BÁO CÁO TUẦN 3

Người báo cáo: Trần Đức Thọ

**Mô tả thuật toán SVM:**

Thuật toán SVM là thuật toán dùng được trong cả các bài toán phân loại và hồi quy, nhưng chúng ta thường sử dụng chúng trong các bài toàn phân loại bởi trong các bài toán có số các feature cao, SVM có thể phân loại chính xác hơn các bài toán phân loại khác.

SVM là phương pháp học có giám sát.

Với các điểm dữ liệu và các nhãn lớp gắn liền với các điểm dữ liệu cho trước, ý tưởng của bài toán SVM là chúng ta có thể tạo 1 siêu phẳng phân chia thành 2 nữa (ở đây em ví dụ với 2 nhãn lớp). để tìm kiếm 1 đường thẳng (siêu phẳng) để phân chia thành 2 lớp thì có rất nhiều trường hợp xảy ra, thuật toán SVM hướng đến chúng ta tìm kiếm được 1 siêu phẳng sao cho có lề lớn nhất.

Chart, scatter chart

Description automatically generated

Khi tập dữ liệu có nhiễu, thuật toán SVM có khả năng thực hiện khá tốt trong việc xử lí nhiễu bằng việc thêm tham số slack.

Với các kiểu dữ liệu phi tuyến như hình dưới:

Chart, scatter chart

Description automatically generated

Chúng ta có thể giải quyết bài toán này bằng cách ánh xạ các điểm dữ liệu trên sang 1 hàm khác (tăng số chiều của bài toán lên)

**Phân tích paper “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”:**

Tổng quan: Gradient Boosting Decision Tree là thuật toán học máy được sử dụng rộng rãi, do hiệu quả độ chính xác và khả năng diễn giải của nó, nó thường được nhắc nhiều trong các tác vụ học máy như phân loại nhiều lớp , xếp hạng, dự đoán.

Với sự phát triển của dữ liệu lớn, chúng ta luôn phải cân nhắc giữa hiệu quả và chi phí với từng loại mô hình.

Gradient Boosting Decision Tree luôn phải quét toàn bộ các trường hợp nên khi số lượng instances và số features tăng, độ phức tạp tính toán tăng.

Để giải quyết bài toán này, ý tưởng là giảm số lượng features và số instances, nhưng để thực hiện việc này thì rất khó.

Bài báo đề xuất ra 2 ý tưởng đó là Gradient-based One-Side Sampling (GOSS) và Exclusive Feature Bundling (EFB)

* Gradient-based One-Side Sampling (GOSS): Nên giữ tốc độ học lớn với 1 số cá thể dữ liệu.
* Exclusive Feature Bundling (EFB): với các features độc quyền, tức là hiếm khi chúng hiếm khi nhận các giá trị đồng thời, chúng ta đóng gói đồng thời tập trung

Phân tích thuật toán Gradient Boosting Decision Tree: thuật toán GBDT là thuật toán phân tích từ các điểm dữ liệu từ ra thành các nút lá, từ độ chênh lệch của đầu ra thực tế và trên lí thuyết, chúng ta sẽ xây dựng tốc độ học cho bài toán, cứ tiếp tục các vòng lặp như vậy cho đến khi độ chênh lệch nhỏ hơn 1 ngưỡng nào đó thì có thể dừng.

Tức là ở đây chúng ta cần tốn O(#data × #feature) cho việc xây dựng biểu đồ và O(#bin × #feature) cho việc tìm điểm phân chia. Như vậy, nếu có thể giảm số data và số feature xuống chúng ta có thể rất có lợi và đẩy nhanh tiến độ thuật toán GBDT.

Có một cách để có thể giảm kích thước đào tạo đó là phương pháp lấy mẫu với tỉ lệ nhất định, tỉ lệ sẽ liên tục được chuyển đổi trong quá trình traning mô hình để đưa ra tỉ lệ lấy mẫu một cách hợp lý nhất. Tuy nhiên thuật toán GBDT không thể áp dụng phương pháp lấy mẫu vì nó không có trọng số gốc cho dữ liệu, không thế đánh giá để lất mẫu được.

Tiếp theo là vấn đề giảm số lượng các tính năng, chúng ta nên lọc các tính năng yếu. Các tính ăng thường được phân tích thành phần nguyên tắc hoặc đuổi theo phép chiếu. Tuy nhiên nhưng tính năng này dựa trên giả định các tính năng chứa những dư thừa đáng kể, tuy nhiên điều này hầu như khó xảy ra trong thực tế.

Diễn giải chi tiết:

GOSS:

GOSS (Lấy mẫu một mặt dựa trên Gradient) là một phương pháp lấy mẫu mới để lấy mẫu các thể hiện trên cơ sở gradient. Như chúng ta biết các trường hợp có độ dốc nhỏ được đào tạo tốt (lỗi đào tạo nhỏ) và những trường hợp có độ dốc lớn đang được đào tạo. Một cách tiếp cận đơn giản đối với mẫu giảm là loại bỏ các trường hợp có độ dốc nhỏ bằng cách chỉ tập trung vào các trường hợp có độ dốc lớn nhưng điều này sẽ làm thay đổi phân phối dữ liệu. Tóm lại, GOSS giữ lại các trường hợp có độ dốc lớn trong khi thực hiện lấy mẫu ngẫu nhiên trên các trường hợp có độ dốc nhỏ.

Các bước trực quan để tính toán GOSS

- Sắp xếp các thể hiện theo độ dốc tuyệt đối theo thứ tự giảm dần

- Chọn các trường hợp a \* 100% hàng đầu. [Dưới độ dốc được đào tạo / lớn]

- Lấy mẫu ngẫu nhiên b \* 100% các trường hợp từ phần còn lại của dữ liệu. Điều này sẽ làm giảm sự đóng góp của các ví dụ được đào tạo tốt theo hệ số b (b <1)

- Nếu không có điểm 3, số lượng mẫu có độ dốc nhỏ sẽ là 1-a (hiện tại là b). Để duy trì phân phối ban đầu, LightGBM khuếch đại sự đóng góp của các mẫu có độ dốc nhỏ bằng hằng số (1-a) / b để tập trung nhiều hơn vào các cá thể chưa được đào tạo. Điều này tập trung nhiều hơn vào các trường hợp chưa được đào tạo mà không làm thay đổi nhiều phân phối dữ liệu.

EFB:

Hãy nhớ việc xây dựng biểu đồ lấy O (#data \* #feature). Nếu chúng tôi có thể lấy mẫu #feature, chúng tôi sẽ tăng tốc độ học cây. LightGBM đạt được điều này bằng cách kết hợp các tính năng lại với nhau. Chúng tôi thường làm việc với dữ liệu có kích thước cao. Dữ liệu như vậy có nhiều tính năng loại trừ lẫn nhau, tức là chúng không bao giờ nhận các giá trị 0 đồng thời. LightGBM xác định một cách an toàn các tính năng như vậy và gói chúng thành một tính năng duy nhất để giảm độ phức tạp thành O (#data \* #bundle) trong đó #bundle << #feature.

Giải thích trực quan để tạo các gói tính năng

-Xây dựng một biểu đồ với các cạnh có trọng số (đo xung đột giữa các đối tượng địa lý). Xung đột là thước đo phần nhỏ các đối tượng địa lý độc quyền có các giá trị khác 0 chồng chéo lên nhau.

-Sắp xếp các tính năng theo số lượng các trường hợp khác 0 theo thứ tự giảm dần.

-Lặp lại danh sách các tính năng được sắp xếp theo thứ tự và gán tính năng cho một gói hiện có (nếu xung đột < ngưỡng) hoặc tạo một gói mới (nếu xung đột > ngưỡng).

Các bước hợp nhất tính năng

-Tính toán phần bù sẽ được thêm vào mọi tính năng trong gói tính năng.

-Lặp lại mọi đối tượng và tính năng dữ liệu.

-Khởi tạo nhóm mới bằng 0 đối với các trường hợp trong đó tất cả các tính năng đều bằng 0.

-Tính toán nhóm mới cho mọi trường hợp khác 0 của đối tượng bằng cách thêm phần bù tương ứng vào nhóm ban đầu của đối tượng đó.

Tóm lại:

Bài báo mong muốn chúng ta hiểu 2 trụ cột chính trong LightGBM: GOSS và EFB:

GOSS: chúng ta chọn lọc các dữ liệu có độ dốc lớn và chỉ lấy 1 phần dữ liệu có độ dốc bé (lấy mẫu)

EBF: hợp nhất các tính năng

2 phương pháp trên đã giúp giảm thiểu thời gian xử lý dữ liệu lớn với nhiều tính năng giảm thiểu thời gian chạy so với các mô hình truyền thống