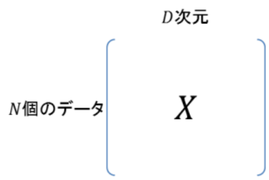
PCA

$N \times D$のデータ行列X^*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D次元で、全部でN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。

$$X = X^* - \bar{X}$$

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Xの共分散行列は、

$$\frac{1}{N} X^T X$$

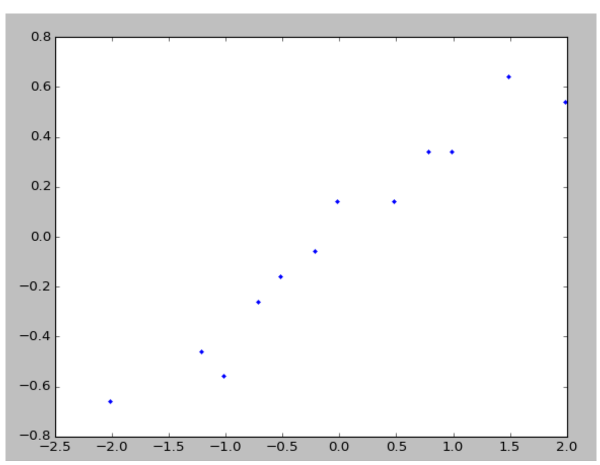
だ。この時、PCA は、この共分散行列の SVD で計算できる。つまり、

$$\frac{1}{N} X^T X = U \Sigma V$$

と表される時、共分散行列の固有値はΣ、固有ベクトルはV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、$[0.94, 0.33]$と$[-0.33, 0.94]$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

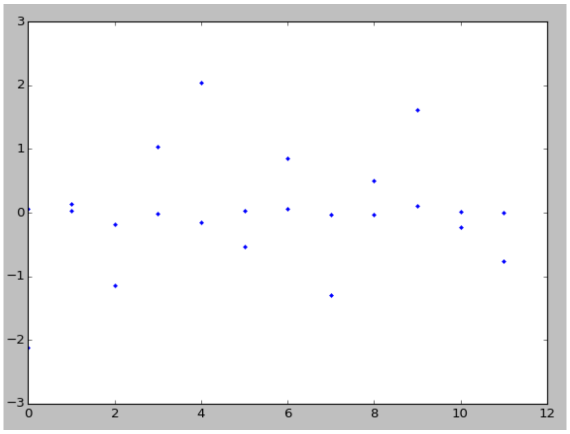
固有値は、各固有ベクトル（つまり、主成分）の寄与率に相当する。上のデータの例では、固有値は、1.440 と 0.008 となる。つまり、第一主成分の寄与率は、

$$\frac{1.440}{1.440 + 0.008} \approx 0.99$$

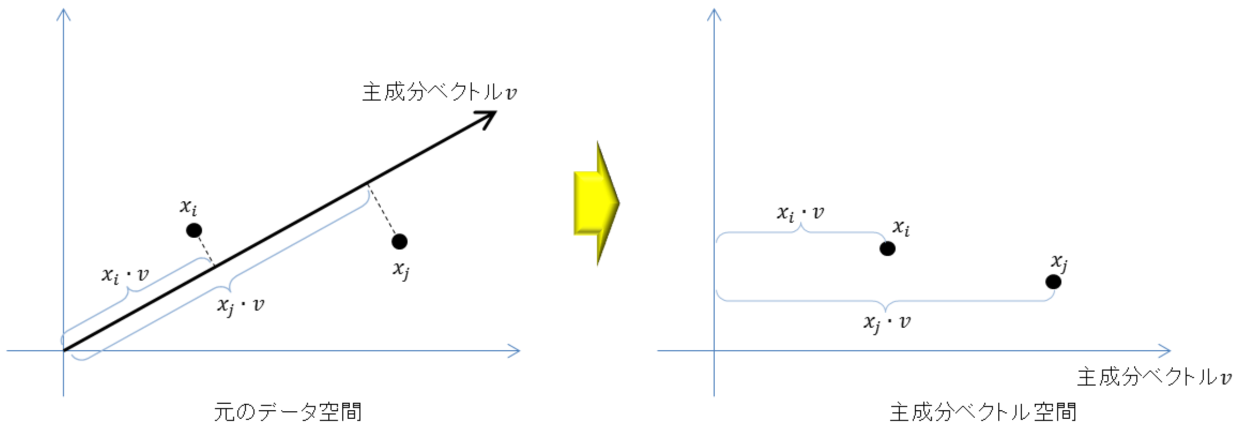
となり、第一主成分が、データのほぼ 99%を表していると解釈できる。

主成分空間への写像

元のデータX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Vを使い、XVにより計算される。ただし、V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、XV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$N \times 1$行列として格納され、固有ベクトルは、`pca.eigenvectors` に$N \times D$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、X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、XVで計算できる。この結果の各行が、各データの写像だ。

```
>> X*V
```