Nội dung

[TÓM TẮT 3](#_Toc279404221)

[NỀN TẢNG LÝ THUYẾT 3](#_Toc279404222)

[1. Grid Search 3](#_Toc279404223)

[Thuật toán 3](#_Toc279404224)

[2. Pattern Search 4](#_Toc279404225)

[Sơ bộ về ý tưởng 4](#_Toc279404226)

[Thuật toán 5](#_Toc279404227)

[Các ý quan trọng về thuật toán Pattern Search 6](#_Toc279404228)

[CÀI ĐẶT 6](#_Toc279404229)

[1. Grid Search 7](#_Toc279404230)

[2. Pattern Search 7](#_Toc279404231)

[DỮ LIỆU 7](#_Toc279404232)

[1. Kịch bản 7](#_Toc279404233)

[2. Đầu vào 8](#_Toc279404234)

[THÍ NGHIỆM 8](#_Toc279404235)

[1. Phần SVR 8](#_Toc279404236)

[2. Phần ANN 14](#_Toc279404237)

[3. ANN và SVR 19](#_Toc279404238)

[KẾT LUẬN 20](#_Toc279404239)

[CÔNG VIỆC TƯƠNG LAI 21](#_Toc279404240)

[TÀI LIỆU THAM KHẢO 21](#_Toc279404241)

Báo cáo kết quả dự đoán xu hướng

# TÓM TẮT

Mục đích của báo cáo là đưa ra sự so sánh ban đầu về kết quả của mô hình cài đặt với kết quả mà các nghiên cứu trước đã có được. Ngoài ra, nhóm cũng thử nghiệm với mã chứng khoán của Việt Nam để phần nào hình dung về dữ liệu của thị trường này. Từ đó, xem xét và nhận định về sự phù hợp của mô hình, những điểm yếu cần khắc phục và những cải tiên trong thời gian sắp tới.

Thêm vào đó, đối với SVR, nhóm cũng xem xét phương pháp grid search và pattern search (do Kiên đề xuất) cho việc chọn tham số.

# NỀN TẢNG LÝ THUYẾT

## Grid Search

Grid Search chẳng qua là vét cạn không gian tìm kiếm. Đầu tiên, ta cần phải có miền giá trị của các biến cần tìm kiếm. Chẳng hạn, trong không gian hai chiều, đã biết . Kế đến, ta cần phải chọn một hệ số Delta nào đó để rời rạc hóa các miền giá trị này. Chẳng hạn, Delta = 2, ta sẽ xét x = 22, 24, 26, …210 và y = 2-5, 2-3, 2-1, …, 25. Sau đó, ta sẽ xét hết tất cả các trường hợp có thể (ở đây là 5\*6 trường hợp) để chọn ra cặp biến tốt nhất.

Ta nhận thấy hệ số rời rạc hóa Delta càng lớn thì sẽ làm giảm thời gian tính toán, nhưng ngược lại, độ chính xác của nó cũng sẽ giảm đi.

Grid Search là phương pháp truyền thống dùng để chọn bộ tham số cho SVM. Từ nhược điểm cố hữu của nó là quá tốn chi phí, người ta mới đề ra cải tiến như sau:

* Đầu tiên, ta sẽ dùng một grid thưa (Delta lớn) để chọn ra bộ tham số tốt nhất.
* Sau đó, ta sẽ làm kỹ hơn: ta sẽ dùng một grid dày hơn (Delta nhỏ hơn) xung quanh bộ tham số vừa tìm được ở trên.

### Thuật toán

#### Bước 1: Khởi tạo

* Giá trị bắt đầu của C, gamma, epsilon.
* Số interval (số này thể hiện số điểm grid), chẳng hạn nếu chọn 7 thì sẽ có 7\*7\*7 điểm cần search.
* Delta

#### Bước 2: Search thưa

1. Ứng với từng bộ ba C, gamma, epsilon, thực hiện cross validation để tìm giá trị lỗi
2. Cập nhật bộ ba mới nếu độ lỗi thấp hơn độ lỗi nhỏ nhất hiện tại.
3. Tăng giá trị của từng tham số theo delta
4. Quay lại bước 1 nếu chưa hết interval của cả 3 tham số.

#### Bước 3: Thay đổi giá trị khởi tạo

Để chuẩn bị cho việc search kỹ:

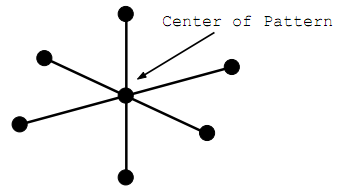
* Gán lại giá trị bắt đầu bằng giá trị lân cận của bộ ba có độ lỗi nhỏ nhất
* Giảm delta

#### Bước 4: Thực hiện search kỹvới các công đoạn như bước 2.

## Pattern Search

### Sơ bộ về ý tưởng

Đầu tiên, ta cần hiểu thế nào là pattern? Pattern đơn giản chỉ là một nhóm các điểm kề cận nhau trong không gian, trong đó có một điểm ở giữa được gọi là trung tâm của pattern.



Ta sẽ bắt đầu thuật toán từ một điểm bất kỳ trong không gian tìm kiếm. Điểm này được coi là trung tâm của pattern và là điểm “tốt nhất” cho đến thời điểm hiện tại.

Từ điểm trung tâm này, ta sẽ tính giá trị của hàm mục tiêu tại các điềm kề cận nhằm tìm ra được một điểm tốt hơn điểm hiện tại. Các điểm kề cận được với điểm trung tâm được định bởi **hướng** (luôn không đổi) và một thành phần để xác định **khoảng cách** gọi là bước tìm kiếm (search step.)

Nếu tìm được, ta sẽ gán lại trung tâm của pattern là điểm tốt hơn đó. Ngược lại, ta sẽ giảm bước tìm kiếm và lặp lại quá trình trên. Thuật toán sẽ ngừng khi bước tìm kiếm nhỏ hơn một ngưỡng nào đó.

### Thuật toán

* Bước 1: Khởi tạo
  + k = 1
  + Chọn bước tìm kiếm ban đầu và ngưỡng của bước tìm kiếm để dừng thuật toán .
  + Chọn ngẫu nhiên điểm trung tâm xk của pattern, tính f(xk) và gán min = f(xk).
  + BestPoint = xk.
* Bước 2: Nếu , dừng thuật toán!
* Bước 3: For i = 1, …, n//n là số chiều của không gian tìm kiếm
  + và . Tính f()// là dòng thứ i của ma trận đơn vị In
  + Nếu f() < min:
    - min = f()
    - BestPoint =
  + Ngược lại://Ta đổi ngược hướng tìm kiếm với hy vọng sẽ tìm được điểm tốt hơn
    - và . Tính f()
    - Nếu f() < min:
      * min = f()
      * BestPoint =
* Bước 4:
  + Nếu BestPoint == //Các điểm kế cận không tốt hơn điểm trung tâm 🡪giảm bước tìm kiếm
  + Ngược lại://Tìm được điểm tốt hơn điểm trung tâm 🡪Gán lại trung tâm của pattern là điểm tốt hơn đó
    - = BestPoint
  + Quay lại bước 2

### Các ý quan trọng về thuật toán Pattern Search

* Thuật toán đơn giản, dễ hiểu, dễ cài đặt, chi phí thấp.
* Khác với cách truyền thống là lấy đạo hàm của hàm mục tiêu (dở khi hàm mục tiêu không “convex”, có nhiều điểm cực trị địa phương; không làm được khi hàm mục tiêu không khả vi), ở đây ta không cần biết hàm mục tiêu có liên tục và khả vi hay không.
* Ngươi ta đã chứng minh được rằng thuật toán hội tụ về cực trị đia phương (dễ hiểu vì ta thấy rằng sau mỗi vòng lặp giá trị hàm mục tiêu tại điểm trung tâm pattern đều giảm hoặc không đổi) và hơn nữa, điểm cực trị này thường khá tốt.

# CÀI ĐẶT

Ở đây, ta chỉ xét SVR với hàm lỗi và hàm Gaussian kernel.

Với mô hình này, ta cần tìm 3 tham số: C, gamma (của hàm Gauss), và epsilon.

Căn cứ vào các báo cáo và theo kinh nghiệm, ta chọn miền giá trị cho ba biến này như sau:

## Grid Search

Để bắt đầu, giá trị delta sẽ được gán bằng 2. Sau khi tìm được bộ ba ưng ý, delta sẽ được giảm xuống 0.25. Ở đây, interval được cho bằng 7.

Ví dụ: Giả sử đã tìm được các giá trị ở bước search thưa: C = 24, gamma = 2-2, epsilon = 2-6. Gán lại giá trị bắt đầu lân cận bộ ba này, sao cho từ vị trí đó đến giá trị hiện tại là 7/2 = 3 interval:

C = 23.25, gamma = 2-1.25, epsilon = 2-5.25.

## Pattern Search

* Đầu tiên, ta khởi tạo ngẫu nhiên một trung tâm của pattern nằm trong miền này.
* Khởi tạo bước tìm kiếm , ngưỡng
* Còn lại, cài đặt giống như lý thuyết ở trên.

# DỮ LIỆU

## Kịch bản

Để thấy được sự tương quan về kết quả đạt được, đầu tiên, ta sử dụng dữ liệu của [1]. Đây là giá đóng cửa của chỉ số NYSE Composite Stock Index lấy từ website của NYSE.COM từ 9-1-2004 đến 31-12-2007 bao gồm 1000 điểm dữ liệu. Trong [1], người ta chỉ khảo sát kết quả xu hướng sau 1 ngày(period = 1). Để tổng quát và thấy được sự toàn diện trong mô hình cài đặt, nhóm thực hiện trên 4 period: 1, 5, 10, 30.

Do mục đích đề tài là dự đoán cho thị trường Việt Nam nên ở đây, nhóm đã chọn ra một mã đại diện để thử nghiệm với mô hình sơ khai ban đầu. Mã đại diện này có các đặc tính sau:

* Thời gian dữ liệu quá khứ lớn gần bằng với thời gian thị trường tôn tại.
* Đã trải qua một thời gian đi lên mạnh và hiện đang giảm mạnh về giá trị.
* Các chỉ số kỹ thuật chưa dự đoán tốt được mã này, tỷ lệ trung bình dưới 40%.

Những đặc tính đó làm cho mã “đủ khó” để dự đoán và phần nào phản ánh sự biến động của thị trường chứng khoán non trẻ của Việt Nam. Thời gian từ 9-1-2004 đến 31-12-2008, bao gồm 1083 điểm dữ liệu.

Tất cả dữ liệu được chi theo tỷ lệ huấn luyện : kiểm thử là 9:1 và được huấn luyện với cross validation, số fold là 10.

## Đầu vào

Giá đóng cửa được scale về dạng return theo công thức

 với I là giá

Và khảo sát với số node là 5.

Đối với từng period khác nhau, giá trị từng node có sự điều chỉnh.

* Period = 1 và 5: mỗi node là return của 1 ngày trước đó.
* Period = 10: mỗi node là return của từng bộ 2 ngày trước.
* Period = 30: mỗi node là return của từng bộ 6 ngày trước.

Ví dụ với period = 10, thời điểm hiện tại là ngày 10, thời điểm cần dự đoán là ngày 20. Nhãn dự đoán sẽ là return 20 – 10 (của ngày 20 so với ngày 10), các node bên trong lần lượt là:

1: return 2 – 0, 2: return 4 – 2, 3: return 6 – 4, 4: return 8 – 6, 5 : return 10 – 8.

# THÍ NGHIỆM

## Phần SVR

* **Kiểm thử với mã chứng khoán NYSE Composite**

Với period = 1, thực hiện 3 lần kiểm thử với bộ ba tham số C, gamma, epsilon khác nhau. Bộ đầu tiên, lấy kết quả từ [1], bộ thứ hai được tìm bằng grid search, bộ thứ ba từ pattern search và thực hiện cross validation với số fold là 10.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | C | Gamma | Epsilon | NMSE | Sign(%) |
| Mô hình cài đặt | Bô [1] | 0.01 | 2.5 | 0.3 | 99.18 | 51 |
| Pattern search | 0.829 | 0.022 | 0.024 | 96.75 | 60 |
| Grid search | 5.01213 | 0.023 | 0.00314 | 96.74 | 62 |
| Mô hình [1] | Bộ [1] | 0.01 | 2.5 | 0.3 | 99.41 | 68.69 |
| 99.24 | 70.71 |

Figure Đồ thị kết quả mô hình cài đặt với grid search

Ở đây ta có nhận xét nhỏ, bộ tham số số của mô hình [1] áp dụng vào mô hình hiện tại không phù hợp. Kết quả sau khi dùng pattern search và grid search có cải thiện đáng kể tuy chưa tối ưu bằng kết quả của [1]. Điều này hiển nhiên vì [1] áp dụng mô hình SVR có cải tiến.

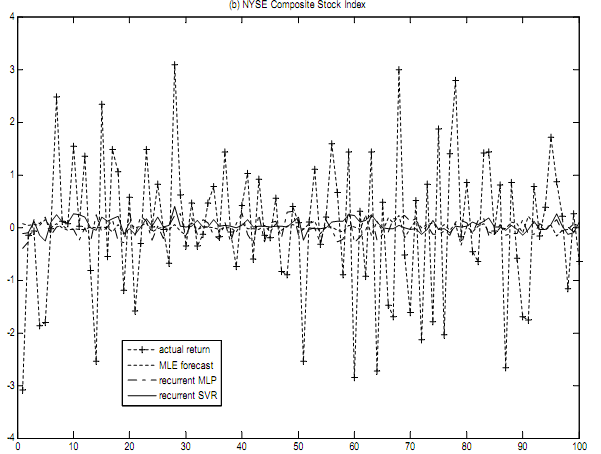


Figure Đồ thị kết quả của mô hình [1]

Với period = 5, 10 và 30, sau khi kiểm thử với grid search, ta được kết quả như sau:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Period = 5 | Period = 10 | Period = 30 |
| NMSE | 98.64 | 99.11 | 87.31 |
| Sign(%) | 58 | 57.6 | 51 |

Bằng pattern search, ta có được kết quả lần lượt như sau:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Period = 5 | Period = 10 | Period = 30 |
| NMSE | 103.9 | 107.2 | 82.32 |
| Sign(%) | 56 | 54 | 51 |

Kết quả giảm dần theo sự tăng lên của chu kỳ, điều này thể hiện sự thiếu thông tin trong vector đầu vào. Hơn nữa, nhìn chung, kết quả từ grid search tốt hơn hẳn pattern search.

Figure Kết quả với period = 5

Figure Kết quả với period = 10

Figure Kết quả với period = 30

Ta thấy, đường return thực đã có dạng của một xu hướng nhưng đường dự đoán không phản ánh được điều đó. Điều này phản ánh sự chưa phù hợp trong cách chọn dữ liệu cho từng node.

* **Kiểm thử với mã chứng khoán REE**

Sau đây là kết quả kiểm thử có được :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Period = 1 | Period = 5 | Period = 10 | Period = 30 |
| Grid search | NMSE | 102.3 | 121.3 | 110.8 | 234 |
| Sign(%) | 44 | 47 | 55 | 57 |
| Pattern search | NMSE | 103.3 | 102.5 | 103.5 | 185.2 |
| Sign(%) | 42 | 39 | 46 | 51 |

Figure Kết quả với period = 1

Figure Kết quả với period = 5

Figure Kết quả với period = 10

Figure Kết quả với period = 30

Từ kết quả trên ta rút ra vài kết luận sau:

* Grid search tỏ ra hiệu quả hơn pattern search.
* Nhược điểm của đầu vào như đã phản ánh ở mã NYSE cũng tồn tại ở đây.
* Điều đáng ghi nhận là công thức scale về dạng return. Ở NYSE, giá chạy trong khoảng 6000 - 10500, còn ở REE giá từ 20 – 300 nên công thức không còn phù hợp nữa.

## Phần ANN

Các tham số của mạng như sau:

* Số node tầng nhập: 5.
* Số node tầng ẩn: 3.
* Số node tầng xuất: 1.
* **Kiểm thử với mã chứng khoán NYSE Composite**

Period = 1:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Hệ số học = 0.01 | Hệ số học = 0.3 | Hệ số học = 0.5 |
| NMSE | 98 | 103 | 121 |
| Sign(%) | 53 | 53 | 54 |

Figure 6 Kết quả với period = 1, hệ số học = 0,3

Period = 5:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Hệ số học = 0.01 | Hệ số học = 0.3 | Hệ số học = 0.5 |
| NMSE | 95 | 108 | 116 |
| Sign(%) | 47 | 46 | 48 |

Figure 7 Kết quả với period = 5, hệ số học = 0,3

Period = 10:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Hệ số học = 0.01 | Hệ số học = 0.3 | Hệ số học = 0.5 |
| NMSE | 101 | 127 | 119 |
| Sign(%) | 52 | 38 | 41 |

Figure 8 Kết quả với period = 10, hệ số học = 0,3

Period = 30:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Hệ số học = 0.01 | Hệ số học = 0.3 | Hệ số học = 0.5 |
| NMSE | 71 | 77 | 82 |
| Sign(%) | 54 | 65 | 51 |

Figure 9 Kết quả với period = 30, hệ số học = 0,3

Với những thí nghiệm trên với từng period khác nhau và với hệ số học khác nhau ta nhận thấy:

* Ở period = 1, đường dự đoán biến động khá lớn, điều này phần nào thể hiện sự quá khớp của mô hình ANN.
* Với cùng một hệ số học (0.3) thì đường dự đoán và đường thực chưa “dính” vào nhau mặc dù ta thấy đường dự đoán có hình dạng tương tự với đường thực nhưng chúng có những chổ “xa” nhau.
* Vấn đề ở đây là chưa chọn được các bộ số thích hợp cho mạng ANN, cần cải tiến return đầu vào cho tầng nhập.
* **Kiểm thử với mã chứng khoán REE**
* Period = 1:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Hệ số học = 0.01 | Hệ số học = 0.3 | Hệ số học = 0.5 |
| NMSE | 100 | 103 | 127 |
| Sign(%) | 39 | 44 | 44 |

Figure 10 Kết quả với period = 1, hệ số học = 0,3

* Period = 5:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Hệ số học = 0.01 | Hệ số học = 0.3 | Hệ số học = 0.5 |
| NMSE | 108 | 126 | 124 |
| Sign(%) | 47 | 49 | 50 |

Figure 11 Kết quả với period = 5, hệ số học = 0,3

* Period = 10:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Hệ số học = 0.01 | Hệ số học = 0.3 | Hệ số học = 0.5 |
| NMSE | 106 | 145 | 139 |
| Sign(%) | 41 | 49 | 48 |

Figure 12 Kết quả với period = 10, hệ số học = 0,3

* Period = 30:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Hệ số học = 0.01 | Hệ số học = 0.3 | Hệ số học = 0.5 |
| NMSE | 127 | 1198 | 1610 |
| Sign(%) | 61 | 70 | 70 |

Figure 13 Kết quả với period = 30, hệ số học = 0,3

Với mã chứng khoán REE, ta nhận thấy có sự thay đổi rất lớn khi period lớn, nhìn vào biểu đồ, ta có thể nhận ra sự quá khớp của dữ liệu dự đoán. Tuy rằng ở period 30, tỷ lệ đạt 70% nhưng kết quả không cho thấy sự ổn định so với các period khác. Ngoài ra, một điểm dễ nhận thấy ở tình trạng quá khớp là độ lỗi NMSE quá lớn.

## ANN và SVR

Dưới đây là bảng tổng hợp kết quả so sánh được của 2 mô hình dựa trên độ chính xác.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Period | ANN(Sign) | SVR(Sign) |
| NYSE Composite | 1 | 54 | 62 |
| 5 | 48 | 58 |
| 10 | 52 | 57.6 |
| 30 | 65 | 51 |
| REE | 1 | 44 | 44 |
| 5 | 50 | 47 |
| 10 | 49 | 55 |
| 30 | 70 | 57 |

Nhìn chung, kết quả của SVR ổn định hơn ANN. SVR cho độ lỗi thấp hơn, và duy trì ổn định qua các period. ANN đối với số period lớn lại cho thấy sự đột biến về kết quả dự đoán. Tuy nhiên, kết quả này không phải do hiệu quả cài đặt mang lại, mà như phân tích ở trên, chúng bị quá khớp vì cho độ lỗi rất lớn.

Một điều dễ thấy là, đối với thị trường ổn định, SVR vượt trội so với ANN. Còn riêng với mã REE, được xem là khó dự đoán, ANN với sự quá khớp của mình “may mắn” có được kết quả tốt ở period lớn.

# KẾT LUẬN

Về phần SVR cho xu hướng, với sự so sánh tương quan kết quả trong [1] và kiểm thử với nhiều period khác nhau, ta rút ra được một vài kết luận sau đây:

* So sánh với [1], kết quả kiểm thử phản ánh việc cài đặt đúng hướng. Tuy nhiên, có thể do đặc thù của mô hình mà kết quả cao nhất đạt được có bộ tham số khác với [1].
* Grid search phản ánh sự hiệu quả của nó, khi tìm được bộ tham số giúp nâng tỷ lệ chính xác lên 10 điểm phần trăm. Tuy nhiên, với cách cài đặt grid search như trên, có thể mô hình đã bỏ qua điểm tối ưu hơn do việc chọn bước nhảy delta (2 cho search thưa và 0.25 cho search kỹ). Pattern search trong nhiều trường hợp không cho kết quả tốt bằng grid search.
* Đối với period bằng 1, cách chọn dữ liệu là 5 node phản ảnh return của 5 bộ 1 ngày trước không phản ánh được xu hướng, hiệu quả, dẫn đến kết quả (đồ thị) dự đoán không bám sát được với giá trị thực. Điều này ta cũng có thể dễ dàng nhận ra qua sự biến thiên quá đột ngột của return từng ngày.
* Đối với period từ 5 trở lên, dữ liệu đã có xu hướng hơn, nhưng do đầu vào chưa tạo được sự phản ánh phù hợp mà kết quả dự đoán chưa bám được vào đường xu hướng thật hay nói cách khác là chưa khớp (underfit).
* Cần có một công thức return phù hợp hơn với thị trường Việt Nam.

Về phần ANN, sự quá khớp làm cho kết quả dự đoán không ổn định, độ lỗi lớn.

# CÔNG VIỆC TƯƠNG LAI

Từ kết luận trên, ta rút ra được một vài điều cần làm sắp tới như sau:

* Định dạng dữ liệu đầu vào
* Công thức return phù hợp cho thị trường VN.
* Cân nhắc việc đưa vào một vài chỉ số kỹ thuật.
* Việc bố trí số node đầu vào, định dữ liệu trong từng node đó
* Phương pháp huấn luyện
* Theo kịch bản ban đầu, nhóm chưa thực hiện cách huấn luyện multi-period, sắp tới sẽ thử nghiệm phương pháp này.
* Tìm cách đặt trọng vào dữ liệu quan trọng hơn.
* Cân nhắc phương pháp chọn bộ tham số tối ưu
* Do pattern search không thể hiện được sự tối ưu cả về thời gian lẫn kết quả dự đoán, nên sắp tới vẫn tập trung vào grid search.
* Xem xét đưa GA vào chọn bộ tham số.
* Xem xét về kiến trúc
* Nghiên cứu việc đưa ra mối tương quan giữa các node đầu vào, đặt trọng vào node quan trọng hơn. (Hướng tới việc đưa fuzzy vào mô hình)
* Giải quyết tình trạng overfit của ANN và underfit của SVR.
* Online
* Song song với việc cải tiến mô hình hiện tại, nhóm sẽ nghiên cứu một mô hình có khả năng thực hiện dự đoán online, không hẳn là chỉ thay đổi cách huấn luyện.

# TÀI LIỆU THAM KHẢO

[1] ShiyiChen, Kiho Jeong,Wolfgang K.Hardle,*Recurrent Support Vector Regression for a Nonlinear ARMA Model with Applications to Forecasting Financial Returns*, SFB 649 Discussion Paper, 2008