ĐẠI HỌC CÔNG NGHỆ - ĐHQGHN
KHOA CÔNG NGHỆ THÔNG TIN

INT3234E 53 - Phân tích dữ liệu dự báo
Báo cáo tiểu luận cuối kì

PHÂN TÍCH DỰ ĐOÁN GIÁ BITCOIN SỬ DỤNG HỌC MÁY

Bài báo gốc: Analysis of Bitcoin Price Prediction Using Machine Learning

Hoàng Bảo An - 22024545
Lê Tuấn Kiệt - 22024546

Dưới sự hướng dẫn của:
TS. Nguyễn Thị Hậu

Hà Nội, 2024

Phân chia công việc

Phần nghiên cứu

Thành viên	**Công việc thực hiện**
Lê Tuấn Kiệt	Phân tích và nghiên cứu bài toán; Nghiên cứu phương pháp mô hình RNN và LSTM; Đánh giá, nghiên cứu và mở rộng so sánh kết quả với các bài báo khác cùng chủ đề.
Hoàng Bảo An	Phân tích và nghiên cứu bài toán; Nghiên cứu phương pháp mô hình Cây quyết định và Random Forest; Cài đặt và thực hiện code thí nghiệm Random Forest & LSTM.

Phần làm báo cáo

Thành viên	**Công việc thực hiện**
Lê Tuấn Kiệt	Viết bản báo cáo Chương 1, Chương 2, Chương 3, Chương 6. Làm slide thuyết trình.
Hoàng Bảo An	Viết bản báo cáo phần 3.2, Chương 4, 5. Làm slide thuyết trình.

Tóm tắt

[1]Bài toán phân tích và dự đoán giá Bitcoin trong bối cảnh thị trường tiền mã hóa đầy biến động đã trở thành một thách thức quan trọng đối với các nhà đầu tư và các công ty tài chính. Bài báo cáo tiểu luận của chúng tôi sẽ trình bày, phân tích, mở rộng và đánh giá một bài báo nghiên cứu tiềm năng tập trung vào phương pháp sử dụng ứng dụng học máy. Mục tiêu của nghiên cứu này là phát triển một mô hình thuật toán với độ chính xác cao trong việc dự đoán giá Bitcoin vào ngày tiếp theo, thông qua việc áp dụng các phương pháp hồi quy rừng ngẫu nhiên (Random Forest) và LSTM, đồng thời phân tích những yếu tố ảnh hưởng đến giá của Bitcoin.

Nghiên cứu tập trung vào việc so sánh hai mô hình học máy là LSTM và Random Forest để kiểm chứng khả năng dự đoán giá Bitcoin, nhằm trả lời câu hỏi liệu các mô hình này có thể cung cấp dự báo chính xác trong bối cảnh biến động của thị trường tiền mã hóa hay không. Các nghiên cứu trước đây đã chỉ ra rằng Bitcoin là một tài sản có độ biến động cao, khiến các phương pháp dự đoán truyền thống gặp nhiều thách thức trong việc nhận diện xu hướng giá. Việc áp dụng các kỹ thuật học máy được coi là cách tiếp cận hiện đại, nhằm cải thiện khả năng dự đoán và cung cấp góc nhìn sâu sắc hơn về các yếu tố ảnh hưởng đến giá Bitcoin.

Trong nghiên cứu này, dữ liệu được thu thập từ nhiều nguồn khác nhau, bao gồm giá Bitcoin, các chỉ số thị trường tài chính, giá hàng hóa, và mức độ quan tâm của công chúng, từ năm 2015 đến năm 2022. Phương pháp nghiên cứu bao gồm việc áp dụng mô hình LSTM và Random Forest, hai công cụ phổ biến trong phân tích chuỗi thời gian và dự báo.

Kết quả nghiên cứu chỉ ra rằng cả hai mô hình đều có khả năng dự đoán giá Bitcoin tương đối tốt, tuy nhiên, mô hình Random Forest cho thấy hiệu suất dự đoán vượt trội hơn so với LSTM. Điều này khẳng định tính hiệu quả của Random Forest trong việc xử lý dữ liệu phức tạp và đa chiều từ thị trường tiền mã hóa.

Mục lục

Danh sách hình vẽ

Danh sách bảng

Mở đầu

Trong những năm gần đây, xu hướng chuyển đổi số và sự bùng nổ của kỷ nguyên dữ liệu lớn đã mở ra nhiều cơ hội cho các công ty và tổ chức trong việc xử lý, giải quyết các bài toán kinh doanh, đặc biệt là trong lĩnh vực tài chính và đầu tư. Thị trường tiền điện tử cũng không nằm ngoài xu thế này khi ngày càng thu hút sự quan tâm mạnh mẽ của các nhà đầu tư, nhà nghiên cứu và các cơ quan quản lý. Bitcoin, đồng tiền điện tử đầu tiên được giới thiệu bởi Satoshi Nakamoto vào năm 2008, đã trở thành biểu tượng của cuộc cách mạng công nghệ tài chính nhờ vào khả năng hoạt động phi tập trung, bảo mật cao và khả năng lưu trữ giá trị. Tuy nhiên, tính biến động mạnh mẽ của Bitcoin đã đặt ra một thách thức lớn trong việc dự đoán giá trị của nó, đồng thời tạo nên nhu cầu nghiên cứu nhằm giảm thiểu rủi ro và tối ưu hóa chiến lược đầu tư cho các bên liên quan.

Việc dự đoán giá Bitcoin trở nên quan trọng khi giá trị của đồng tiền này liên tục trải qua những giai đoạn tăng giảm đột ngột, đặc biệt trong các giai đoạn bùng nổ giá vào năm 2017 và 2021. Tính biến động cao của Bitcoin được thể hiện rõ qua độ lệch chuẩn của tỷ suất lợi nhuận hàng ngày lên tới 3,85% trong khoảng thời gian từ năm 2015 đến 2022, cao hơn nhiều so với vàng hay chỉ số S&P500. Điều này đặt ra câu hỏi làm thế nào để dự đoán giá Bitcoin một cách hiệu quả nhằm giúp các nhà đầu tư giảm thiểu rủi ro và tận dụng cơ hội. Trong bối cảnh đó, các phương pháp học máy đã nổi lên như một hướng tiếp cận tiềm năng cho bài toán dự đoán giá Bitcoin.

Bài báo "Analysis of Bitcoin Price Prediction Using Machine Learning"của Junwei Chen (2023) đã nghiên cứu việc áp dụng các mô hình học máy - hồi quy rừng ngẫu nhiên (Random Forest Regression) và mạng thần kinh LSTM (Long Short-Term Memory) - để dự đoán giá Bitcoin vào ngày tiếp theo. Hai mô hình này được lựa chọn nhờ vào khả năng phân tích dữ liệu thời gian, trong đó LSTM là mô hình phổ biến trong việc xử lý chuỗi thời gian với tính phụ thuộc, còn hồi quy rừng ngẫu nhiên được đánh giá cao bởi khả năng giải thích các yếu tố ảnh hưởng. Nghiên cứu không chỉ tìm hiểu mô hình nào có độ chính xác cao hơn mà còn đánh giá những yếu tố ảnh hưởng chính đến biến động giá Bitcoin trong các giai đoạn khác nhau.

Chương 1

Phân tích mở rộng
Thực nghiệm và so sánh mở rộng

1.1 Vấn đề bài báo gốc

1.2 Mô hình của tác giả

1.2.1 LSTM

1.2.2 Random Forest

1.3 Mô hình của sinh viên

1.3.1 GRU

1.3.2 XGBoost

Giới thiệu

Boosting là một trong các kỹ thuật hoc máy kết hợp phổ biến. Bằng cách kết hợp những mô hình yếu lại với nhau - phổ biến nhất là cây quyết định giống như trong thuật toán Random Forest, Boosting cũng đem lại hiệu quả rất tốt. XGBoost là một trong số những mô hình mạnh và đã giành được rất nhiều những thành công

và cũng đã có một thời kỳ rất áp đảo trên các cuộc thi trên Kaggle. Ý tưởng tổng quan của kỹ thuật boosting là huấn luyện các mô hình dự đoán một cách tuần tự - các mô hình sau sẽ cố gắng sửa lại những gì còn sai sót của các mô hình từ các bước trước đó.

Gradient Boosting

Với ý tưởng tổng quan là cải thiện dự đoán của các mô hình trước, ta có thể hình dung cơ chế hoạt động của Gradient Boosting là như sau Ta gọi tập đầu vào là tập X, tập đầu ra là y. Trước hết ta sẽ tiến hành dự đoán cho cây quyết định thứ nhất. Ở đây ta sẽ sử dụng mô-đun *DecisionTreeRegressor* trong *sci-kit learn*

```
from sklearn.tree import DecisionTreeRegressor
tree_reg1 = DecisionTreeRegressor()
tree_reg1.fit(X, y)
```

Tiếp theo đó, ta sẽ tiếp tục huấn luyện một cây quyết định mới dựa trên lỗi dư thừa (residual error) từ cây quyết định đầu tiên

```
y2 = y - tree_reg1.predict(X)
tree_reg2 = DecisionTreeRegressor()
tree_reg2.fit(X, y2)
```

Tiếp tục lặp lại, như vậy cho đến khi đạt đến một ngưỡng cây nhất định n - một trong những siêu tham số được cài đặt bởi người dùng. Giờ ta đã có một mô hình kết hợp gồm n cây quyết định, và dự đoán của mô hình sẽ được tính bằng tổng dự đoán của các cây

```
trees = [tree_reg1, tree_reg2, ..., tree_regn]
y_pred = sum(tree.predict(X_test) for tree in trees)
```

Dưới đây sẽ là đoạn mã giả mô phỏng cách hoạt động của mô hình Gradient Boosting

Algorithm 1: Gradient Boosting với Cây Quyết Định

Input: Tập dữ liệu (X, y), số lượng cây n_{trees}, tốc độ học α
Output: Dự đoán cho tập kiểm tra $y_{\text{test\_pred}}$
Khởi tạo trees $\leftarrow$ [] (danh sách các cây rỗng);
Function `FitTree`(X, y)**:**
    Huấn luyện một DecisionTreeRegressor trên (X, y);
    return cây đã được huấn luyện;
Bước 1: Huấn luyện cây đầu tiên;
$\text{tree}_1 \leftarrow$ `FitTree`(X, y);
Thêm tree_1 vào trees;
$y_{\text{pred}} \leftarrow \text{tree}_1(X)$ (dự đoán ban đầu);
Bước 2: Huấn luyện các cây tiếp theo;
for $i = 2$ *đến* n_{trees} **do**
    Tính sai sót: $r \leftarrow y - y_{\text{pred}}$;
    $\text{tree}_i \leftarrow$ `FitTree`(X, r);
    Thêm tree_i vào trees;
    Cập nhật dự đoán: $y_{\text{pred}} \leftarrow y_{\text{pred}} + \alpha \cdot \text{tree}_i(X)$;
Bước 3: Thực hiện dự đoán trên tập kiểm tra;
$y_{\text{test\_pred}} \leftarrow \sum_{\text{tree} \in \text{trees}} \alpha \cdot \text{tree}(X_{\text{test}})$;
return $y_{\text{test\_pred}}$;

So sánh Gradient Boosting và XGBoost

Những điểm mạnh và cải tiến của mô hình XGBoost so với Gradient Boosting truyền thống:

- **Điều chuẩn - Regularization:** XGBoost có thêm một số siêu tham số để penalize những mô hình phức tạp thông qua điều chuẩn L1 và L2 giúp tránh việc overfitting.
- **Kiểm soát dữ liệu thiếu hoặc thưa:** Trong một số trường hợp, dữ liệu có thể bị thiếu, hoặc các kỹ thuật tiền xử lý dữ liệu đôi khi có thể làm dữ liệu bị thưa. Tuy nhiên, XGBoost đã được tích hợp sẵn sàng thuật toán chia nhánh có kiểm soát dữ liệu thưa.
- **Thuật toán Weighted Quantile Sketch:** Hầu hết các thuật toán cây hiện có đều chỉ tìm được điểm chia khi các điểm dữ liệu có trọng số bằng nhau (sử dụng thuật toán quantile sketch thông thường). Tuy nhiên, chúng không được thiết kế để xử lý dữ liệu có trọng số. XGBoost có một thuật toán phân tán

weighted quantile sketch, cho phép xử lý hiệu quả dữ liệu có trọng số không đồng đều.

- **Cấu trúc khối để học song song (Block Structure for Parallel Learning):** Để tính toán nhanh hơn, XGBoost có thể sử dụng nhiều lõi (cores) trên CPU. Điều này khả thi nhờ vào cấu trúc khối trong thiết kế hệ thống. Dữ liệu được sắp xếp và lưu trữ trong các đơn vị bộ nhớ gọi là blocks. Khác với các thuật toán khác, cách tiếp cận này cho phép tái sử dụng cấu trúc dữ liệu đã lưu trữ thay vì tính toán lại qua mỗi lần lặp. Tính năng này đặc biệt hữu ích cho các bước như tìm điểm chia (split finding) và lấy mẫu cột (column sub-sampling).

- **Nhận thức bộ nhớ đệm (Cache Awareness):** Trong học máy với XGBoost, ngôn ngữ Scala yêu cầu truy cập bộ nhớ không liên tục để lấy các thống kê gradient theo chỉ mục hàng. Do đó, Tianqi Chen đã thiết kế XGBoost để tối ưu hóa việc sử dụng phần cứng. Quá trình tối ưu này được thực hiện bằng cách cấp phát các bộ đệm nội bộ (internal buffers) trong từng luồng xử lý, nơi mà workflow có thể lưu trữ thống kê gradient. Nhờ đó, các cây được xây dựng song song một cách hiệu quả hơn, đặc biệt khi tận dụng ngôn ngữ Julia và Java.

- **Tính toán ngoài bộ nhớ (Out-of-Core Computing):** Tính năng này tối ưu hóa không gian đĩa có sẵn và tận dụng tối đa khi xử lý các tập dữ liệu lớn không thể vừa trong bộ nhớ RAM.

1.3.3 Stacking

1.4 So sánh và kết luận

Chương 2

Ứng dụng mở rộng vào thực tế

2.1 Ứng dụng vào thực tế

2.2 Mô hình của sinh viên

2.2.1 GRU

2.2.2 Stacking

2.2.3 GRU-XGBoost

2.2.4 LSTM-GRU

2.3 Kết quả và kết luận

Tài liệu tham khảo

[1] Junwei Chen. Analysis of bitcoin price prediction using machine learning. *Journal of Risk and Financial Management*, 16(1):51, 2023.