**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN KHOA HỌC MÁY TÍNH**

- - - - 🙞🙞🟔🙜🙜 - - - -

**NHÓM LOTUS**

**0712228 – TRẦN TRUNG KIÊN**

**0712263 – VẠN DUY THANH LONG**

**PHÁT HIỆN KHUÔN MẶT VỚI THUẬT TOÁN ADABOOST**

**ĐỒ ÁN MÔN HỌC KỸ THUẬT TRÍ TUỆ NHÂN TẠO**

**GIẢNG VIÊN LÝ THUYẾT**

**TS. LÊ HOÀNG THÁI**

**NĂM 2010**

# MỤC LỤC

MỤC LỤC 2

Chương 1 TÓM TẮT 4

Chương 2 GIỚI THIỆU 4

2.1 TẦM QUAN TRỌNG CỦA BÀI TOÁN DỰ ĐOÁN CHỨNG KHOÁN 4

2.2 PHÁT BIỂU VẤN ĐỀ 4

2.3 GIẢI QUYẾT VẤN ĐỀ 4

2.3.1 Phân tích cơ bản 5

2.3.2 Phân tích kỹ thuật 5

Chương 3 CÁC VẤN ĐỀ CỦA HƯỚNG TIẾP CẬN MÁY HỌC 5

3.1 CHỌN INPUT/ OUTPUT 6

3.2 TIỀN XỬ LÝ 6

3.3 CHỌN MÔ HÌNH MÁY HỌC 6

3.4 ĐÁNH GIÁ MÔ HÌNH 7

Chương 4 NỀN TẢNG LÝ THUYẾT 9

4.1 ARTIFICIAL NEURAL NETWORK (ANN) 9

4.2 SUPPORT VECTOR REGRESSION (SVR) 9

4.2.1 Tổng quan 9

4.2.2 SVR 9

4.2.3 Chọn các tham số cho mô hình 14

4.2.3.1 Grid Search 14

4.2.3.2 Pattern Search 14

4.2.3.3 Cross-Validation truyền thống 14

Chương 5 CÁC CẢI TIẾN 15

5.1.1 CẢI TIẾN CROSS-VALIDATION TRONG TÌM THAM SỐ CỦA SVR 15

5.1.2 TĂNG ĐỘ CHÍNH XÁC VỀ HƯỚNG TRONG DỰ ĐOÁN GIÁ 16

Chương 6 CÀI ĐẶT 17

Chương 7 THÍ NGHIỆM 17

7.1 DỰ ĐOÁN GIÁ 17

7.2 DỰ ĐOÁN XU HƯỚNG 17

Chương 8 KẾT LUẬN VÀ CÔNG VIỆC TƯƠNG LAI 17

Chương 9 HƯỚNG DẪN SỬ DỤNG 17

TÀI LIỆU THAM KHẢO 17

# TÓM TẮT

# GIỚI THIỆU

## TẦM QUAN TRỌNG CỦA BÀI TOÁN DỰ ĐOÁN CHỨNG KHOÁN

## PHÁT BIỂU VẤN ĐỀ

Ở đây, nhóm chia vấn đề dự đoán chứng khoán thành 2 vấn đề con:

* *Dự đoán giá:* mục tiêu của ta là dự đoán giá của một số ngày tiếp theo (**ngắn hạn**.) Cái ta hướng đến là dự đoán **vừa đúng về giá trị vừa đúng về xu hướng**(tăng bao nhiêu, giảm bao nhiêu.) Output là giá dự đoán.
* *Dự đoán xu hướng:* mục tiêu của ta là dự đoán xu hướng xét trong 1 khoảng thời gian nào đó, mà thường là **trung hạn và dài hạn**. Cái ta hướng tới là dự đoán **đúng xu hướng** (tăng hay giảm, không quan tâm là tăng bao nhiêu hay giảm bao nhiêu.) Output là một trong hai giá trị ứng với 2 trường hợp: xu hướng tăng, xu hướng giảm (Ta cũng có thể đưa thêm trường hợp thứ 3: xu hướng không đổi.)

## GIẢI QUYẾT VẤN ĐỀ

Một cách chung nhất, có hai hướng tiếp cận đối với bài toán dự đoán chứng khoán:

* Phân tích cơ bản.
* Phân tích kỹ thuật.

### Phân tích cơ bản

* Phân tích bản cân đối tài khoản và bản báo cáo lợi tức của công ty để tìm ra giá trị nội tại (tình hình phát triển của công ty) đó.
* Phân tích cơ bản cho rằng trong ngắn hạn, thị trường có thể đánh giá sai về giá trị nội tại của cổ phiếu của công ty đó nhưng về dài hạn, giá cổ phiếu sẽ hội tụ về giá trị nội tại.
* Như vậy, ta có thể thu lợi nhuận bằng cách mua cổ phiểu khi bị thị trường đánh giá thấp giá trị nội tại và bán ra khi thị trường đánh giá đúng giá trị nội tại.

### Phân tích kỹ thuật

* Phân tích kỹ thuật cho rằng tất cả các thông tin về công ty đều được phản ánh qua giá cả 🡪Phân tích cơ bản chỉ mất công.
* Cách thức: kết hợp dữ liệu quá khứ với các mô hình sóng.

Gần đây, cộng đồng phân tích kỹ thuật có xu hướng sử dụng các mô hình máy học như là ANN, SVM, Gaussian Process, Hidden Markov Model, các mô hình lai, …

Ở đây, nhóm giải quyết vấn đề theo hướng phân tích kỹ thuật mà cụ thể là sử dụng 2 mô hình máy học được cho là mạnh nhất hiện nay trong lĩnh vực dự đoán chứng khoán: ANN (Artificial Neural Network) và SVR (Support Vector Regression.)

# CÁC VẤN ĐỀ CỦA HƯỚNG TIẾP CẬN MÁY HỌC

Để giải được bài toán khó này, trước hết ta cần phải nhận ra được các vấn đề cần phải giải quyết của nó. Với hướng tiếp cận sử dụng các mô hình máy học, ta có 4 vấn đề cơ bản như sau:

## CHỌN INPUT/ OUTPUT

Đây là bước rất quan trọng. Ta phải chọn input/ output sao cho nó cung cấp đủ thông tin để mô hình của ta có thể nhận ra được các mẫu tiềm ẩn. Nhưng nếu ta chọn quá nhiều input thì cũng có thể gây ra nhiễu, làm giảm khả năng dự đoán của mô hình.

Nhìn chung, các input có thể sử dụng gồm có: Giá mở cửa, giá cao nhất, giá thấp nhất, giá đóng cửa, các indicator.

Ở đây, để khởi đầu, nhóm chọn input chỉ là giá đóng cửa.

## TIỀN XỬ LÝ

Nhìn chung, bước tiền xử lý thực hiện hai nhiệm vụ: khử nhiễu và chuẩn hóa.

Ở đây, với input chỉ là giá đóng cửa, ở bước tiền xử lý nhóm chỉ thực hiện nhiệm vụ chuẩn hóa.

* Dự đoán giá: đơn giản là scale về [0, 1] theo công thức:
* Dự đoán xu hướng:

## CHỌN MÔ HÌNH MÁY HỌC

Nhìn chung, các mô hình máy học được xây dựng lên dựa trên các giả sử nào đó về sự phân bố của dữ liệu. Một mô hình có thể phù hợp ở ứng dụng này nhưng khi đưa qua ứng dụng khác thì chưa chắc. Vì vậy, cách duy nhất để chọn được mô hình máy họcphù hợp là thử với các mô hình máy học khác nhau.

Ở đây, nhóm khởi đầu với hai mô hình được coi là mạnh nhất hiện nay trong lĩnh vực dự đoán chứng khoán: ANN và SVR.

Khi đã chọn được mô hình máy học rồi thì lại nảy sinh một vấn đề con là **chọn các tham số cho mô hình**.

## ĐÁNH GIÁ MÔ HÌNH

Ở đây, ta có thể chia các độ đo để đánh giá mô hình thành 2 nhóm:

* Nhóm các độ đo về giá trị: càng nhỏ càng tốt.
  + MSE (Mean Square Error):

Với y’n và yn lần lượt là giá trị dự đoán và giá trị thực.

* + MAE (Mean Absolute Error):
  + MAPE (Mean Absolute Percentage Error):
  + NMSE (Normalized Mean Square Error):

Với là giá trị trung bình của y1, …, yn

* Nhóm các độ đo về hướng:
  + DS (Directional Symmetry): càng lớn càng tốt.

Với:

* + Sign:
  + DM4Price (Direction Measure For Price): đây là độ đo do nhóm tự đặt ra cho bài toán dự đoán giá. Nó xuất phát từ hai lý do:
    - Như đã nói, mục tiêu của ta trong bài toán dự đoán giá là dự đoán đúng cả về giá trị lẫn xu hướng. Giá trị thì ta đã có rất nhiều độ đo rồi, nhưng xu hướng thì chỉ có độ đo DS. Tuy nhiên, việc xét dấu của tích không cho ta nhiều thông tin lắm. Ở đây, ta cần phải xét dấu của tích
    - Các cổ phiếu Việt Nam có đặc điểm là giá thường không đổi, dẫn đến tích trên = 0. Nếu ta cho khi tích >= 0 như ở độ đo DS thì với các cổ phiếu Việt Nam sẽ cho DS rất cao, trong khi điều đó không nói lên rằng mô hình của ta là tốt.

Từ đây, ta có công thức DM4Price như sau:

Xét tích

* + - Nếu tích > 0: dự đoán đúng.
    - Nếu tích < 0: dự đoán sai.
    - Nếu tích = 0: không đánh giá được.

Độ đo DM4Price sẽ cho ra 3 kết quả lần lượt ứng với phần trăm số phần tử dự đoán đúng, dự đoán sai và không đánh giá được.

# NỀN TẢNG LÝ THUYẾT

## ARTIFICIAL NEURAL NETWORK (ANN)

### Tổng quan

Ý tưởng của ANN là mô phỏng một mạng neural nhân tạo có thể hoạt động giống với mạng neural sinh học bằng cách kết hợp tuyến tính các hàm truyền phi tuyến lại với nhau để xây dựng mô hình hồi quy.

Mục đích của ANN là tối thiểu hóa hàm độ lỗi thông qua việc xác định một bộ tham số phù hợp cho các hàm truyền cơ bản.

### Mô hình ANN

Mô hình ANN có thể xem nhưlà một hàm phi tuyến nhận vào một tập các giá trị đầu vào và trả ra tập các giá trị thông qua sự chi phối của bộ trọng số w.

Để đơn giản hóa công thức, ta giả sử có thêm biến đầu vào , chúng ta sẽ có công thức tương đương như sau:

Hướng tiếp cận của vấn đề là ta phải tối thiểu hóa hàm độ lỗi sum-of-squares thông qua việc xác định bộ tham số cho mạng. Cho một bộ huấn luyện với tập các giá trị đầu vào và nhãn của nó là , ta có hàm độ lỗi là:

Ta thấy rằng do là một hàm phi tuyến nên không thể là bao lồi (nonconvex) do đó một khuyết điểm của ANN là dễ bị bẫy vào các giá trị tối ưu cục bộ.

Vì hàm lỗi là một hàm liên tục theo biến w, nên các giá trị cực tiểu của nó sẽ xuất ở điểm mà có đạo hàm bằng 0: .

Tuy nhiên, như đã nói hàm độ lỗi phụ thuộc phi tuyến vào bộ trọng số w, nên sẽ có rất nhiều điểm trong không gian trọng số có giá trị đạo hàm bằng 0 (các cực tiểu cục bộ). Việc tối ưu hóa phải hướng đến tìm giá trị nhỏ nhất của hàm độ lỗi, hay nói cách khác chính là phải tìm được cực tiểu toàn cục.

Không có hy vọng trong cách giải quyết vấn đề tìm cực tiểu toàn cục với . Một vài thuật toán khác tập trung vào việc lựa chọn giá trị khởi tạo và việc cập nhật vector trọng số

Trong đó thuật toán Gradient descent được áp dụng phổ biến trong việc huấn luyện mô hình ANN với bộ dữ liệu lớn. Ta có hàm độ lỗi cho một tập các biến quan sát được độc lập nhau là:

Ý tưởng của thuật toán Gradient descent là từ việc khởi tạo một giá trị ngẫu nhiên cho bộ trọng số w, qua việc chạy thuật toán nhiều lần ta sẽ tiến hành cập nhật dần bộ trọng số cho đến khi hội tụ, các lần chạy sau cho kết quả tối ưu hơn các kết quả trước đó.

Ta sẽ thấy được rằng hệ số nhỏ sẽ làm thuật toán lâu hội tụ, và nếu nó quá lớn thì hàm lỗi bị xoay quanh các giá trị biên của điểm hội tụ.

Tóm lại mô hình ANN sẽ qua các bước xử lý sau:

* Cho vector input I = (i1, i2, …, in) được truyền vào lớp input của mạng. Sau quá trình tiền xử lý (khử nhiễu, chuẩn hóa), mỗi node ở lớp input sẽ tạo ra vector O(i) = (o1, o2, …, on) truyền đến lớp ẩn.
* Giá trị đầu vào của neuron thứ j của lớp ẩn (h) sẽ được tính như sau:
* Giá trị output của neuron thứ j trong lớp ẩn là được tính theo công thức:

Ở đây net chính là được gọi là hàm truyền phi tuyến.

* Giá trị đầu vào của neuron thứ k trong lớp output (o) được tính theo công thức:
* Giá trị đầu ra của neuron thứ k trong lớp output là được tính theo công thức:

Ở đây net chính là và là hàm truyền phi tuyến cho lớp output.

* Tiếp theo khi đã có kết quả ở lớp output, ta cần tính độ lỗi giữa giá trị dự đoán với giá trị thực để lan truyền ngược lại cập nhật trọng số theo công thức:

Ở đây là giá trị thực và là đạo hàm của hàm truyền phi tuyến ở lớp xuất.

* Ta tính tiếp độ lỗi ở các neuron lớp ẩn theo công thức:

Ở đây là đạo hàm của hàm truyền phi tuyến ở lớp ẩn.

* Độ thay đổi trọng số ở lớp output và ở lớp ẩn ở vòng lặp t được tính như sau:

Ở đây là hệ số học và là nhân tố hướng để hạn chế bị rơi vào bẫy tối ưu cục bộ.

* Trọng số mới được cập nhật theo công thức sau:

## SUPPORT VECTOR REGRESSION (SVR)

### Tổng quan

Ý tưởng cơ bản của SVR là ta sẽ ánh xạ không gian đầu vào (mà nếu ta áp dụng hồi qui tuyến tính thì không hiệu quả) sang một không gian mới cao chiều hơn mà ở đó, ta có thể áp dụng được hồi qui tuyến tính.

Đặc điểm của SVR là cho ta một giải pháp thưa; nghĩa là để xây dựng được hàm hồi qui, ta không cần phải sử dụng hết tất cả các điểm dữ liệu trong bộ huấn luyện. Những điểm có đóng góp vào việc xây dựng hàm hồi qui được gọi là những Support Vector.

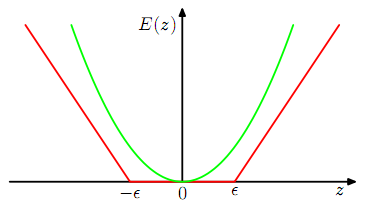
Điểm mạnh của SVR là sử dụng tối ưu hóa rủi ro cấu trúc (structural risk minimization), nhờ đó mà khả năng tổng quát hóa cao, tránh overfit (ANN thì dễ bị overfit.) Hơn nữa, hàm mục tiêu của SVR là hàm “convex”, do đó điểm cực trị tìm được sẽ là cực trị toàn cục (hàm mục tiêu của ANN thì không như vậy, nó có nhiều điểm cực trị và dẫn dến cực trị tìm được thường là cực trị cục bộ.)

### SVR

Với bài toán hồi qui tuyến tính đơn giản, ta phải minimize hàm lỗi chuẩn hóa:

Để có được một giải pháp thưa, ta sẽ thay hàm lỗi trên bằng hàm lỗi -insensitive. Đặc điểm của hàm lỗi này là nếu trị tuyệt đối của sự sai khác giữa giá trị dự đoán y(x) và giá trị đích nhỏ hơn epsilon (với epsilon > 0) thì nó coi như là độ lỗi bằng 0.

Để hiểu thêm, ta hãy ngó qua hình vẽ dưới đây:

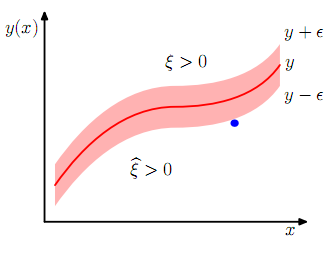


Trong đó, đường màu xanh là hàm lỗi bậc hai thông thường; đường màu đỏ là hàm lỗi -insensitive.

Như vậy bây giờ, ta phải minimize hàm lỗi chuẩn hóa sau:

Với:

Để cho phép một số điểm nằm ngoài ống epsilon, ta sẽ đưa thêm các biến “slack” vào. Đối với mỗi điểm dữ liệu xn, ta cần hai biến slack và ; trong đó ứng với điểm mà (nằm ngoài và phía trên ống) và ứng với điểm mà (nằm ngoài và phía dưới ống.)



Điều kiện để một điểm đích nằm trong ống là: với yn = y(xn). Với việc sử dụng các biến slack, ta cho phép các các điểm đích nằm ngoài ống (ứng với các biến slack > 0) và như thế thì điều kiện của ta bây giờ sẽ là:

[4.2.2.1]

[4.2.2.2]

Như vậy, ta có hàm lỗi cho SVR:

[4.2.2.3]

Mục tiêu của ta là minimize hàm lỗi này với các ràng buộc:

* [4.2.2.1]
* [4.2.2.2]

Ta hãy gọi đây là vấn đề tối ưu hóa A.

Có ngay hàm Lagrange:

[4.2.2.4]

Với

Lấy đạo hàm theo w, b, , và cho bằng 0, ta được:

[4.2.2.5]

[4.2.2.6]

[4.2.2.7]

[4.2.2.8]

Dùng 4 kết quả này thế vào hàm Lagrange, ta sẽ loại bỏ được w, b, , , , :

[4.2.2.9]

Với k là hàm kernel:

Như vậy, ta đã chuyển từ vấn đề tối ưu hóa A sang vấn đề tối ưu hóa B:

Maximize [4.2.2.9] với các ràng buộc:

* [4.2.2.6]

Lợi ích chính của việc chuyển đổi từ vấn đề tối ưu hóa A sang vấn đề tối ưu hóa B là vấn đề tối ưu hóa B có sử dụng hàm kernel. Điều này sẽ giúp cho việc tính toán trong không gian cao chiều trở nên rất hiệu quả.

Thế [4.2.2.5] vào hàm hồi qui ban đầu, ta có sự dự đoán cho một mẫu mới x:

[4.2.2.10]

Theo điều kiện KKT, có ngay:

[4.2.2.11]

[4.2.2.12]

[4.2.2.13]

[4.2.2.14]

Từ đây, ta có thể rút được những thông tin quan trọng như sau:

* Nếu thì : điểm nằm ở biên trên của ống () hoặc nằm ngoài về phía trên của ống ()
* Nếu thì : điểm nằm ở biên dưới của ống () hoặc nằm ngoài về phía dưới của ống ()
* và không thể cùng dương vì nếu vậy thì ta có: và , cộng lại ta sẽ thấy ngay vế trái luôn dương, trong khi vế phải bằng 0: vô lý!
* Những điểm Support Vector là những điểm đóng góp vào hàm dự đoán [4.2.2.10], nghĩa là những điểm có hoặc : những điểm nằm trên biên ống hoặc nằm ngoài ống.
* Những điểm nằm trong ống sẽ có và do đó không đóng góp gì vào quá trình dự đoán.

*Tính b:*

Thấy ngay ta dễ tính được b bằng cách xét một điểm xncó . Từ [4.2.2.13] ta có . Từ [4.2.2.11] ta có . Kết hợp với [4.2.2.10] có ngay:

Ta cũng sẽ được kết quả tương tự nếu xét điểm có .

Để vững chắc hơn, ta nên lấy trung bình của tất cả các giá trị của b lại.

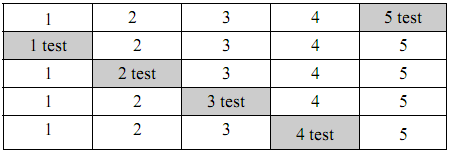
### Chọn các tham số cho mô hình

#### Grid Search

#### Pattern Search

#### Cross-Validation truyền thống

Để chọn các tham số cho mô hình thì Grid Search và Pattern Search sẽ được kết hợp với Cross-Validation.



Với Cross-Validation truyền thống thì ta sẽ chia bộ train thành k phần bằng nhau (k fold), chẳng hạn ở đây là 5 phần.

* Phần 5 sẽ được test dựa trên train phần 1 – 4.
* Phần 1 sẽ được test dựa trên train phần 2 – 5.
* Phần 2 sẽ được test dựa trên train phần 1, 3, 4, 5.
* Phần 3 sẽ được test dựa trên train phần 1, 2, 4, 5.
* Phần 4 sẽ được test dựa trên train phần 1, 2, 3, 5.

Độ lỗi ứng với 5 lần test sẽ được lấy trung bình lại. Bộ tham số nào cho độ lỗi trung bình này nhỏ nhất sẽ được chọn.

# CÁC CẢI TIẾN

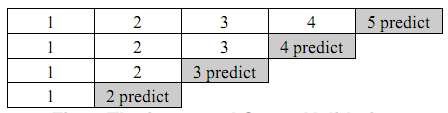
### CẢI TIẾN CROSS-VALIDATION TRONG TÌM THAM SỐ CỦA SVR

Ở đây, nhóm sử dụng hai phương pháp tìm tham số cho SVR là Grid Search và Pattern Search. Nhóm nhận thấy Grid Search với bản chất là vét cạn cho kết quả tốt hơn và ổn định hơn; tuy nhiên, nhược điểm của Grid Search là thời gian chạy quá lâu, đặc biệt là khi bộ huấn luyện lớn.

Điểm nữa là hiện nay trong hàm Cross-Validation của LIBSVM sử dụng phương pháp shuffle (xáo trộn các phần tử trong bộ huấn luyện rồi mới chia fold.) Phương pháp này có 2 nhược điểm:

* Tính không ổn định: mỗi lần ra một kết quả khác nhau, thời gian chạy khi nhanh khi lâu.
* Theo [2] thì phương pháp này không phù hợp đối với dữ liệu time series. Dữ liệu time series có đặc trưng là tính có hướng; nghĩa là ta chỉ có thể dùng dữ liệu được tạo ra sớm hơn để dự đoán dữ liệu được tạo ra trễ hơn mà không thể làm theo chiều ngược lại. Như vậy thì rõ ràng phương pháp shuffle sẽ làm mất tính chất này của dữ liệu time series.

Từ đây, [2] đề xuất phương pháp Cross-Validation cải tiến dành cho dữ liệu time series như sau:



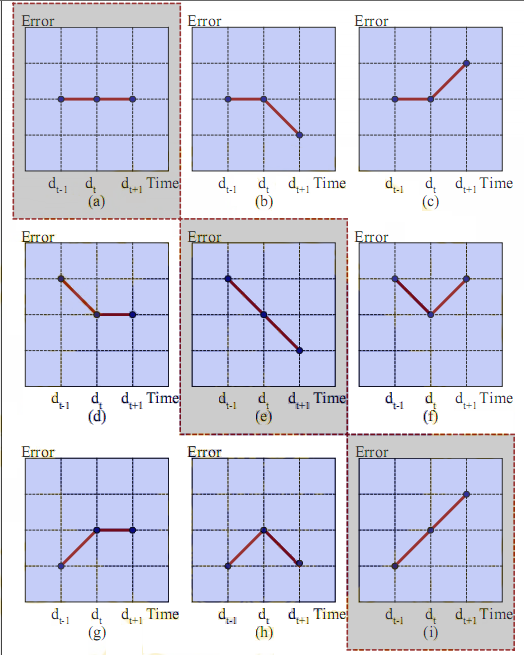
Ta chia bộ huấn luyện làm 5 phần bằng nhau mà không xáo trộn bộ huấn luyện.

* Phần 5 sẽ được test dựa trên train phần 1-4.
* Phần 4 sẽ được test dựa trên train phần 1-3.
* Phần 3 sẽ được test dựa trên train phần 1-2.
* Phần 2 sẽ được test dựa trên train phần 1.

Nhận thấy với cách chia này thì ngoài việc đảm bảo tính có hướng của dữ liệu time series thì nó còn có 1 tác dụng nữa là làm giảm thời gian chạy cho Grid Search (Vì kích thước bộ train giảm dần thay vì luôn cố định như trong Cross-Validation truyền thống.)

### TĂNG ĐỘ CHÍNH XÁC VỀ HƯỚNG TRONG DỰ ĐOÁN GIÁ

Trong quá trình thí nghiệm, nhóm nhận thấy những trường hợp cho độ lỗi về giá trị nhỏ hơn chưa chắc đã cho sự đúng về xu hướng cao hơn.

Như đã nói mục tiêu của bài toán dự đoán giá là dự đoán đúng cả về giá trị lẫn xu hướng. Nhưng hiện nay quá trình huấn luyện chỉ tập trung vào việc minimize độ lỗi về giá trị. Như vậy, để đạt được mục tiêu đã nói ta cần đưa thêm thông tin về xu hướng vào trong quá trình huấn luyện.

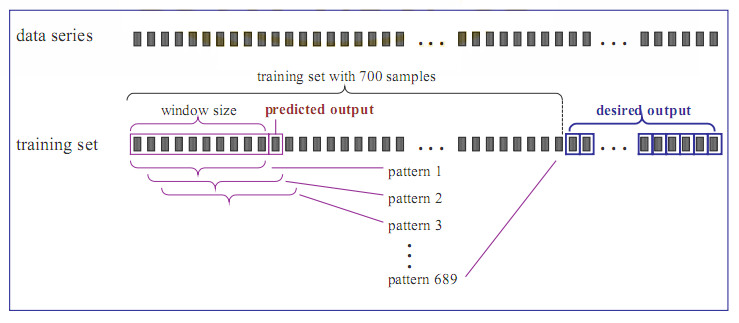
Ở đây dt chính là giá trị thực của ngày dự đoán, dt+1 là giá trị của ngày tiếp theo.Chúng ta chỉ quan tâm đến trường hợp (a), (e), (i) vì nó biểu lộ xu hướng rõ ràng tăng, giảm hoặc giữ nguyên.

Như vậy giá trị tn sẽ được tính lại theo công thức sau:

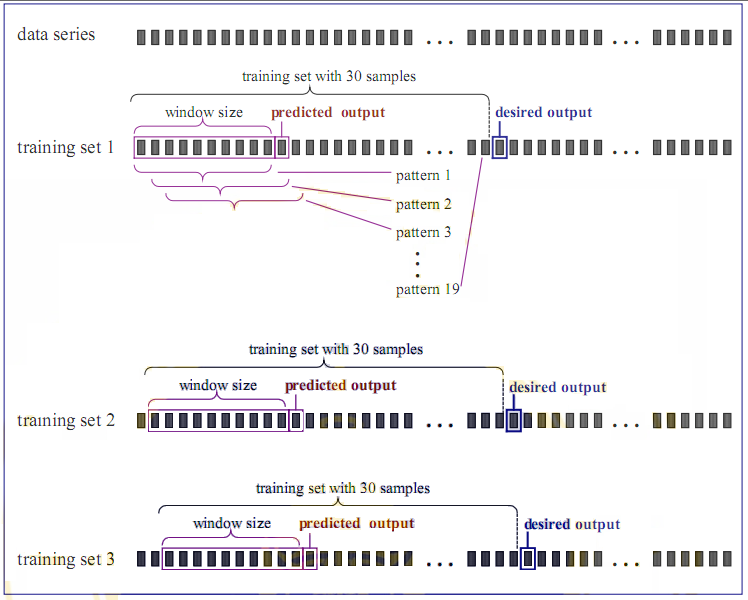
### Cải tiến cách tổ chức dữ liệu đưa vào mô hình:

Hiện tại có 2 cách tổ chức dữ liệu đưa vào mô hình như sau:

* Cách truyền thống: phân dữ liệu có được ra làm 2 bộ dữ liệu con là training và test theo một tỉ lệ cho trước, yêu cầu bộ dữ liệu cho training nên có kích thước lớn.



* Cách cải tiến: Mỗi lần dự đoán một giá trị mới ta phải huấn luyện lại toàn bộ mô hình (một lần huấn luyện chỉ để dự đoán duy nhất 1 giá trị liền sau nó, để dự đoán giá trị tiếp theo đó ta phải xây dựng lại mô hình), nhưng bộ dữ liệu huấn bây giờ có kích thước nhỏ hơn rất nhiều so với cách truyền thống



Như đã nói, một trong những nhược điểm lớn của mô hình hiện tại là cần một lượng lớn dữ liệu mẫu cho quá trình huấn luyện, đó là một trong những nguyên nhân làm quá trình huấn luyện chậm đi, dẫn đến không thể xử lý trực tuyến khi có thông tin dữ liệu mớido quá trình huấn luyện mất nhiều thời gian, cho kết quả không tốt cho các mã chứng khoán mới ra vì bộ dữ liệu thu được ít.

Một lý do khác chúng ta tiến hành xử lý dữ liệu chuỗi thời gian nên sẽ có một số mẫu sau một thời gian nó không còn mang tính hợp lệ (như giai đoạn đầu mới mở giá có thể là 20$, nhưng sau môt thời giản hoạt động công ty làm ăn phát triển giá trị cổ phiếu khó có thể trở về 20$ như ban đầu). Đó cũng là nhược điểm ở cách truyền thống

# CÀI ĐẶT

# THÍ NGHIỆM

## DỰ ĐOÁN GIÁ

## DỰ ĐOÁN XU HƯỚNG

# KẾT LUẬN VÀ CÔNG VIỆC TƯƠNG LAI

# HƯỚNG DẪN SỬ DỤNG

# TÀI LIỆU THAM KHẢO

[1]Christopher M. Biship, *Pattern Recognition and Machine Learning,* Springer (2007)

[2]Shukuan Lin; Shaomin Zhang; Jianzhong Qiao; Hualei Liu; Ge Yu,[*A Parameter Choosing Method of SVR for Time Series Prediction*, IEEE, 2008](http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=4708961&queryText%3Dtime+series+prediction%26searchWithin%3Dsvr%26openedRefinements%3D*%26ranges%3D2006_2011_Publication_Year%26searchField%3DSearch+All)

[3]