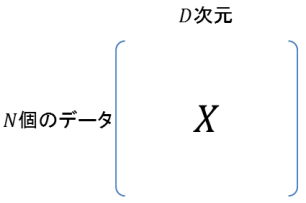
# PCA

のデータ行列は、各行が各データに対応しているものとする。各データは次元で、全部で個のデータがあるわけだ。



まず、各次元において、平均値を計算し、それを引く。まぁ、normalizationみたいな感じかな。

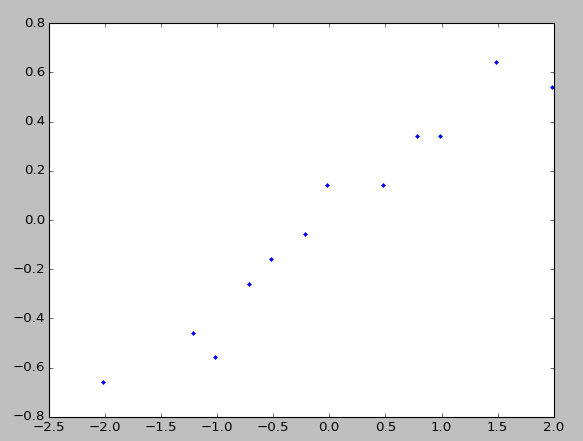
標準偏差で割るというやり方もあるみたい（？）だが、よく分からん。このの共分散行列は、

だ。この時、PCAは、この共分散行列のSVDで計算できる。つまり、

と表される時、共分散行列の固有値は、固有ベクトルはだ。そして、この固有ベクトルが、PCAにおける主成分ベクトルに対応する。

## 固有ベクトルの解釈

例えば、以下のような2次元データ（normalize済み）がある時、固有ベクトルは、ととなる。最初のベクトルは、下図の右肩上がりの傾きに相当し、まさに第一主成分であることが分かる。一方、2つ目のベクトルは、それと直交するベクトルで、まさに第二主成分だ。



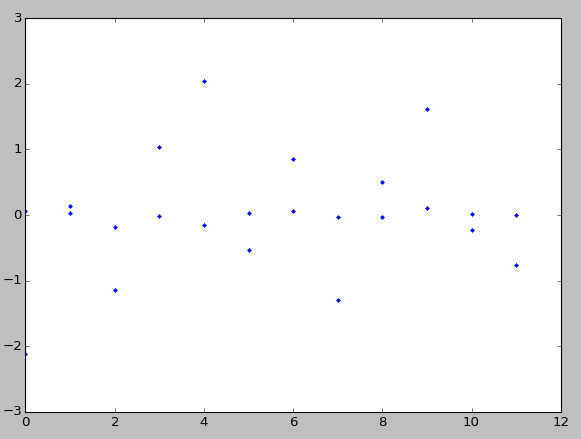
## 固有値の解釈

固有値は、各固有ベクトル（つまり、主成分）の寄与率に相当する。上のデータの例では、固有値は、1.440と0.008となる。つまり、第一主成分の寄与率は、

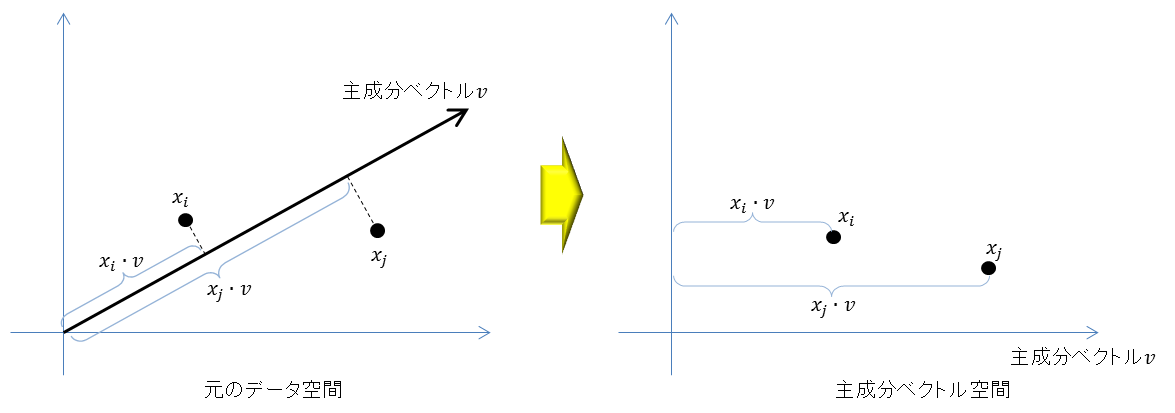
となり、第一主成分が、データのほぼ99%を表していると解釈できる。

## 主成分空間への写像

元のデータを、主成分ベクトルの空間へ写像した結果って、よくPCAの結果として使われるよね。例えば、上のデータの例では、下図のような結果となる。



この図は、主成分ベクトルの行列を使い、により計算される。ただし、は、各列が主成分ベクトルとなっている。これは、各データについて、主成分ベクトルとの内積を計算することで、主成分方向の長さを計算しているに過ぎない。



なお、PCAの結果を二次元で表現する場合は、の最初の2列を使って表示すれば良いだけ。

# プログラム例

## OpenCV

OpenCVのPCAクラスだと、一発でPCAを実行してくれる。

cv::PCA pca(X, cv::Mat(), CV\_PCA\_DATA\_AS\_ROW);

この時、固有値は、pca.eigenvaluesに行列として格納され、固有ベクトルは、pca.eigenvectorsに行列として格納される。例えば、１つ目の固有ベクトルは、pca.eigenvectors.row(0)で取得できる。

また、主成分ベクトル空間への写像は、pca.project()で計算できる。

## Matlab

Matlabだと、SVDを使って計算することになる。まず、から平均を引いてnormalizeする。

X=X-repmat(mean(X),length(X),1)

次に、SVDにより固有値、固有ベクトルを計算する。

[U,S,V]=svd(X’\*X/length(X))

この時、固有値は、行列Sの対角成分だ。

S =

1.440 0

0 0.008

また、固有ベクトルは、行列Vの各列だ。**※各行ではないので注意しろ！**

V =

-0.94 -0.33

-0.33 0.44

主成分ベクトル空間への写像は、で計算できる。この結果の各行が、各データの写像だ。

>> X\*V