79.566	53.784	1.253
SVM with tune	76.827	49.077	1.136

Chú thích: Các giá trị được tính bằng phần mềm thống kê R



Hình 4.9. Kết quả dự báo thành phần phi tuyến tính của chuỗi thời gian bằng phương pháp SVM

4.3.3 Kết hợp các kết quả dự báo

Kết quả dự báo của mô hình kết hợp ARIMA và Support Vector Machine là tổng hợp kết quả dự báo của thành phần tuyến tính \widehat{L}_t bằng mô hình ARIMA và kết quả dự báo của thành phần phi tuyến tính \widehat{N}_t bằng phương pháp SVM trong ước lượng hồi quy.

$$\widehat{x}_t = \widehat{L}_t + \widehat{N}_t \quad (4.6)$$

Bằng cách cộng hai kết quả dự báo \widehat{L}_t và \widehat{N}_t lại ta thu được kết quả dự báo \widehat{x}_t sau cùng của mô hình kết hợp ARIMA và Support Vector Machine. \widehat{x}_t bao gồm kết quả dự báo cho cả hai thành phần tuyến tính và phi tuyến tính của chuỗi thời gian.

4.4 Kết quả dự báo và đánh giá

4.4.1 Giới thiệu các độ đo

Nhiều nghiên cứu và ứng dụng sử dụng các độ đo sau để đánh giá hiệu quả của các phương pháp, mô hình trong dự báo dữ liệu chuỗi thời gian.

Root Mean Square Error (RMSE): là một độ đo cho biết sự khác biệt giữa giá trị dự báo và giá trị thực tế của chuỗi thời gian. Giá trị RMSE càng bé thì mô hình cho kết quả dự báo càng chính xác [4].

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (\hat{x}_t - x_t)^2}{n}} \quad (4.7)$$

Mean Absolute Error (MAE): cũng giống như RMSE, MAE là một độ đo cho biết sự khác biệt giữa giá trị dự báo và giá trị thực tế của chuỗi thời gian. Giá trị MAE càng bé thì mô hình cho kết quả dự báo càng chính xác [4].

$$MAE = \frac{\sum_{t=1}^n |\hat{x}_t - x_t|}{n} \quad (4.8)$$

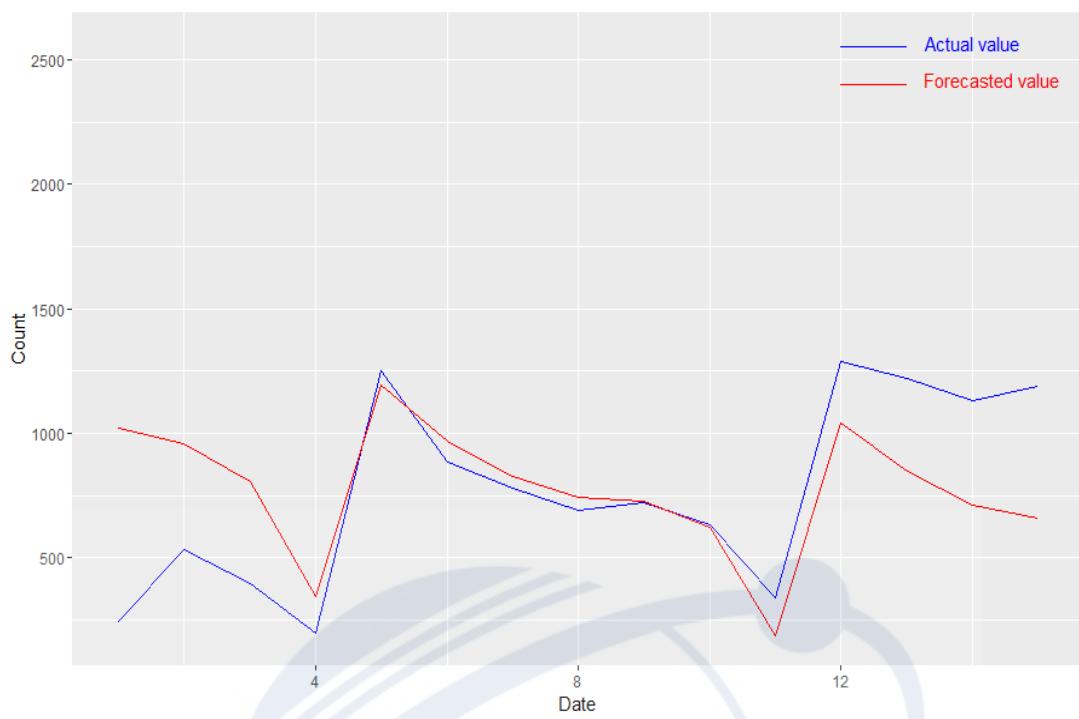
Mean Absolute Percentage Error (MAPE): là một độ đo cho biết tỉ lệ phần trăm sai lệch của kết quả dự báo và kết quả thực tế so với kết quả thực tế. Giá trị MAPE càng bé thì mô hình cho kết quả dự báo càng chính xác [4].

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|\hat{x}_t - x_t|}{x_t} \times 100 \quad (4.9)$$

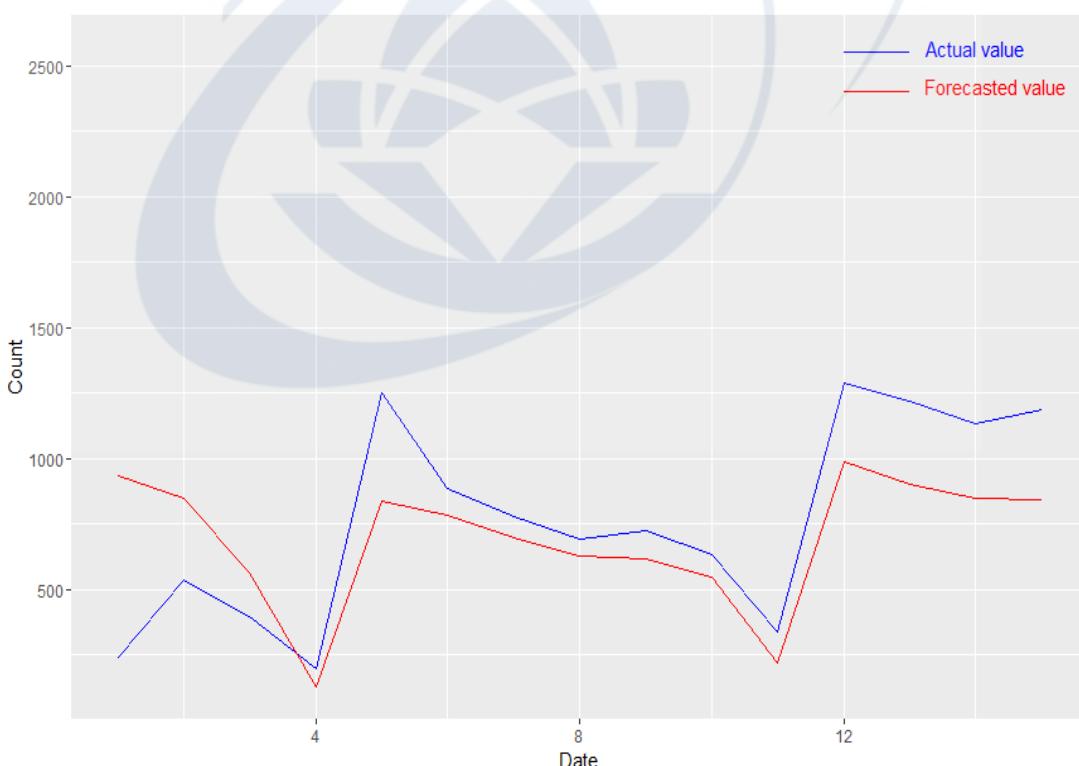
4.4.2 Kết quả dự báo và đánh giá

Để đánh giá hiệu quả dự báo của mô hình kết hợp ARIMA và Support Vector Machine, báo cáo sẽ trình bày kết quả dự báo của các mô hình tự hồi quy (AR), mô hình ARIMA, mô hình kết hợp ARIMA và mạng neural, mô hình kết hợp ARIMA và thuật giải di truyền trên cùng một tập dữ liệu. Hình 4.10 là kết quả dự báo bằng mô hình tự hồi quy (AR), hình 4.11 là kết quả dự báo bằng mô hình ARIMA, hình 4.12 là kết quả dự báo của mô hình kết hợp ARIMA và mạng neural, hình 4.13 là kết quả dự báo của mô hình kết hợp ARIMA và thuật giải di truyền, hình 4.14 là kết quả dự báo của mô hình kết hợp ARIMA và Support Vector Machine.

Chương 4. Dự báo tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt

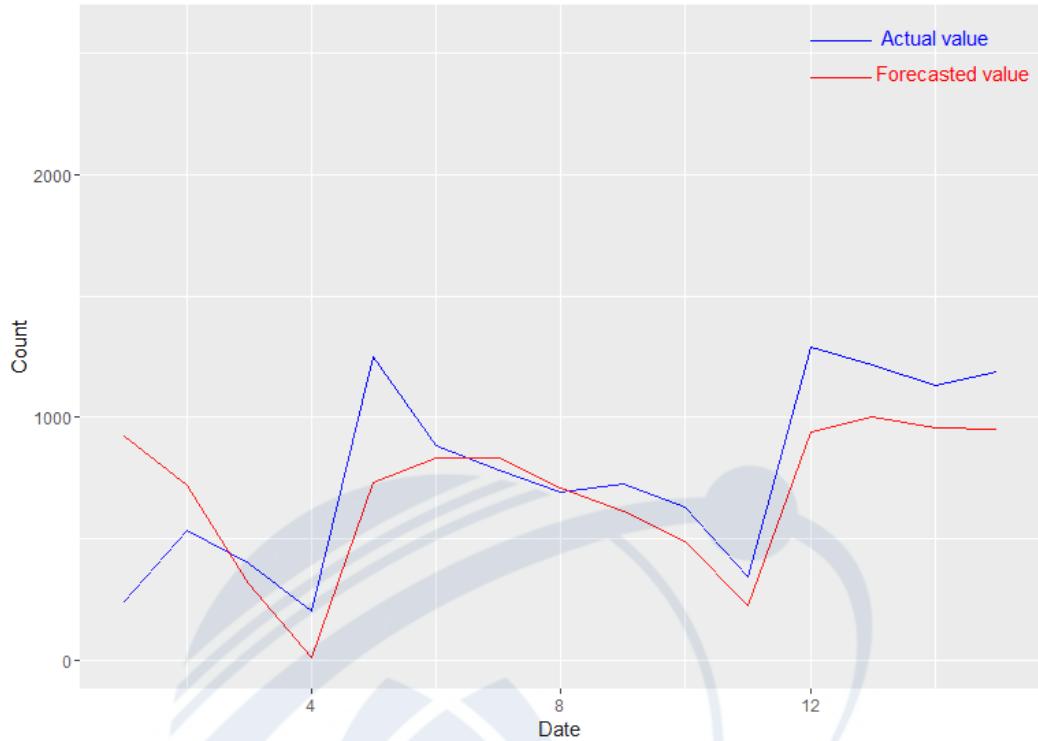


Hình 4.10. Kết quả dự báo của mô hình tự hồi quy

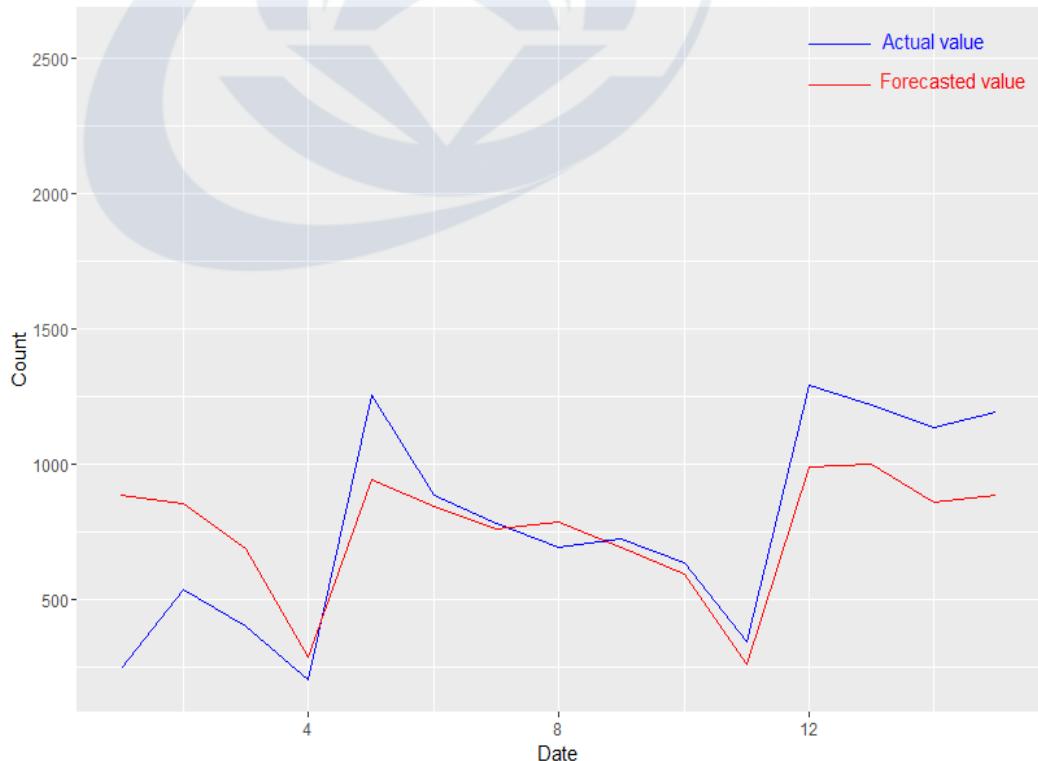


Hình 4.11. Kết quả dự báo của mô hình ARIMA

Chương 4. Dự báo tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt

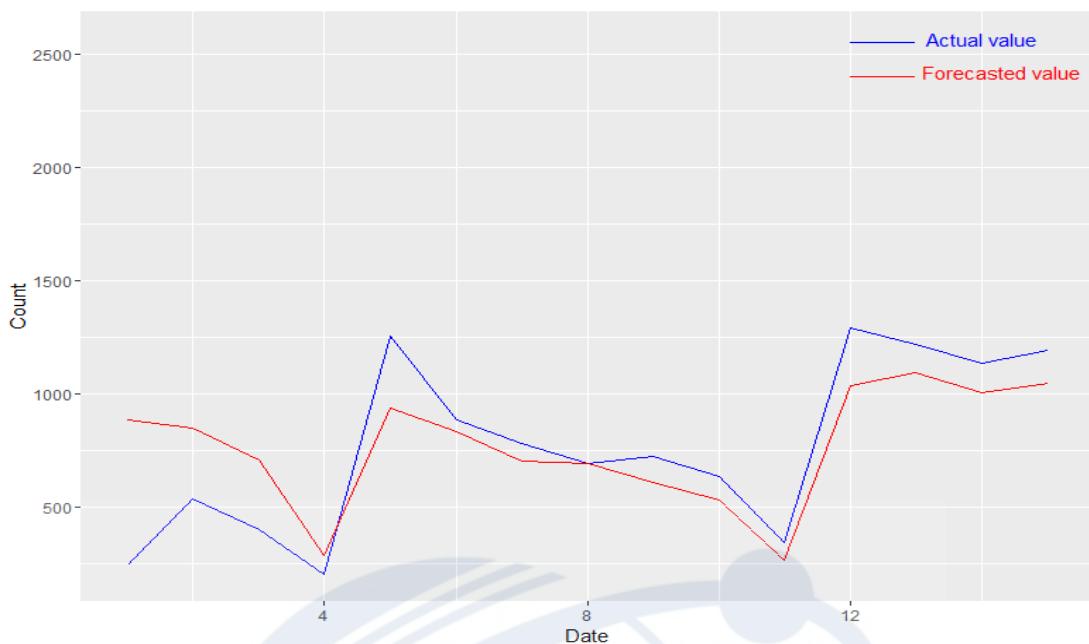


Hình 4.12. Kết quả dự báo của mô hình kết hợp ARIMA và mạng neural



Hình 4.13. Kết quả dự báo của mô hình kết hợp ARIMA

và thuật giải di truyền



Hình 4.14. Kết quả dự báo của mô hình kết hợp ARIMA và Support Vector Machine

Bảng 4.4. Kết quả dự báo của các mô hình

Mô hình	RMSE	MAE	MAPE
AR	332.747	248.124	0.519
ARIMA	285.315	230.821	0.435
ARIMA + NN	273.224	208.541	0.427
ARIMA + GA	261.428	203.677	0.406
ARIMA + SVM	240.723	182.843	0.394

Chú thích: Các giá trị được tính bằng phần mềm thống kê R

Bảng 4.4 là kết quả dự báo của các mô hình dựa trên các độ đo được trình bày trong mục 4.4.1. Từ kết quả dự báo này có thể thấy mô hình kết hợp ARIMA và Support Vector Machine cho kết quả dự báo tốt nhất trên cùng một tập dữ liệu so với các mô hình khác là mô hình tự hồi quy (AR), mô hình ARIMA, mô hình kết hợp ARIMA và mạng neural, mô hình kết hợp ARIMA và thuật giải di truyền. Điều đó chứng tỏ mô hình kết hợp ARIMA và Support Vector Machine phù hợp để dự báo cho dữ liệu tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt. Do đó có thể sử dụng mô hình kết hợp ARIMA và Support Vector Machine vào dự báo số lượng giao dịch theo từng ngày tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt.

Chương 5. KẾT LUẬN VÀ KHUYẾN NGHỊ

5.1 Kết luận

Kết quả dự báo số lượng giao dịch trên ngày tại Công ty Dịch vụ Trực tuyến Cộng Đồng Việt cung cấp thêm tính đúng đắn của hướng tiếp cận kết hợp các mô hình dự báo dữ liệu chuỗi thời gian nói chung và mô hình dự báo dữ liệu chuỗi thời gian kết hợp ARIMA và Support Vector Machine nói riêng.

Mô hình kết hợp ARIMA và Support Vector Machine thể hiện kết quả dự báo vượt trội hơn so với các mô hình khác như mô hình tự hồi quy (AR) hay mô hình ARIMA trong dự báo dữ liệu chuỗi thời gian. Phương pháp Support Vector Machine trong ước lượng hồi quy giúp tăng độ chính xác cho kết quả dự báo của mô hình ARIMA.

Lý do chính giúp kết quả dự báo của mô hình kết hợp ARIMA và Support Vector Machine vượt trội hơn so với các mô hình khác là do chuỗi thời gian trong thực tế thường bao gồm hai thành phần tuyến tính và phi tuyến tính. Nếu một mô hình dự báo chỉ có thể dự báo tốt cho một trong hai thành phần đó thì kết quả dự báo thường không sát với dữ liệu thực tế.

Mặc dù kết quả dự báo của mô hình kết hợp ARIMA và Support Vector Machine là vượt trội hơn so với các mô hình khác nhưng do đây là một mô hình kết hợp của hai mô hình khác nhau nên chi phí để xây dựng mô hình cũng lớn hơn so với các mô hình đơn lẻ khác. Bên cạnh đó thời gian dự báo của mô hình này cũng lớn hơn so với các mô hình khác do phải trải qua hai giai đoạn dự báo là giai đoạn dự báo thành phần tuyến tính bằng mô hình ARIMA và dự báo thành phần phi tuyến tính bằng phương pháp Support Vector Machine trong ước lượng hồi quy.

Về ý nghĩa thực tiễn, kết quả dự báo của mô hình kết hợp ARIMA và Support Vector Machine giúp ích cho Công ty Dịch vụ Trực tuyến Cộng Đồng Việt trong việc dự báo về số lượng giao dịch, số lượng khách hàng đến thanh

Chương 5. Kết luận và khuyến nghị

toán theo từng ngày từ đó có kế hoạch bố trí nhân sự sao cho phù hợp hoặc có thể tham khảo kết quả dự báo của mô hình để có các chiến lược kinh doanh và marketing hiệu quả vào từng thời điểm.

5.2 Khuyến nghị

Trong hầu hết các nghiên cứu hay ứng dụng về mô hình kết hợp ARIMA và các phương pháp máy học như Support Vector Machine, mạng neural, thuật giải di truyền, người ta luôn sử dụng mô hình ARIMA để dự báo thành phần tuyến tính của chuỗi thời gian trước khi sử dụng các phương pháp máy học để dự báo thành phần phi tuyến tính còn lại. Chưa có một nghiên cứu hay ứng dụng nào trong lĩnh vực này thực hiện ngược lại quá trình trên, tức là dự báo thành phần phi tuyến tính của chuỗi thời gian bằng các phương pháp máy học trước khi dự báo thành phần tuyến tính của chuỗi thời gian bằng mô hình ARIMA. Do đó đây có thể là một hướng tiếp cận mới cho mô hình kết hợp các phương pháp dự báo dữ liệu chuỗi thời gian khi thành phần phi tuyến tính của chuỗi thời gian được dự báo trước thành phần tuyến tính.

Bên cạnh đó, trong các mô hình kết hợp ARIMA và các phương pháp máy học, sự kết hợp của các phương pháp bên trong đó chưa thật sự sâu rộng, sự kết hợp này chỉ dừng lại ở việc tổng hợp các kết quả dự báo của các mô hình đơn lẻ lại với nhau để cho ra kết quả dự báo cuối cùng. Chẳng hạn như với mô hình kết hợp ARIMA và Support Vector Machine, sự kết hợp của hai mô hình ARIMA và Support Vector Machine chỉ thể hiện ở việc cộng hai kết quả dự báo của hai mô hình này lại với nhau để có kết quả dự báo cuối cùng, ngoài ra giữa hai mô hình này không có liên kết gì với nhau. Do đó để kết quả dự báo chuỗi thời gian hiệu quả hơn cần có sự kết hợp chặt chẽ giữa các mô hình sao cho các mô hình này có thể hỗ trợ cho nhau trong việc dự báo. Chính vì vậy mà vấn đề làm thế nào để kết hợp chặt chẽ các phương pháp dự báo trong các mô hình kết hợp cũng là một hướng phát triển của đề tài.

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] Đỗ Quang Giám, Vũ Thị Hân, Lý Thị Lan Phương, Nguyễn Thu Thủy (2012), “Xây dựng mô hình ARIMA cho dự báo khách du lịch Quốc tế đến Việt Nam”, Tạp chí Khoa học và Phát triển, tập 10 (2), tr. 364–370.
- [2] Nguyễn Thị Kim Loan (2009), “Mô hình chuỗi thời gian mờ trong dự báo chuỗi thời gian”, Luận văn Thạc sĩ Công nghệ thông tin, ĐH Thái Nguyên.
- [3] Bùi Quang Trung, Nguyễn Quang Minh Nhi, Lê Văn Hiếu, Nguyễn Hồ Diệu Uyên (2010), “Ứng dụng mô hình ARIMA để dự báo VNINDEX”, Tuyển tập Báo cáo Hội nghị Sinh viên Nghiên cứu Khoa học lần thứ 7 Đại học Đà Nẵng.

Tiếng Anh

- [4] Ratnadip Adhikari, R. K. Agrawal (2013), “An Introductory Study on Time Series Modeling and Forecasting”, LAP Lambert Academic Publishing, Germany.
- [5] Ayodele A. Adebiyi, Aderemi O. Adewumi, Charles K. Ayo (2014), “Stock Price Prediction Using the ARIMA Model”, 16th International Conference on Computer Modelling and Simulation.
- [6] Peter J. Brockwell, Richard A. Davis (2002), “Introduction to Time Series and Forecasting”, Springer-Verlag, USA.
- [7] L. J. Cao, Francis E. H. Tay (2003), “Support Vector Machine With Adaptive Parameters in Financial Time Series Forecasting”, IEEE Transactions on Neural Networks, pp. 1506-1518.
- [8] Arghya Ghosh, Satyendra Nath Mandal, Subhojit Roy, J. Pal Choudhury, S. R. Bhadra Chaudhuri (2012), “A Novel Approach of Genetic Algorithm in prediction of Time Series Data”, Special Issue of International Journal of Computer Applications.
- [9] Keith W. Hipel, A. Ian McLeod (1994), “Time Series Modelling of Water Resources and Environmental Systems”, Amsterdam, Elsevier.

- [10] Joarder Kamruzzaman, Rezaul Begg, Ruhul Sarker (2006), “*Artificial Neural Networks in Finance and Manufacturing*”, Idea Group Publishing, USA.
- [11] Christoph Klose, Marion Pircher, Stephan Sharma for 406347/UK “Ökonometrische Prognose” in SS04 (2004), “*Univariate Time Series Forecasting*”, University of Vienna Department of Economics.
- [12] Wei Ming, Yukun Bao, Zhongyi Hu, and Tao Xiong (2014), “*Multistep-Ahead Air Passengers Traffic Prediction with Hybrid ARIMA-SVMs Models*”, The Scientific World Journal.
- [13] Hongzhan Nie , Guohui Liu , Xiaoman Liu , Yong Wang (2012), “*Hybrid of ARIMA and SVMs for Short-Term Load Forecasting*”, International Conference on Future Energy, Environment, and Materials, pp. 1455-1460.
- [14] Ping-Feng Pai, Chih-Sheng Lin (2004), “*A hybrid ARIMA and support vector machines model in stock price forecasting*”, OMEGA The International Journal of Management Science.
- [15] Baxter Tyson Smith, B.Sc., B.Eng., Ph.D. Candidate (2014), “*Lagrange Multipliers Tutorial in the Context of Support Vector Machines*”, Faculty of Engineering and Applied Science Memorial University of Newfoundland St. John’s, Newfoundland, Canada.
- [16] Alex J Smola, Bernhard Scholkopf (1998), “*A Tutorial on Support Vector Regression*”, NeuroCOLT Technical Report Series.
- [17] Qiang Song, Brad S. Chissom (1993), “*Fuzzy Time Series and Its Models*”, Fuzzy Sets and Systems, pp. 269-277.
- [18] Qiang Song, Brad S. Chissom (1993), “*Forecasting Enrollments with Fuzzy Time Series*”, Fuzzy Sets and Systems, pp. 1-9.
- [19] Thoranin Sujjaviriyasup, Komkrit Pitiruek (2013), “*Hybrid ARIMA-Support Vector Machine Model for Agricultural Production Planning*”, Applied Mathematical Sciences, Vol. 7 (53), pp. 2833–2840

- [20] Fang-Mei Tseng, Gwo-Hshiung Tzeng, Hsiao-Cheng Yu, Benjamin J.C. Yuan (2001), “*Fuzzy ARIMA model for forecasting the foreign exchange market*”, Fuzzy Sets and Systems, pp. 9–19.
- [21] G. Peter Zhang (2003), “*Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model*”, Neurocomputing, pp. 159 – 175.
- [22] Anallyz (Oct. 2016), <http://www.anallyz.com/timeseries.html>
- [23] CRAN (Dec. 2016), <https://cran.r-project.org/web/packages/e1071/e1071.pdf>
- [24] CRAN (Dec. 2016), <ftp://cran.r-project.org/pub/R/web/packages/tseries/tseries.pdf>
- [25] ETHZurich, Department of Mathematics, (Dec. 2016), <https://stat.ethz.ch/R-manual/R-devel/library/stats/html/predict.arima.html>
- [26] Quora (Nov. 2016), <https://www.quora.com/What-are-Kernels-in-Machine-Learning-and-SVM>
- [27] Scribd (Nov. 2016), <https://www.scribd.com/document/252656326/Time-Series-Forecasting-by-Using-Wavelet-Kernel-SVM>
- [28] SVM Tutorial (Nov. 2016), <http://www.svm-tutorial.com/2014/11>
- [29] SVMs (Nov. 2016), <http://www.svms.org/regression/MSRS00.pdf>
- [30] TeX (Dec. 2016), <http://tex.stackexchange.com/questions/132444/diagram-of-an-artificial-neural-network>
- [31] Univesity of Florida. The Foundation for The Gator Nation (Nov. 2016), http://www.cise.ufl.edu/class/cis4930sp11dtm/notes/intro_svm_new.pdf

PHỤ LỤC

MÃ NGUỒN

Báo cáo sử dụng phần mềm thống kê R phiên bản 3.2.4 để tính toán các giá trị và cài đặt chương trình. Phần này trình bày các đoạn mã nguồn được thực hiện trong báo cáo.

Mã nguồn dự báo chuỗi thời gian bằng mô hình tự hồi quy (AR)

```
## AutoRegression
# Load library
library(hydroGOF)
library(ggplot2)
# Read file
train <- read.csv('D:/train.csv')
test <- read.csv('D:/test.csv')

# Forecast
burg <- ar(train$Count,method="burg")
pred.burg <- predict(burg,n.ahead=15)

# Measured
RMSE <- rmse(pred.burg$pred,test$Count); RMSE
MAE <- mae(pred.burg$pred,test$Count); MAE
MAPE <- mean(abs((test$Count -
pred.burg$pred)/test$Count)); MAPE

# Plot
df <- data.frame(test>Date
test$Count,pred.burg$pred)
g <- ggplot(df,aes(x=test>Date,y=test$Count))
g <- g + geom_line(aes(y=test$Count),
colour="blue")
g <- g + geom_line(aes(y=pred.burg$pred),
colour="red")
g <- g + ylab("Count") + xlab("Date")
```

```

g1 <- g +
annotate("segment",x=12,xend=13,y=2550,yend=2550,
colour="blue")

g1 <- g1 +
annotate("text",x=14,y=2570,label="Actual
value",colour="blue")

g2 <- g1 +
annotate("segment",x=12,xend=13,y=2400,yend=2400,
colour="red")

g2 <- g2 +
annotate("text",x=14.3,y=2420,label="Forecasted
value",colour="red")

g2

```

Mã nguồn dự báo chuỗi thời gian bằng mô hình ARIMA(21, 0, 19)

```

## Auto Regression Integrated Moving Average

# Load library

library(hydroGOF)

library(ggplot2)

library(tseries)

library(nlme)

# Read file

train <- read.csv('D:/Train.csv')

test <- read.csv('D:/Test.csv')

# Forecast

arima <- arima(train$Count,order = c(21,0,19))

pred.arima <- predict(arima,n.ahead=15)

#BIC <- BIC(arima); BIC

#AIC <- AIC(arima); AIC

#SEE <- sqrt(sum((test$Count -
pred.arima$pred)^2) / (15-2)); SEE

# Measured

RMSE <- rmse(pred.arima$pred,test$Count); RMSE

MAE <- mae(pred.arima$pred,test$Count); MAE

```

```

MAPE <- mean(abs((test$Count -
pred.arima$pred)/test$Count)); MAPE

# Plot

df <-
data.frame(test$Date,test$Count,pred.arima$pred)
g <- ggplot(df,aes(x=test$Count,y=test$Count))
g <- g + geom_line(aes(y=test$Count),
colour="blue")
g <- g + geom_line(aes(y=pred.arima$pred),
colour="red")
g <- g + ylab("Count") + xlab("Date")
g1 <- g +
annotate("segment",x=12,xend=13,y=2550,yend=2550,
colour="blue")
g1 <- g1 +
annotate("text",x=14,y=2570,label="Actual
value",colour="blue")
g2 <- g1 +
annotate("segment",x=12,xend=13,y=2400,yend=2400,
colour="red")
g2 <- g2 +
annotate("text",x=14.3,y=2420,label="Forecasted
value",colour="red")
g2

```

Mã nguồn dự báo chuỗi thời gian bằng mô hình kết hợp ARIMA và mạng neural

```

## Hybrid Model (ARIMA + NN)

# Load library

library(hydroGOF)
library(ggplot2)
library(e1071)
library(neuralnet)

# Read file

train <- read.csv('D:/Train.csv')

```

```

train_nn <- read.csv('D:/TestARIMA.csv')
test <- read.csv('D:/Test.csv')
# Forecasting use ARIMA
arima <- arima(train[,2],order = c(15,0,14))
pred.arima <- predict(arima,n.ahead=15)
# Calculated Residuals
residuals <- train_nn[,2] - pred.arima$pred;
# Training NN
# Data preprocessing
t3 = residuals[1:12]
t2 = residuals[2:13]
t1 = residuals[3:14]
result = residuals[4:15]
df = data.frame(t1,t2,t3,result)
# Scale data from 0 to 1
maxs <- apply(df, 2, max)
mins <- apply(df, 2, min)
scaled.data <- as.data.frame(scale(df,center =
mins, scale = maxs - mins))
# Training
feats <- names(scaled.data[1:3])
formula <- paste(feats,collapse=' + ')
formula <- paste('result ~',formula)
nn <-
neuralnet(formula,scaled.data,hidden=c(2),linear.
output=TRUE)
#plot(nn)
# Continue forecasting ARIMA
pred.arima <- predict(arima,n.ahead=30)
pred.arima <- pred.arima$pred[16:30]
# Calculated Residuals

```

```

residuals_pre <- test[,2] - pred.arima;
# Forecasting use NN
# Data preprocessing
t3 = c(tail(t2,n=1), tail(t1,n=1),
tail(result,n=1), residuals_pre[1:12])
t2 = c(tail(t1,n=1), tail(result,n=1),
residuals_pre[1:13])
t1 = c(tail(result,n=1), residuals_pre[1:14])
result = residuals_pre
df = data.frame(t1,t2,t3,result)
# Scale data from 0 to 1
maxs <- apply(df, 2, max)
mins <- apply(df, 2, min)
scaled.data <- as.data.frame(scale(df,center =
mins, scale = maxs - mins))

# Forecasting
pr.nn <- compute(nn,scaled.data[,1:3])
# Scale back
pred.nn <- pr.nn$net.result*(max(df$result)-
min(df$result))+min(df$result)
# Combine two results
pred <- pred.arima + pred.nn[,1];
# Measured
RMSE <- rmse(pred,test[,2]); RMSE
MAE <- mae(pred,test[,2]); MAE
MAPE <- mean(abs((test[,2] - pred)/test[,2]));
MAPE
# Plot
df <- data.frame(test[,1],test[,2],pred)
g <- ggplot(df,aes(x=test[,1],y=test[,2]))
g <- g + geom_line(aes(y=test[,2]),
colour="blue")

```

```

g <- g + geom_line(aes(y=pred), colour="red")
g <- g + ylab("Count") + xlab("Date")
g1 <- g +
annotate("segment",x=12,xend=13,y=2550,yend=2550,
colour="blue")
g1 <- g1 +
annotate("text",x=14,y=2570,label="Actual
value",colour="blue")
g2 <- g1 +
annotate("segment",x=12,xend=13,y=2400,yend=2400,
colour="red")
g2 <- g2 +
annotate("text",x=14.3,y=2420,label="Forecasted
value",colour="red")
g2

```

Mã nguồn dự báo chuỗi thời gian bằng mô hình kết hợp ARIMA và thuật giải di truyền

```

## Hybrid Model (ARIMA + GA)
# Load library
library(hydroGOF)
library(ggplot2)
library(e1071)
library(genalg)
# Read file
train <- read.csv('D:/Train.csv')
train_ga <- read.csv('D:/TestARIMA.csv')
test <- read.csv('D:/Test.csv')
# Forecasting use ARIMA
arima <- arima(train[,2],order = c(15,0,14))
pred.arima <- predict(arima,n.ahead=15)

# Calculated Residuals

```

```

residuals_train <- train_ga[,2] -
pred.arima$pred;

# Continue forecasting ARIMA
pred.arima <- predict(arima,n.ahead=30)
pred.arima <- pred.arima$pred[16:30]
residuals_test <- test[,2] - pred.arima;
# Forecasting use GA
evalFunc <- function(x) {
  current_predict <- (x %*% data)/15
  error = actual - current_predict
  return(error)
}
residuals = c(residuals_train,residuals_test)
pred.ga = c(1:15)
for(i in 1:15)
{
  data = residuals[i:(i+14)]
  actual = residuals[(i+15)]
  GAmode1 <- rbga.bin(size = 15, popSize = 20,
  iters = 10, mutationChance = 0.01,
  elitism = T, evalFunc = evalFunc)
  pred.ga[i] = actual - GAmode1$best[10]
}
#Combine two results
pred <- pred.arima + pred.ga;
# Measured
RMSE <- rmse(pred,test[,2]); RMSE
MAE <- mae(pred,test[,2]); MAE
MAPE <- mean(abs((test[,2] - pred)/test[,2]));
MAPE
# Plot

```

```

df <- data.frame(test[,1],test[,2],pred)
g <- ggplot(df,aes(x=test[,1],y=test[,2]))
g <- g + geom_line(aes(y=test[,2]),
colour="blue")
g <- g + geom_line(aes(y=pred), colour="red")
g <- g + ylab("Count") + xlab("Date")
g1 <- g +
annotate("segment",x=12,xend=13,y=2550,yend=2550,
colour="blue")
g1 <- g1 +
annotate("text",x=14,y=2570,label="Actual
value",colour="blue")
g2 <- g1 +
annotate("segment",x=12,xend=13,y=2400,yend=2400,
colour="red")
g2 <- g2 +
annotate("text",x=14.3,y=2420,label="Forecasted
value",colour="red")
g2

```

Mã nguồn dự báo chuỗi thời gian bằng mô hình kết hợp ARIMA và SVM

```

## Hybrid Model
# Load library
library(hydroGOF)
library(ggplot2)
library(e1071)
# Read file
train <- read.csv('D:/Train.csv')
train_svm <- read.csv('D:/TestARIMA.csv')
test <- read.csv('D:/Test.csv')
# Forecasting use ARIMA
arima <- arima(train$Count,order = c(15,0,14))
pred.arima <- predict(arima,n.ahead=15)
# Calculated Residuals

```

```

residuals <- train_svm$Count - pred.arima$pred;
# Forecasting use SVM
index <- 1:15
svm <- svm(residuals ~ index,residuals)
pred.svm <- predict(svm)
# Continue forecasting ARIMA
pred.arima <- predict(arima,n.ahead=30)
pred.arima <- pred.arima$pred[16:30]
# Combine two results
pred <- pred.arima + pred.svm;
# Measured
RMSE <- rmse(pred,test$Count); RMSE
MAE <- mae(pred,test$Count); MAE
MAPE <- mean(abs((test$Count -
pred)/test$Count)); MAPE
# Plot
df <- data.frame(test$Date,test$Count,pred)
g <- ggplot(df,aes(x=test$Date,y=test$Count))
g <- g + geom_line(aes(y=test$Count),
colour="blue")
g <- g + geom_line(aes(y=pred), colour="red")
g <- g + ylab("Count") + xlab("Date")
g1 <- g +
annotate("segment",x=12,xend=13,y=2550,yend=2550,
colour="blue")
g1 <- g1 +
annotate("text",x=14,y=2570,label="Actual
value",colour="blue")
g2 <- g1 +
annotate("segment",x=12,xend=13,y=2400,yend=2400,
colour="red")
g2 <- g2 +
annotate("text",x=14.3,y=2420,label="Forecasted
value",colour="red") g2

```