3.4.3. Phép phân tích thành phần chính

Giả sử rằng dữ liệu được giảm bớt bao gồm các bộ giá trị hoặc vectơ dữ liệu được mô tả bởi n thuộc tính hoặc thứ nguyên.

Phép phân tích thành phần chính (PCA) tìm kiếm k vectơ trực giao n-chiều có thể được sử dụng tốt nhất để biểu diễn dữ liệu, trong đó k ≤ n.

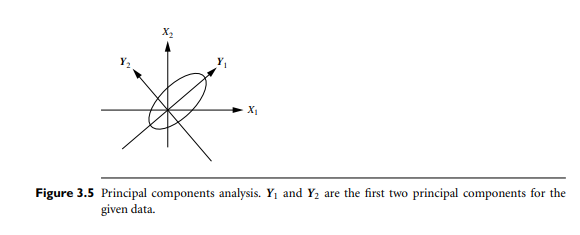
Do đó, dữ liệu ban đầu được chiếu vào một không gian nhỏ hơn nhiều, dẫn đến giảm kích thước.

Không giống như Trích chọn đặc trưng mà làm giảm kích thước tập hợp thuộc tính bằng cách giữ lại một tập hợp con của tập thuộc tính ban đầu, PCA “kết hợp” bản chất của các thuộc tính bằng cách tạo một tập hợp biến thay thế nhỏ hơn.

Dữ liệu ban đầu sau đó có thể được chiếu vào tập hợp nhỏ hơn này.

PCA thường tiết lộ các mối quan hệ mà trước đây không bị nghi ngờ và do đó cho phép các diễn giải mà thông thường không dẫn đến kết quả.

1. Dữ liệu đầu vào được chuẩn hóa để mỗi thuộc tính nằm trong cùng một phạm vi. Bước này giúp đảm bảo rằng các thuộc tính có miền lớn sẽ không lấn át các thuộc tính có miền nhỏ hơn.
2. PCA tính k vectơ trực chuẩn cung cấp cơ sở cho dữ liệu đầu vào chuẩn hóa. Đây là các vectơ đơn vị mà mỗi điểm theo phương vuông góc với các vectơ khác. Các vectơ này được gọi là các thành phần chính. Dữ liệu đầu vào là sự kết hợp tuyến tính của các thành phần chính.
3. Các thành phần chính được sắp xếp theo thứ tự giảm dần “mức độ quan trọng” hoặc độ mạnh. Các thành phần chính về cơ bản đóng vai trò như một tập hợp các trục mới cho dữ liệu, cung cấp thông tin quan trọng về phương sai. Nghĩa là, các trục được sắp xếp sao cho trục đầu tiên hiển thị phương sai nhiều nhất trong số dữ liệu, trục thứ hai hiển thị phương sai cao nhất tiếp theo, v.v. Ví dụ, Hình 3.5 cho thấy hai thành phần chính đầu tiên, Y1 và Y2, cho tập dữ liệu đã cho ban đầu được ánh xạ tới các trục X1 và X2. Thông tin này giúp xác định các nhóm hoặc mẫu trong dữ liệu.



1. Bởi vì các thành phần được sắp xếp theo thứ tự giảm dần về “mức độ quan trọng”, nên kích thước dữ liệu có thể được giảm xuống bằng cách loại bỏ các thành phần yếu hơn, tức là những thành phần có độ lớn biến thiên thấp. Sử dụng các thành phần chính mạnh nhất, sẽ có thể tạo lại một số liệu gần đúng tốt của dữ liệu ban đầu.

PCA có thể được áp dụng cho các thuộc tính có thứ tự và không có thứ tự, đồng thời có thể xử lý dữ liệu thưa thớt và dữ liệu lệch. Dữ liệu đa chiều của nhiều hơn hai chiều có thể được xử lý bằng cách giảm vấn đề xuống hai chiều. Các thành phần chính có thể được sử dụng làm đầu vào cho phân tích hồi quy nhiều lần và phân tích theo cụm. So với các phép biến đổi wavelet, PCA có xu hướng xử lý dữ liệu thưa thớt tốt hơn, trong khi các phép biến đổi wavelet phù hợp hơn với dữ liệu có kích thước cao

3.4.4. Trích chọn đặc trưng

(nói thêm)

Các tập dữ liệu để phân tích có thể chứa hàng trăm thuộc tính, nhiều thuộc tính có thể không phù hợp với nhiệm vụ khai thác hoặc dư thừa. Ví dụ: nếu nhiệm vụ là phân loại khách hàng dựa trên việc họ có khả năng mua một đĩa CD mới phổ biến tại AllElectronics hay không khi được thông báo về việc giảm giá, thì các thuộc tính như số điện thoại của khách hàng có thể không liên quan, không giống như các thuộc tính như tuổi hoặc gu âm nhạc. Mặc dù chuyên gia miền có thể chọn ra một số thuộc tính hữu ích, nhưng đây có thể là một nhiệm vụ khó khăn và tốn thời gian, đặc biệt là khi hành vi của dữ liệu không được biết rõ. Việc loại bỏ các thuộc tính có liên quan hoặc giữ lại các thuộc tính không liên quan có thể gây bất lợi, gây nhầm lẫn cho thuật toán khai thác được sử dụng. Điều này có thể dẫn đến việc phát hiện ra các mẫu có chất lượng kém. Ngoài ra, khối lượng bổ sung của các thuộc tính không liên quan hoặc dư thừa có thể làm chậm quá trình khai thác

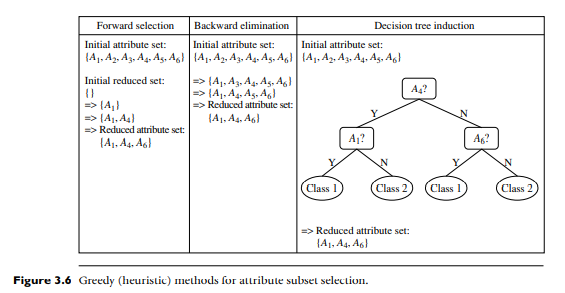
(có thể làm power point từ phần này)

**Trích chọn đặc trưng** làm giảm kích thước tập dữ liệu bằng cách loại bỏ các thuộc tính (hoặc thứ nguyên) không liên quan hoặc dư thừa. Mục tiêu của Trích chọn đặc trưng là tìm một tập hợp tối thiểu các thuộc tính sao cho phân phối xác suất kết quả của các lớp dữ liệu càng gần với phân phối ban đầu càng tốt bằng cách sử dụng tất cả các thuộc tính. Khai thác trên một tập hợp các thuộc tính giảm có một lợi ích bổ sung: Nó làm giảm số lượng các thuộc tính xuất hiện trong các mẫu đã phát hiện, giúp làm cho các mẫu dễ hiểu hơn.

Đối với n thuộc tính, có thể có hai mũ n tập con. Một tìm kiếm toàn diện cho tập hợp con tối ưu của các thuộc tính có thể rất tốn kém, đặc biệt là khi n và số lượng lớp dữ liệu tăng lên. Do đó, các phương pháp suy nghiệm **(heuristic methods)** khám phá không gian tìm kiếm thu gọn thường được sử dụng để lựa chọn tập con thuộc tính. Các phương thức này tham lam ở chỗ, trong khi tìm kiếm trong không gian thuộc tính, chúng luôn làm những gì được cho là lựa chọn tốt nhất tại thời điểm đó. Chiến lược của các phương thức này là đưa ra một lựa chọn tối ưu cục bộ với hy vọng rằng điều này sẽ dẫn đến một giải pháp tối ưu trên toàn cầu. Các phương pháp tham lam như vậy có hiệu quả trong thực tế và có thể ước tính một giải pháp gần tối ưu.

Các thuộc tính “tốt nhất” (và “tồi tệ nhất”) thường được xác định bằng cách sử dụng các thử nghiệm có ý nghĩa thống kê, giả định rằng các thuộc tính độc lập với nhau. Nhiều biện pháp đánh giá thuộc tính khác có thể được sử dụng chẳng hạn như thước đo thu thập thông tin được sử dụng trong việc xây dựng cây quyết định để phân loại.

Các phương pháp suy nghiệm cơ bản của việc lựa chọn tập hợp con thuộc tính bao gồm các kỹ thuật sau đây, một số trong số đó được minh họa trong Hình 3.6



1. **Lựa trọn chuyển tiếp từng bước (Stepwise forward selection)**: Thủ tục bắt đầu với một tập hợp rỗng các thuộc tính là tập hợp đã rút gọn. Các thuộc tính gốc tốt nhất được xác định và thêm vào tập hợp đã rút gọn. Ở mỗi bước hoặc lần lặp tiếp theo, các thuộc tính gốc tốt nhất còn lại sẽ được thêm vào tập hợp.
2. **Lược bỏ ngược từng bước (Stepwise backward elimination)**: Thủ tục bắt đầu với tập hợp đầy đủ các thuộc tính. Ở mỗi bước, nó loại bỏ thuộc tính xấu nhất còn lại trong tập hợp.
3. **Kết hợp giữa “Lựa trọn chuyển tiếp” và “Lược bỏ ngược”:** Tại mỗi bước, thủ tục chọn thuộc tính tốt nhất và loại bỏ thuộc tính xấu nhất trong số các thuộc tính còn lại.
4. **Cảm ứng cây quyết định (Decision tree induction):** Các thuật toán cây quyết định ban đầu được dùng để phân loại. Quy nạp cây quyết định xây dựng một sơ đồ giống như cấu trúc trong đó mỗi nút bên trong (không phải lá) biểu thị một bài kiểm tra trên một thuộc tính, mỗi nhánh tương ứng với một kết quả của bài kiểm tra và mỗi nút bên ngoài (lá) biểu thị một dự đoán lớp. Tại mỗi nút, thuật toán chọn thuộc tính "tốt nhất" để phân vùng dữ liệu thành các lớp riêng lẻ.

Khi quy nạp cây quyết định được sử dụng để lựa chọn tập hợp con thuộc tính, một cây được cấu trúc từ dữ liệu đã cho. Tất cả các thuộc tính không xuất hiện trong cây được coi là không liên quan. Tập hợp các thuộc tính xuất hiện trong cây tạo thành tập hợp con rút gọn của các thuộc tính.

Các tiêu chí dừng cho các phương pháp có thể khác nhau. Quy trình có thể sử dụng một ngưỡng trên thước đo được sử dụng để xác định thời điểm dừng quá trình lựa chọn thuộc tính.

Trong một số trường hợp, chúng tôi có thể muốn tạo các thuộc tính mới dựa trên các thuộc tính khác. Việc xây dựng thuộc tính như vậy có thể giúp cải thiện độ chính xác và sự hiểu biết về cấu trúc trong dữ liệu có chiều cao. Ví dụ: chúng ta có thể muốn thêm vùng thuộc tính dựa trên chiều cao và chiều rộng của thuộc tính. Bằng cách kết hợp các thuộc tính, cấu trúc thuộc tính có thể khám phá thông tin còn thiếu về mối quan hệ giữa các thuộc tính dữ liệu, từ đó có thể hữu ích cho việc khám phá kiến thức.