**I) Các kiến thức cần biết trước khi tìm hiểu PCA**

**1. Eigenvalue và EigenVector**

Giá trị riêng(Eigenvalue) và Vector riêng(Eigenvector): Một vector x khác không được gọi là vector riêng( Vector đặc trưng) của ma trận vuông A nếu tồn tại một hằng số vô hướng sao cho:

Các tính chất của Eigenvector và Eigenvalue

Eigenvector(Vector riêng):

* Là một vector không thay đổi hướng khi bị biến đổi bởi một ma trận tuyến tính.
* Nó chỉ bị co giãn hoặc đảo chiều, chứ không bị xoay theo các hướng khác.

Eigenvalue(Giá trị riêng): Là một **hệ số tỉ lệ** mô tả mức độ co giãn hoặc đảo chiều của vector riêng khi bị biến đổi.

* Một ma trận có thể có **nhiều** eigenvalues và eigenvectors.
* Eigenvectors **không được bằng vector 0**.
* Eigenvalues có thể **âm, dương hoặc bằng 0**.

Cách tìm Eigenvalues và Eigenvectors của ma trận:  
Vd: Tìm Eigenvalues và Eigenvectors của ma trận:

Tìm A- và giải phương trình, ta có 2 giá trị (Eigenvalues):

Thay ngược lại 2 giá trị eigenvalues để giải phương trình A-

Trường hợp

A number and number in a row

AI-generated content may be incorrect.

Tương đương với: , chọn x = 1, y = 1 thì eigenvector:

Làm tương tự với ta được

PCA sử dụng eigenvectors của ma trận hiệp phương sai để tìm hướng chính của dữ liệu.

Eigenvalues quyết định tầm quan trọng của từng hướng. Hay nói cách khác, Eigenvalue là hệ số co giãn của vector riêng sau khi biến đổi và vector không đổi hướng khi biến đổi.

**2. Ma trận hiệp phương sai(Covarrience Matrix)**

Ma trận hiệp phương sai (Covariance Matrix) là một ma trận vuông biểu diễn hiệp phương sai giữa các biến trong một tập dữ liệu. Nó giúp ta hiểu mức độ liên quan và tương quan giữa các biến.

Giả sử có 2 biến ngẫu nhiên X và Y, hiệp phương sai của chúng được tính bằng:

: Giá trị của biến tại quan sất thứ i.

Giá trị trung bình của các biến.

Nếu hiệp phương sai dương → hai biến có mối quan hệ đồng biến (cùng tăng hoặc cùng giảm).  
Nếu hiệp phương sai âm → hai biến có mối quan hệ nghịch biến.  
Nếu hiệp phương sai bằng 0 → hai biến không có mối quan hệ tuyến tính.

Ma trận hiệp phương sai

Với p biến ngẫu nhiên ma trận hiệp phương sai là một ma trận pxp

* Các phần tử trên đường chéo là phương sai của từng biến: .
* Các phần tử ngoài đường chéo là hiệp phương sai giữa các biến.

Ứng dụng:  
- Giảm chiều dữ liệu (PCA - Phân tích thành phần chính): Giúp xác định các hướng có phương sai lớn nhất để giảm số chiều dữ liệu.

- Phân tích mối quan hệ giữa các biến: Xác định xem biến nào có ảnh hưởng đến nhau.

- Mô hình hóa thống kê: Dùng trong phân phối chuẩn đa biến, kiểm tra tương quan giữa các biến.

Sự khác nhau giữa hiệp phương sai và Cosine Similarity:

Điểm giống nhau

Cả hai đều đo lường **mức độ tương quan** giữa hai tập dữ liệu.

Nếu hai biến có mối quan hệ tuyến tính mạnh, thì cả hiệp phương sai và cosine similarity có thể cho thấy điều đó.

Điểm khác nhau

|  |  |  |
| --- | --- | --- |
|  | Hiệp phương sai(Covariance) | Cosine Similarity |
| Khái niệm | Đo mức độ biến đổi cùng nhau giữa 2 biến | Đo độ giống nhau về hướng giữa 2 vector |
| Giá trị | Có thể dương, âm hoặc bằng 0 | Luôn nằm trong khoảng [-1, 1] |
| Ảnh hưởng của độ lớn | Bị ảnh hưởng bởi đơn vị đo lường của dữ liệu | Không bị ảnh hưởng bởi độ lớn của vector |

Dùng hiệp phương sai khi muốn biết mức độ thay đổi của hai biến (có thể âm hoặc dương).

Dùng cosine similarity khi muốn đo độ giống nhau về hướng mà không quan tâm đến độ lớn.

3. Giảm chiều dữ liệu(Dimensionality Reduction)

Giảm chiều dữ liệu(Dimensionality Reduction) là quá trình giảm số lượng biến(Features) trong tập dữ liệu nhưng vẫn giữa được các thông tin quan trọng.

Lý do cần giảm chiều dữ liệu:

* Tránh Curse of Dimensionality(Lời nguyền chiều cao): Khi số chiều tăng cao, dữ liệu trở nên thưa thớt.
* Giảm thời gian tính toán: Ít đặc trưng thì thuật toán chạy nhanh hơn.
* Tránh overfitting: Khi có quá nhiều biến dư thừa, mô hình có thể sẽ học quá kỹ vào dữ liệu train và kém hiệu quả trên dữ liệu mới.
* Dễ dàng trực quan hóa dữ liệu: Dữ liệu có 2 hoặc 3 chiều để biểu diễn dễ hơn so với dữ liệu hàng trăm chiều

Nói dễ hiểu hơn, là chúng ta đi tìm một hàm số, Hàm số này lấy đầu vào là một điểm dữ liệu ban đầu với D rất lớn, và tạo ra một điểm dữ liệu mới mà có số chiều K<D.

4. Norm(Chuẩn)

Một Norm trên không gian vector V là một hàm thỏa mãn các tính chất sau với mọi

Tính đồng nhất tuyệt đối(Absolutely homogeneous):

Bất đẳng thức tam giác(Triangle inequality):

Tính xác định dương(Positive definite):

Chuẩn Manhattan(Manhattan Norm norm):

Chuẩn manhattan của một vector được định nghĩa là:

* Đây là tổng độ lớn tuyệt đối của các thành phần của vector.
* Còn gọi là chuẩn đường phố (taxicab norm) vì nó đo khoảng cách bằng cách đi theo trục tọa độ, giống như cách một chiếc taxi di chuyển trên các đường phố vuông góc.

Chuẩn Euclid(Euclidean Norm - norm)

Chuẩn Euclid của một vector x được định nghĩa là:

* Đây là khoảng cách thông thường trong không gian Euclid, đo bằng cách sử dụng định lý Pythagoras.
* Nó là chiều dài của vector từ gốc tọa độ đến điểm đại diện cho vector.

A black background with a triangle and numbers

AI-generated content may be incorrect.

Hình ảnh trên: Chuẩn Manhattan(Màu đỏ) và chuẩn Euclid(Màu xanh).

Khoảng cách Manhattan là tổng khoảng cách trên các trục tọa độ.

Khoảng cách euclid là đường thẳng nối 2 điểm.

1.4.2 Norm của 2 ma trận

Giả sử hàm số là một norm bất kỳ của x. Ứng với norm này với ma trận A ta có norm:

A là một ma trận có kích thước m x n, có n cột

x là một vector thuộc

Ax là một vector thuộc , vì phép nhân ma trận A với vector x cho ra vector có số chiều bằng số hàng của A.

Bài toán này thành bài toán tìm giá trị lớn nhất của tỷ số norm của Ax(Vector đầu ra) và norm của x(vector đầu vào).

Chuẩn ma trận đặc trưng cho cách mà AAA biến đổi các vector đầu vào trong không gian.

Ví dụ:  
Với norm Euclid(:  
Nếu chọn norm Euclid, norm của ma trận A trở thành

Đây chính là giá trị riêng lớn nhất của ma trận (Nếu A vuông, đây là giá trị kỳ dị lớn nhất của A)

Với norm Manhattan(

Tương ứng với tổng lớn nhất của các phần tử trong các cột của A.

Giờ ta xét Norm2:

Nếu x là nghiệm tối ưu của bài toán tối ưu (1) thì kx cũng là nghiệm với k là một số thực khác 0. Giả sử mẫu số = 1 và giữ lại tử, khi đó bài toán tối ưu được viết lại thành:

Vậy ta cần tìm x sao cho:

Chúng ta sử dụng phương pháp nhân tử Lagrange của (2):

Nghiệm của (2) sẽ thỏa mãn với hệ phương trình:

Từ (4) ta có: (5)

Ta thấy theo (5) thì là một giá trị riêng của và x là một vector riêng ứng với giá trị riêng. Nhân 2 vế của (5) với ta được:

là giá trị riêng lớn nhất của hay chính là singular value lớn nhất của ma trận A

Như vậy norm 2 của một ma trận chính là singular value lớn nhất của ma trận đó.Và nghiệm của (2) chính là right-singular vector ứng với singular value đó:

Bài toán:

Có nghiệm là vector riêng ứng với giá trị riêng nhỏ nhất của . Khi đó hàm số đạt giá trị nhỏ nhất chính là giá trị riêng nhỏ nhất

1.5 Biểu diễn Vector trong các hệ cơ sở khác nhau

Trong không gian D chiều , toạ độ của mỗi điểm được xác định dựa trên một hệ toạ độ nào đó. Ở các hệ toạ độ khác nhau, hiển nhiên là toạ độ của mỗi điểm cũng khác nhau.

Mỗi vector cột x = , biểu diễn của nó trong hệ đơn vị:

Một hệ cơ sở khác (Các vector độc lập tuyến tính). Biểu diễn của vector x trong hệ cơ sở mới có dạng:

U là ma trận mà cột thứ d của nó là vector , vector y chính là biểu diễn của x trong hệ cơ sở mới, y sẽ được tính bằng:

U là ma trận khả nghịch vì các cột của U là độc lập tuyến tính, U là các ma trận trực giao, nên:

Nên ta có:

A diagram of a triangle with a red arrow

AI-generated content may be incorrect.

1.6 Trace

Trace là tổng các phần tử trên đường chéo của một ma trận vuông

Các tính chất của Trace:

: Với A là một ma trận vuông và là toàn bộ các giá trị riêng.

1.7 Kì vọng và ma trận hiệp phương sai

Cho N giá trị . Kì vọng và phương sai của bộ dữ liệu này là:

Kỳ vọng là trung bình cộng của toàn bộ các giá trị

Phương sai là trung bình cộng của bình phương khoảng cách từ mỗi điểm tới kỳ vọng. Phương sai càng nhỏ thì chúng càng gần.

A diagram of a function

AI-generated content may be incorrect.

**1.8 Ma trận xác định dương và ma trận nửa xác định dương**

**1.8.1 Ma trận xác định dương**

Một ma trận vuông là xác định dương nếu:

Khi nhân bất kỳ vector nào với A, rồi lấy tích vô hướng với chính vector đó, kết quả luôn dương.

Ma trận này cong , không phẳng, không lõm — hình dung như "hàm parabol úp lên" trong không gian.

Tính chất:

* : A phải là đối xứng
* Tất cả giá trị riêng của A đều dương:
* A khả nghịch

**1.8.2 Ma trận xác định dương(PSD)**

Một ma trận vuông là xác định dương nếu:

Giống như ma trận xác định dương nhưng cho phép trường hợp bằng 0.

Nếu A là ma trận hiệp phương sai thì luôn luôn là ma trận xác định dương.

**II.PCA(Principal Component Analysis)**

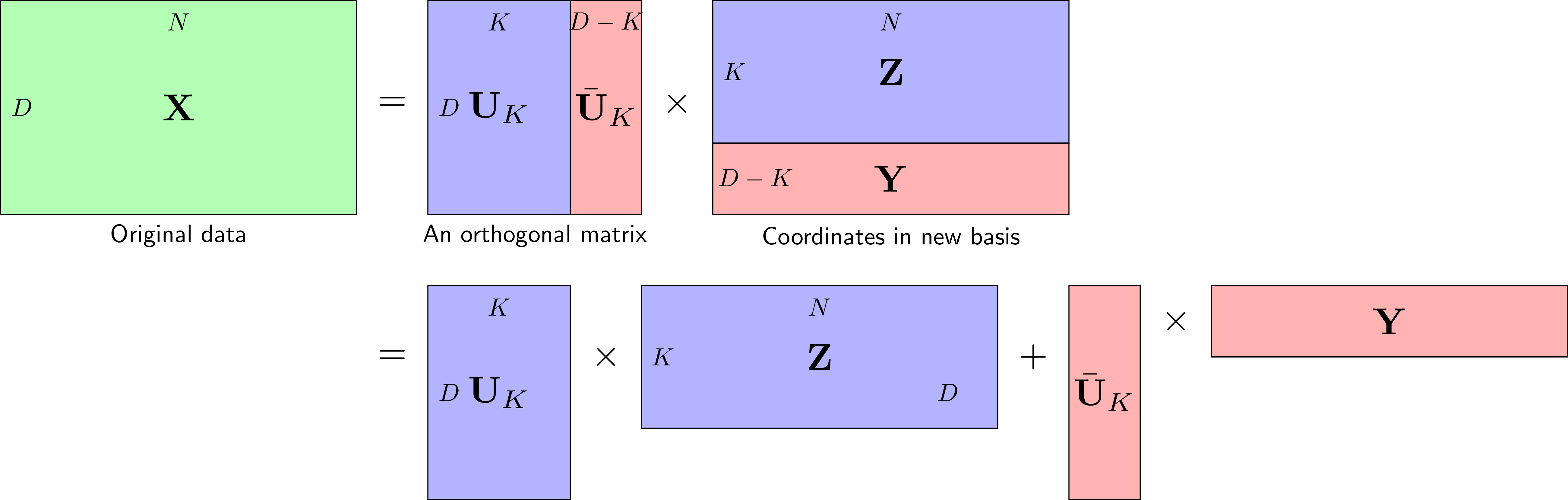
1. Ý tưởng PCA

Giống như phương pháp SVD, cách đơn giản nhất để giảm chiều dữ liệu từ D về K(K<D) là chỉ giữ lại K phần tử quan trọng nhất. Tuy nhiên cách này không phải là cách tốt nhất trong trường hợp lượng thông tin đều có độ quan trọng ngang nhau.

Nhưng nếu chúng ta có thể biểu diễn các vector dữ liệu ban đầu trong một hệ cơ sở mới mà trong hệ đó tầm quan trọng giữa các thành phần khác là rõ rệt thì việc loại bỏ là hoàn toàn có thể.

Một ví dụ tương tự và dễ hiểu hơn là việc ghi âm giọng nói trong một căn phòng có nhiều micro. Giả sử chúng ta muốn thu âm giọng nói của một người và có hai micro: một micro được đặt ngay trước miệng người nói và một micro khác được đặt xa hơn, ở góc phòng. Rõ ràng, micro đặt trước miệng sẽ thu được âm thanh rõ ràng hơn, ít nhiễu hơn và chứa phần lớn thông tin quan trọng về giọng nói. Trong khi đó, micro ở góc phòng cũng thu được âm thanh nhưng lại bị lẫn nhiều tạp âm từ môi trường, như tiếng vang, tiếng quạt hay âm thanh từ bên ngoài. Vì vậy, trong quá trình xử lý dữ liệu, chúng ta có thể bỏ qua dữ liệu từ micro thứ hai mà không làm mất quá nhiều thông tin cần thiết. Điều này minh họa cách chọn một hệ cơ sở phù hợp để giữ lại thông tin quan trọng nhất, đồng thời loại bỏ những thành phần kém quan trọng.

PCA là phương pháp đi tìm một hệ cơ sở mới sao cho thông tin của dữ liệu chủ yếu tập trung ở một vài tọa độ, phần còn lại chỉ mang một lượng nhỏ thông tin, PCA sẽ tìm một hệ trực chuẩn để làm cơ sở mới. Hay nói cách khác, PCA sẽ tìm một hệ trực chuẩn mới sao cho trong hệ này, các thành phần quan trọng nhất trong K thành phần đầu tiên.



X là trận dữ liệu ban đầu mà chúng ta mong muốn giảm chiều với 2 thành phần chính:

* N là số mẫu dữ liệu
* D là chiều của ma trận

Phân tách cơ sở( và ):

* (Kích thước D x K) chứa K thành phần chính quan trọng nhất.
* (Kích thước D x (D – K)) chứa các thành phần ít quan trọng hơn và có thể loại bỏ

Dữ liệu trên không gian mới:

* Dữ liệu gốc X được chiếu lên hệ cơ sở , tạo ra ma trận Z có kích thước K x N
* Phần dữ liệu Y(Kích thước (D-K) x N) chứa thông tin kém quan trọng hơn và có thể bỏ qua.

Ta có thể thấy cơ sở mới U = [, là một hệ trực chuẩn với là ma trận con tạo bởi K cột đầu tiên của U. với Cơ sở mới thì có thể viết lại:

Ta nhân cả 2 vế của phương trình trên với ta được:

Vì U là ma trận trực chuẩn nên ta có:

Sau khi nhân 2 ma trận trực chuẩn này ta có:

Vì các vector trong U trực giao nên:

(Vì cả 2 ma trận là trực giao)

Nên là:

Vậy:

Mục đích của PCA là đi tìm ma trận trực giao UU sao cho phần lớn thông tin được giữ lại ở phần màu xanh và phần màu đỏ  sẽ được lược bỏ và thay bằng một ma trận không phụ thuộc vào từng điểm dữ liệu. Nói cách khác, ta sẽ xấp xỉ Y bởi một ma trận có toàn bộ các cột là như nhau. Gọi mỗi cột đó là b(Coi nó là Bias):

là vector hàng có toàn bộ các phần tử bằng 1. Nếu đã tìm được U, ta cần tìm b thỏa mãn:

Giải phương trình đạo hàm theo b của hàm mục tiêu bằng 0:

Nếu vector kì vọng (Trường hợp chúng ta trừ môi vector dữ liệu đi vector kì vọng của toàn bộ dữ liệu,

Với giá trị b tìm được thì dữ liệu ban đầu sẽ được xấp xỉ với:

Ta định nghĩa hàm mất mát chính như sau:

Nếu các cột của ma trận V tạo thành một hệ trực chuẩn thì với một ma trận W bất kỳ, ta luôn có:

Vì vậy hàm mất mát có thể được biến đổi:

Với là dữ liệu đã chuẩn hóa và S là ma trận hiệp phương sai của dữ liệu. Ta gọi ma trận này là là dữ liệu đã chuẩn hóa với S là ma trận hiệp phương sai của dữ liệu. Ta gọi ma trận này là Zero-Corected data(Dữ liệu đã chuẩn hóa).

Bài toán chúng ta cần giải quyết là tìm các để mất mát là nhỏ nhất. Thay thử K vào (6) ta có:

Với là các giá trị riêng của ma trận nửa xác định dương S. Chú ý rằng các giá trị riêng này là thực và không âm.

Như vậy L không phụ thuộc vào cách chọn ma trận trực giao U và bằng tổng các phần tử trên đường chéo của S. L chính là tổng của các phương sai theo từng thành phần của dữ liệu ban đầu.

Vì vậy, việc tối thiểu hàm mất mát J tương đương với việc tối đa:

Định lý 1: F đạt giá trị lớn nhất bằng khi là các vector riêng có norm 2 bằng 1 ứng với các giá trị riêng với các trực giao. K chính là giá trị riêng lớn nhất của ma trận hiệp phương sai S. Giá trị riêng lớn nhất đợc gọi là thành phần chính thứ nhất, là thành phần chính thứ 2. Nên phương pháp này gọi là phân tích thành phần chính. Ta chỉ giữ lại K thành phần chính của dữ liệu khi muốn giảm chiều dữ liệu.

Đoạn này thừa nhận chứ không chứng minh.

2.2 Các bước thực hiện PCA

Bước 1: Tính Vector kỳ vọng của toàn bộ dữ liệu:

Bước 2: Trừ mỗi điểm dữ liệu đi vector kỳ vọng của toàn bộ dữ liệu:

Bước 3: Tính ma trận hiệp phương sai:

Bước 4: Tính các giá trị riêng và vector riêng có norm bằng 1 của ma trận này, sắp xếp theo thứ tự giảm dần của giá trị riêng.

Bước 5: Chọn K vector riêng ứng với K trị riêng lớn nhất để xây dựng ma trận có các cột tạo thành một hệ trực giao. Các vector K được gọi là các thành phần chính, tạo thành một không gian con gần với phân bố của dữ liệu ban đầu đã chuẩn hóa.

Bước 6: Chiếu dữ liệu ban đầu đã chuẩn hóa xuống không gian con tìm được

Bước 7: Dữ liệu mới chính là tọa độ của các điểm dữ liệu trên không gian mới:

Dữ liệu ban đầu có thể được tính xấp xỉ theo dữ liệu mới như sau:

A diagram of a graph

AI-generated content may be incorrect.

**2. Mối quan hệ giữa PCA và SVD**

**2.1 Nhắc lại SVD**

SVD cho bài toán xấp xỉ low-rank tốt nhất:

Chuẩn **Frobenius** là một cách đo “độ lớn” hay “khoảng cách” của một ma trận, tương tự như chuẩn Euclid cho vector. Nó rất phổ biến trong xử lý ma trận, đặc biệt khi muốn so sánh hai ma trận gần nhau cỡ nào (như trong bài toán xấp xỉ low-rank).

Với ma trận , chuẩn Frobenius được định nghĩa là:

Nói cách khác là chúng ta lấy bình phương từng phần tử trong ma trận để cộng lại và lấy căn bậc 2.

Ứng dụng:

* Đo khoảng cách giữa hai ma trận .
* Là hàm mất mát (loss) trong các bài toán hồi quy hoặc xấp xỉ ma trận.
* Dùng để đánh giá chất lượng của xấp xỉ low-rank: càng nhỏ thì ​ càng gần X.

Cho một ma trận , ta muốn tìm ma trận A sao cho:

Tức là: Tối thiểu hóa với điều kiện rank(A) = K

Gọi Truncated SVD của X là(Lấy K thành phần đầu tiên), nên đây cũng là nghiệm tốt nhất của bài toán:

**2.2 PCA**

PCA là bài toán đi tìm ma trận trực giao U và ma trận mô tả dữ liệu ở không gian thấp chiều là Z sao cho việc xấp xỉ sau đây là tốt nhất:

Với lần lượt là các ma trận tạo bởi K cột đầu tiên và D – K cột cuối cùng của ma trận trực giao U, và là vector kỳ vọng của dữ liệu.

Giả sử rằng vector kỳ vọng . Khi đó ta được:

Vậy nên bài toán tối ưu của PCA sẽ trở thành:

Với là ma trận đơn vị trong K chiều, điều kiện ràng buộc là các cột của tạo thành một hệ trực chuẩn.

Nghiệm cho bài toán này là:

: Gồm K vector riêng tương ứng với K giá trị riêng lớn nhất của ma trận hiệp phương sai .

Là tọa độ của dữ liệu trong hệ trục mới.

**2.3 Quan hệ giữa PCA và SVD**

PCA thực chất là một trường hợp đặc biệt của SVD. Chúng ta có thể dễ dàng nhận thấy.

Các vector chính trong PCA chính là K cột đầu tiên của ma trận U trong SVD của X.

Tọa độ chiếu . Hay có thể coi là tương đương với .

Như vậy, nếu các điểm dữ liệu được biễu diễn bởi các cột của một ma trận, và trung bình cộng của mỗi hàng của ma trận đó bằng 0 (để cho vector kỳ vọng bằng 0), thì nghiệm của bài toán PCA được rút ra trực tiếp từ Truncated SVD của ma trận đó. Nói cách khác, nghiệm của PCA chính là một trường hợp đặc biệt của bài toán Matrix Factorization giải bằng SVD.

3. Cách chọn chiều của dữ liệu mới

Cách xác định K là dựa trên lượng thông tin đã giữ lại. PCA là phương pháp tối đa tổng phương sai được giữ lại, nên ta có thể coi tổng các phương sai được giữ lại là lượng thông tin được giữ lại. Mà phương sai càng lớn thì dữ liệu có độ phân tán cao thể hiện lượng thông tin càng lớn.

Tổng phương sai bằng tổng các giá trị riêng của ma trận hiệp phương sai, PCA thì là phương pháp giữ lại lượng thông tin

Vậy lượng thông tin được giữ lại khi số chiều dữ liệu mới sau khi PCA là K:

Như vậy, giả sử ta muốn giữ lại 99% dữ liệu, ta chỉ cần chọn K là số tự nhiên nhỏ nhất sao cho

Khi dữ liệu phân bố quanh một không gian con, các giá trị phương sai lớn nhất ứng với các  đầu tiên lớn hơn nhiều so với các phương sai còn lại. Khi đó, ta có thể chọn được K khá nhỏ để đạt được

4. Các trường hợp PCA trong bài toán thực tế:

Để cho đơn giản thì ta sẽ coi dữ liệu đã được chuẩn hóa, tức là đã được trừ đi vector kỳ vọng. Khi đó ma trận hiệp phương sai S =

**4.1 Số chiều nhiều hơn số điểm dữ liệu**

Đây là trường hợp ma trận với D>N. Đây là dạng ma trận cao và hẹp.

Gọi:

S =

Đây là ma trận hiệp phương sai. Tuy nhiên trong trường hợp này ma trận S rất lớn, rank của S tối đa chỉ là N vì:

S chỉ có tối đa N trị riêng khác 0.

Giải pháp: Dựa vào các tính chất để giải quyết.

Tính chất 1: Giá trị riêng của A cũng là giá trị riêng của kA với k khác 0 bất kì.

TÍnh chất 2: Nếu thì:  
Giá trị riêng của AB và BA là giống nhau.

Áp dụng vào:

Thay vì tìm giá trị riêng của S, ta tìm giá trị riêng của T, trong trường hợp này T là ma trận nhỏ hơn rất nhiều

Tính chất 3: Chuyển Vector riêng từ T sang S

Giả sử:

Nhân cả 2 vế với X:

Như vậy:

Vẫn là giá trị riêng

là vector riêng của

Ta có thể xây dựng vector riêng của S từ vector riêng của T bằng cách:

4.2 Chuẩn hóa các vector riêng

Không gian riêng ứng với trị riêng của một ma trận là không gian sinh (span subspace) tạo bởi toàn bộ các vector riêng ứng với trị riêng đó.

Việc cuối cùng phải làm là chuẩn hoá các vector riêng tìm được sao cho chúng tạo thành một hệ trực chuẩn. Việc này có thể dựa trên hai tính chất:

Tính chất 1: Trực giao nhờ trị riêng khác nhau

Nếu:

+ A là ma trận đối xứng(VD ma trận hiệp phương sai )

+ là 2 giá trị riêng khác nhau của A thì:

Chứng minh:

Trong ma trận đối xứng, với mọi vector u và v, ta có:

Vì A đối xứng

Mà theo định lý phổ của ma trận đối xứng:

Mà do 2 giá trị riêng khác nhau nên .

Vậy các vector riêng ứng với giá trị riêng khác nhau là trực giao.

Tính chất 2:

Khi có nhiều vector riêng ứng với cùng một trị riêng (bị trùng trị riêng), thì chúng không tự động trực giao.

Nhưng ta vẫn có thể biến tập này thành một hệ trực chuẩn bằng Gram-Schmidt process.

Gram-Schmidt là một phương pháp biến một tập vector độc lập tuyến tính thành một hệ trực chuẩn (orthonormal basis).

Tóm lại để thu được hệ trực chuẩn , các bước ta cần làm như sau:

B1: Tính trị riêng và vector riêng của ma trận hiệp phương sai .

B2: Chọn K vector ứng với K trị riêng lớn nhất.

B3: Áp dụng 2 tính chất 1 và 2.

B4: Chuẩn hóa mỗi vector về độ dài 1.

B5: Tập các vector đó là ma trận trong PCA.

Phương pháp Gram-Schmit:

Cho một tập hợp các vector độc lập tuyến tính, thuật toán Gram-Schmidt tạo ra một tập hợp mới các vectơ trực giao (vuông góc với nhau) và có độ dài đơn vị (chuẩn hóa), đồng thời vẫn bao phủ cùng một không gian con.

Các bước chính của thuật toán Gram-Schmidt:  
Giả sử chúng ta có một tập hợp các vectơ độc lập tuyến tính ​. Thuật toán Gram-Schmidt xây dựng một tập hợp các vectơ trực giao ​ và sau đó chuẩn hóa chúng để tạo thành một tập hợp trực chuẩn như sau:

B1: Đặt Chuẩn hóa để có

B2: Tính = Chuẩn hóa để được .

B3: Tổng quát hơn, cho j từ 1 đến k, ta tính:

Chuẩn hóa

Trong công thức project: Trên tử là tích vô hướng của u và v, dưới mẫu là bình phương độ dài của vector u. u là vector mà v được chiếu lên.

4.3 Với các bài toán large-scale

rong rất nhiều bài toán, cả D và N đều là các số rất lớn, đồng nghĩa với việc ta phải tìm trị riêng cho một ma trận rất lớn. Ví dụ, có 1 triệu bức ảnh 1000 × 1000 pixel, như vậy  là một số rất lớn, việc trực tiếp tính toán trị riêng và vector riêng cho ma trận hiệp phương sai là không khả thi. Tuy nhiên, có một phương pháp cho phép tính xấp xỉ các giá trị này một cách nhanh hơn. Phương pháp đó là Power Method.

Phương pháp Power tìm giá trị riêng và vector riêng đầu tiên của ma trận nửa xác định dương :

1. Chọn một vector bất kỳ.

2. Với k = 1,2,… tính

3. Chuẩn hóa

4. Nếu đủ nhỏ thì dừng lại. Nếu không k:= k+1 rồi quay lại bước 2

5. là vector riêng ứng với giá trị riêng lớn nhất

Để tìm vector riêng và giá trị riêng của ma trận A ta có định lý:

Định lý: Nếu ma trận nửa xác định dương A có các giá trị riêng và các vector riêng tương ứng tạo thành hệ trực chuẩn, thì ma trận:

Có các giá trị riêng và có các vector riêng tương ứng là

Chứng minh:

Với i = 1:

Với i > 1:

do là hệ trực chuẩn. Do mọi ma trận nửa xác định dương trong không gian thực đều là ma trận đối xứng.

Lúc này là cặp giá trị riêng – vector riêng lớn nhất của B. Chúng ta lại thực hiện phương pháp Power để tìm 2 biến số này một lần nữa.

Power Method cơ bản chỉ tìm một giá trị riêng lớn nhất và vector riêng tương ứng.

Để tìm tiếp cặp thứ 2 ta cần loại bỏ ảnh hưởng của ra khỏi ma trận gốc A. Khi đó ta mới dùng lại Power Method trên ma trận mới:

Sau đó tiếp tục lặp lại để tìm một dãy giá trị riêng với vector riêng.

Tài liệu tham khảo:  
1. https://machinelearningcoban.com/2017/06/15/pca/