主元分析(PCA)理论分析及应用

什么是 PCA?

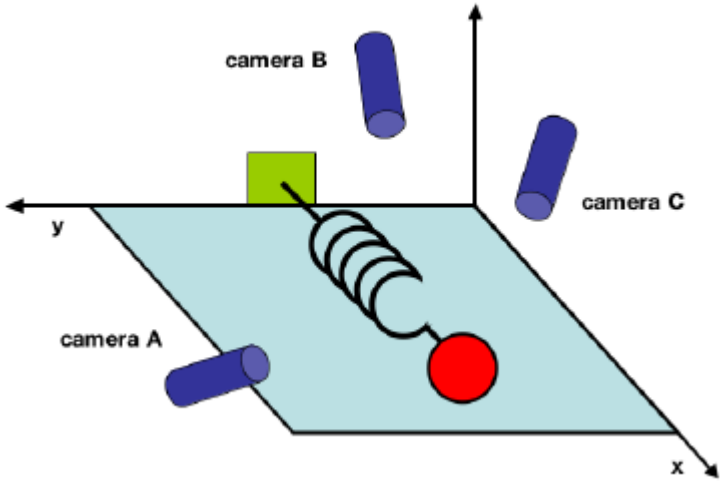
PCA 是 Principal component analysis 的缩写，中文翻译为主元分析。它是一种对数据进行分析的技术，最重要的应用是对原有数据进行简化。正如它的名字：主元分析，这种方法可以有效的找出数据中最“主要”的元素和结构，去除噪音和冗余，将原有的复杂数据降维，揭示隐藏在复杂数据背后的简单结构。它的优点是简单，而且无参数限制，可以方便的应用与各个场合。因此应用极其广泛，从神经科学到计算机图形学都有它的用武之地。被誉为应用线形代数最价值的结果之一。

在以下的章节中，不仅有对 PCA 的比较直观的解释，同时也配有较为深入的分析。首先将从一个简单的例子开始说明 PCA 应用的场合以及想法的由来，进行一个比较直观的解释；然后加入数学的严格推导，引入线形代数，进行问题的求解。随后将揭示 PCA 与 SVD(Singular Value Decomposition)之间的联系以及如何将之应用于真实世界。最后将分析 PCA 理论模型的假设条件以及针对这些条件可能进行的改进。

一个简单的模型

在实验科学中常遇到的情况是，使用大量的变量代表可能变化的因素，例如光谱、电压、速度等等。但是由于实验环境和观测手段的限制，实验数据往往变得极其的复杂、混乱和冗余的。如何对数据进行分析，取得隐藏在数据背后的变量关系，是一个很困难的问题。在神经科学、气象学、海洋学等等学科实验中，假设的变量个数可能非常之多，但是真正的影响因素以及它们之间的关系可能又是非常之简单的。

下面的模型取自一个物理学中的实验。它看上去比较简单，但足以说明问题。如图表1所示。这是一个理想弹簧运动规律的测定实验。假设球是连接在一个无质量无摩擦的弹簧之上，从平衡位置沿 x 轴拉开一定的距离然后释放。



图表 1

对于一个具有先验知识的实验者来说，这个实验是非常容易的。球的运动只是在 x 轴向上发生，只需要记录下 x 轴向上的运动序列并加以分析即可。但是，在真实世界中，对于第一次实验的探索者来说（这也是实验科学中最常遇到的一种情况），是不可能进行这样的假设的。那么，一般来说，必须记录下球的三维位置 (x_0, y_0, z_0)。这一点可以通过在不同角度放置三个摄像机实现（如图所示），假设以 $200Hz$ 的频率拍摄画面，就可以得到球在空间中的运动序列。但是，由于实验的限制，这三台摄像机的角度可能比较任意，并不是正交的。事实上，在真实世界中也并没有所谓的 $\{x, y, z\}$ 轴，每个摄像机记录下的都是一幅二维的图像，有其自己的空间坐标系，球的空间位置是由一组二维坐标记录的：$[(x_A, y_A), (x_B, y_B), (x_C, y_C)]$。经过实验，系统产生了几分钟内球的位置序列。怎样从这些数据中得到球是沿着某个轴运动的规律呢？怎样将实验数据中的冗余变量剔除，化归到这个潜在的轴上呢？

这是一个真实的实验场景，数据的噪音是必须面对的因素。在这个实验中噪音可能来自空气、摩擦、摄像机的误差以及非理想化的弹簧等等。噪音使数据变得混乱，掩盖了变量间的真实关系。如何去除噪音是实验者每天所要面对的巨大考验。

上面提出的两个问题就是 PCA 方法的目标。PCA 主元分析方法是解决此类问题的一个有力的武器。下文将结合以上的例子提出解决方案，逐步叙述 PCA 方法的思想和求解过程。

线形代数：基变换

从线形代数的角度来看，PCA 的目标就是使用另一组基去重新描述得到的数据空间。而新的基要能尽

量揭示原有的数据间的关系。在这个例子中，沿着某 x 轴上的运动是最重要的。这个维度即最重要的“主元”。PCA 的目标就是找到这样的“主元”，最大程度的去除冗余和噪音的干扰。

A. 标准正交基

为了引入推导，需要将上文的数据进行明确的定义。在上面描述的实验过程中，在每一个采样时间点上，每个摄像机记录了一组二维坐标 (x_A, y_A)，综合三台摄像机数据，在每一个时间点上得到的位置数据对应于一个六维列向量。

$$\vec{X} = \begin{pmatrix} x_A \\ y_A \\ x_B \\ y_B \\ x_C \\ y_C \end{pmatrix}$$

如果以 $200Hz$ 的频率拍摄10分钟，将得到 $10 \times 60 \times 200 = 120000$ 个这样的向量数据。

抽象一点来说，每一个采样点数据 $\vec{X}$ 都是在 m 维向量空间（此例中 $m = 6$）内的一个向量，这里的 m 是牵涉的变量个数。由线形代数我们知道，在 m 维向量空间中的每一个向量都是一组正交基的线形组合。最普通的一组正交基是标准正交基，实验采样的结果通常可以看作是在标准正交基下表示的。举例来说，上例中每个摄像机记录的数据坐标为 (x_A, y_A)，这样的基便是 $\{(1,0),(0,1)\}$。那为什么不取 $\{(\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2}),(\frac{-\sqrt{2}}{2},\frac{-\sqrt{2}}{2})\}$ 或是其他任意的基呢？原因是，这样的标准正交基反映了数据的采集方式。假设采集数据点是 $(2,2)$，一般并不会记录 $(2\sqrt{2},0)$（在 $\{(\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2}),(\frac{-\sqrt{2}}{2},\frac{-\sqrt{2}}{2})\}$ 基下），因为一般的观测者都是习惯于取摄像机的屏幕坐标，即向上和向右的方向作为观测的基准。也就是说，标准正交基表现了数据观测的一般方式。

在线形代数中，这组基表示为行列向量线形无关的单位矩阵。

$$B=\begin{vmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{vmatrix}=\begin{vmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{vmatrix}$$

B. 基变换

从更严格的数学定义上来说，PCA 回答的问题是：如何寻找到另一组正交基，它们是标准正交基的线性组合，而且能够最好的表示数据集？

这里提出了 PCA 方法的一个最关键的假设：线性。这是一个非常强的假设条件。它使问题得到了很大程度的简化：1）数据被限制在一个向量空间中，能被一组基表示；2）隐含的假设了数据之间的连续性关系。

这样一来数据就可以被表示为各种基的线性组合。令 X 表示原数据集。X 是一个 $m\times n$ 的矩阵，它的每一个列向量都表示一个时间采样点上的数据 $\vec{X}$，在上面的例子中，$m=6,n=120000$。Y 表示转换以后的新的数据集表示。P 是他们之间的线性转换。

$$PX=Y \tag{1}$$

有如下定义：

- p_i 表示 P 的行向量。
- x_i 表示 X 的列向量（或者 $\vec{X}$）。
- y_i 表示 Y 的列向量。

公式(1)表示不同基之间的转换，在线性代数中，它有如下的含义：

- P 是从 X 到 Y 的转换矩阵。（ 空间转换）

- 几何上来说，P 对 X 进行旋转和拉伸得到 Y。

- P 的行向量，$\{p_1, p_2, ..., p_m\}$ 是一组新的基，而 Y 是原数据在这组新的基表示下得到的重新表示。

下面是对最后一个含义的显式说明：

$$PX = \begin{vmatrix} p_1 \\ \vdots \\ p_m \end{vmatrix} \begin{vmatrix} x_1 & \cdots & x_n \end{vmatrix}$$

注意到 Y 的列向量：

$$y_i = \begin{vmatrix} p_1 x_i \\ \vdots \\ p_m x_i \end{vmatrix}$$

可见 y_i 表示的是 x_i 与 P 中对应列的点积，也就是相当于是在对应向量上的投影（点积就是投影）。所以，P 的行向量事实上就是一组新的基。它对原数据 X 进行重新表示。在一些文献中，将数据 X 成为“源”，而将变换后的 Y 称为“信号”。这是由于变换后的数据更能体现信号成分的原因。

C. 问题

在线性的假设条件下，问题转化为寻找一组变换后的基，也就是 P 的行向量 $\{p_1, ..., p_m\}$，这些向量就是 PCA 中所谓的“主元”。问题转化为如下的形式：

- 怎样才能最好的表示原数据 X？
- P 的基怎样选择才是最好的？

解决问题的关键是如何体现数据的特征。那么，什么是数据的特征，如何体现呢？

方差和目标

“最好的表示”是什么意思呢？下面的章节将给出一个较为直观的解释，并增加一些额外的假设条件。在线性系统中，所谓的“混乱数据”通常包含以下的三种成分：噪音、旋转以及冗余。下面将对这三

种成分做出数学上的描述并针对目标作出分析。

A. 噪音和旋转

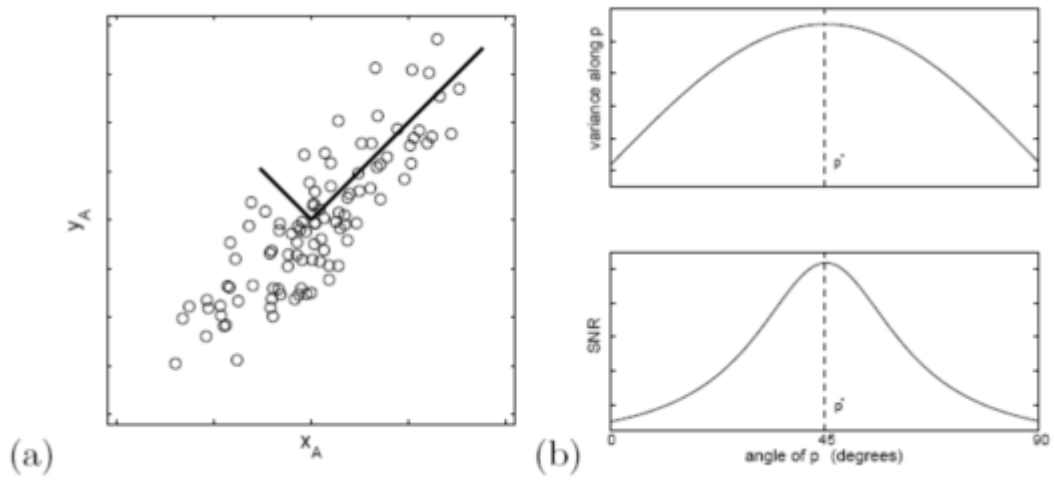
噪音对数据的影响是巨大的，如果不能对噪音进行区分，就不可能抽取数据中有用的信息。噪音的衡量有多种方式，最常见的定义是信噪比(*signal-to-noise ratio*)，或是方差比 σ^2：

$$SNR = \frac{\sigma^2_{signal}}{\sigma^2_{noise}} \tag{2}$$

比较大的信噪比表示数据的准确度高，而信噪比低则说明数据中的噪音成分比较多。那么怎样区分什么是信号，什么是噪音呢？这里假设，变化较大的信息被认为是信号，变化较小的则是噪音。事实上，这个标准等价于一个低通的滤波器，是一种标准的去噪准则。而变化的大小则是由方差来描述的。

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

它表示了采样点在平均值两侧的分布，对应于图表2(a)就是采样点云的“胖瘦”。显然的，方差较大，也就是较“宽”较“胖”的分布，表示了采样点的主要分布趋势，是主信号或主要分量；而方差较小的分布则被认为是噪音或次要分量。



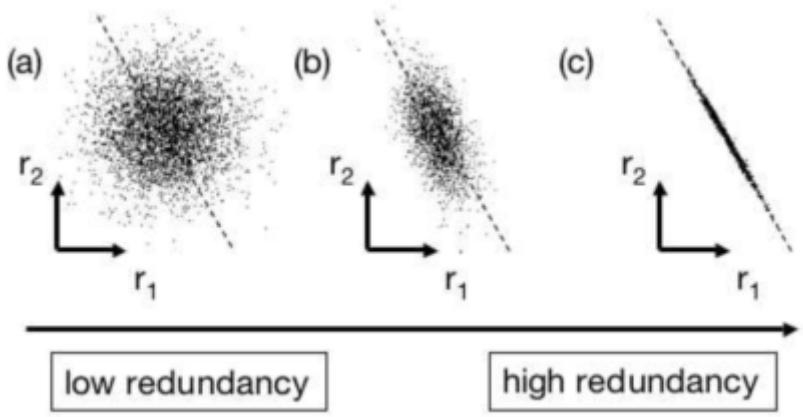
图表 2：(a)摄像机 A 的采集数据。图中黑色垂直直线表示一组正交基的方向。σ^2_{signal} 是采样点云在长线方向上分布的方差，而 σ^2_{noise} 是数据点在短线方向上分布的方差。(b)对 P 的基向量进行旋转使 SNR 和方差最

大。

假设摄像机 A 拍摄到的数据如图表2(a)所示，圆圈代表采样点，因为运动理论上是只存在于一条直线上，所以偏离直线的分布都属于噪音。此时描述的就是采样点云在某对垂直方向上的概率分布的比值。那么，最大限度的揭示原数据的结构和关系，找出某条潜在的，最优的 x 轴，事实上等价寻找一对空间内的垂直直线（图中黑线表示，也对应于此空间的一组基），使得信噪比尽可能大的方向。容易看出，本例中潜在的 x 轴就是图上的较长黑线方向。那么怎样寻找这样一组方向呢？直接的想法是对基向量进行旋转。如图表2(b)所示，随着这对直线的转动以及方差的变化情况。应于最大值的一组基，就是最优的“主元”方向。在进行数学中求取这组基的推导之前，先介绍另一个影响因素。

B. 冗余

有时在实验中引入了一些不必要的变量。可能会两种情况：1）该变量对结果没有影响；2）该变量可以用其它变量表示，从而造成数据冗余。下面对这样的冗余情况进行分析和分类。



图表 3：可能冗余数据的频谱图表示。r_1 和 r_2 分别是两个不同的观测变量。

（比如例子中的 x_A， x_B）。最佳拟合线用 $r_2 = kr_1$ 虚线表示。

如图表3所示，它揭示了两个观测变量之间的关系。(a)图所示的情况是低冗余的，从统计学上说，这两个观测变量是相互独立的，它们之间的信息没有冗余。而相反的极端情况如(c)，和高度相关，完全可以用表示。一般来说，这种情况发生可能是因为摄像机 A 和摄像机 B 放置的位置太近或是数据被重复记录了，也可能是由于实验设计的不合理所造成的。那么对于观测者而言，这个变量的观测数据就是完全冗余的，应当去除，只用一个变量就可以表示了。这也就是 PCA 中“降维”思想的本源。

C. 协方差矩阵

对于上面的简单情况，可以通过简单的线性拟合的方法来判断各观测变量之间是否出现冗余的情况，而对于复杂的情况，需要借助协方差来进行衡量和判断：

$$\sigma_{AB}^2 = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{n-1}$$

A， B 分别表示不同的观测变量所记录的一组值，在统计学中，由协方差的性质可以得到：（六个观测变量，每个观测变量都对应的有一组值的变化）

- $\sigma_{AB}^2 \geq 0$，且 $\sigma_{AB}^2 = 0$ 当且仅当观测变量 A， B 相互独立。
- $\sigma_{AB}^2 = \sigma_A^2$，当 $A = B$。

等价的，将 A， B 写成行向量的形式：

$$A = [a_1 \quad a_2 \quad \cdots \quad a_n], B = [b_1 \quad b_2 \quad \cdots \quad b_n]$$

协方差可以表示为：

$$\sigma_{AB}^2 \equiv \frac{1}{n-1} AB^T \tag{3}$$

那么，对于一组具有 m 个观测变量， n 个采样时间点的采样数据 X ，将每个观测变量的值写为行向量，可以得到一个的矩阵：

$$X = \begin{vmatrix} x_1 \\ \vdots \\ x_m \end{vmatrix} \tag{4}$$

接下来定义协方差矩阵如下：

$$C_X \equiv \frac{1}{n-1} XX^T \tag{5}$$

容易发现协方差矩阵性质如下：

- C_X 是一个 $m \times m$ 的平方对称矩阵。

- C_X 对角线上的元素是对应的观测变量的方差。

- 非对角线上的元素是对应的观测变量之间的协方差。

$$C_X \equiv \begin{vmatrix} \sigma^2_{x_1x_1} & \sigma^2_{x_1x_2} & \cdots & \sigma^2_{x_1x_m} \\ \sigma^2_{x_2x_1} & \sigma^2_{x_2x_2} & \cdots & \sigma^2_{x_2x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^2_{x_mx_1} & \sigma^2_{x_mx_2} & \cdots & \sigma^2_{x_mx_m} \end{vmatrix}$$

协方差矩阵 C_X 包含了所有观测变量之间的相关性度量。更重要的是，根据前两节的说明，这些相关性度量反映了数据的噪音和冗余的程度。

- 在对角线上的元素越大，表明信号越强，变量的重要性越高；元素越小则表明可能是存在的噪音或是次要变量。

- 在非对角线上的元素大小则对应于相关观测变量对之间冗余程度的大小。

一般情况下，初始数据的协方差矩阵总是不太好的，表现为信噪比不高且变量间相关度大。PCA 的目标就是通过基变换对协方差矩阵进行优化，找到相关“主元”。那么，如何进行优化？矩阵的那些性质是需要注意的呢？

D. 协方差矩阵的对角化

总结上面的章节，主元分析以及协方差矩阵优化的原则是：1）最小化变量冗余，对应于协方差矩阵的非对角元素要尽量小；2）最大化信号，对应于要使协方差矩阵的对角线上的元素尽可能的大。因为协方差矩阵的每一项都是正值，最小值为0，所以优化的目标矩阵的非对角元素应该都是0，对应于冗余最小。所以优化的目标矩阵 C_Y 应该是一个对角阵。即只有对角线上的元素可能是非零值。同时，PCA 假设 P 所对应的一组变换基 $\{p_1,...,p_m\}$ 必须是标准正交的，而优化矩阵对角线上的元素越大，就说明信号的成分越大，换句话就是对应于越重要的“主元”。

对于协方差矩阵进行对角化的方法很多。根据上面的分析，最简单最直接的算法就是在多维空间内进行搜索。和图表2(a)的例子中旋转 P 的方法类似：

1） 在 m 维空间中进行遍历，找到一个方差最大的向量，令作 p_1。

2） 在与垂直的向量空间中进行遍历，找出次大的方差对应的向量，记作 p_2。

3） 对以上过程循环，直到找出全部 m 的向量。它们生成的顺序也就是“主元”的排序。

这个理论上成立的算法说明了 PCA 的主要思想和过程。在这中间，牵涉到两个重要的特性：a)转换基是一组标准正交基。这给 PCA 的求解带来了很大的好处，它可以运用线性代数的相关理论进行快速有效的分解。这些方法将在后面提到。b）在 PCA 的过程中，可以同时得到新的基向量所对应的“主元排序”，利用这个重要性排序可以方便的对数据进行光顺、简化处理或是压缩。

A. PCA 的假设和局限

PCA 的模型中存在诸多的假设条件，决定了它存在一定的限制，在有些场合可能会造成效果不好甚至失效。对于学习和掌握 PCA 来说，理解这些内容是非常重要的，同时也有利于理解基于改进这些限制条件的 PCA 的一些扩展技术。

PCA 的假设条件包括：

1. 线形性假设。

如同文章开始的例子，PCA 的内部模型是线性的。这也就决定了它能进行的主元分析之间的关系也是线性的。现在比较流行的 ***kernel*-PCA** 的一类方法就是使用非线性的权值对原有 PCA 技术的拓展。

2. 使用中值和方差进行充分统计。

使用中值和方差进行充分的概率分布描述的模型只限于指数型概率分布模型。（例如高斯分布），也就是说，如果我们考察的数据的概率分布并不满足高斯分布或是指数型的概率分布，那么 PCA 将会失效。在这种模型下，不能使用方差和协方差来很好的描述噪音和冗余，对教化之后的协方差矩阵并不能得到很合适的结果。

事实上，去除冗余的最基础的方程是：

$$p(y_1, y_2) = p(y_1)p(y_2)$$

其中 $p(\bullet)$ 代表概率分布的密度函数。基于这个方程进行冗余去除的方法被称作独立主元分析(ICA)方法(*Independent Component Analysis*)。不过，所幸的是，根据中央极限定理，现实生活中所遇到的大部分采样数据的概率分布都是遵从高斯分布的。所以 PCA 仍然是一个使用于绝大部分领域的稳定且有效的算法。

3. 大方差向量具有较大重要性。

PCA 方法隐含了这样的假设：数据本身具有较高的信噪比，所以具有最高方差的一维向量就可以被看作是主元，而方差较小的变化则被认为是噪音。这是由于低通滤波器的选择决定的。

4. 主元正交。

PCA 方法假设主元向量之间都是正交的，从而可以利用线形代数的一系列有效的数学工具进行求解，大大提高了效率和应用的范围。

PCA 求解：特征根分解

在线形代数中，PCA 问题可以描述成以下形式：

寻找一组正交基组成的矩阵 P，有 $Y = PX$，使得 $C_Y \equiv \frac{1}{n-1}YY^T$ 是对角阵。则 P 的行向量（也就是一组正交基），就是数据 X 的主元向量。

对 C_Y 进行推导：

$$\begin{aligned} C_Y &= \frac{1}{n-1}YY^T \\ &= \frac{1}{n-1}(PX)(PX)^T \\ &= \frac{1}{n-1}PXX^TP^T \\ &= \frac{1}{n-1}P(XX^T)P^T \end{aligned}$$

$$C_Y = \frac{1}{n-1} PAP^T$$

定义 $A = XX^T$，则 A 是一个对称阵。对 A 进行对角化求取特征向量得：

$$A = EDE^T$$

则 D 是一个对角阵而 E 则是对称阵 A 的特征向量排成的矩阵。

这里要提出的一点是，A 是一个 $m \times m$ 的矩阵，而它将有 $r(r \le m)$ 个特征向量。其中 r 是 A 矩阵的秩。如果 $r < m$，则 A 即为退化阵。此时分解出的特征向量不能覆盖整个 m 空间。此时只需要在保证基的正交性的前提下，在剩余的空间中任意取得 $m-r$ 维正交向量填充 E 的空格即可。它们将不对结果造成影响。因为此时对应于这些特征向量的特征值，也就是方差值为零。

求出特征向量矩阵后我们取 $P \equiv E^T$，则 $A = P^T DP$，由线形代数可知矩阵 P 有性质 $P^{-1} = P^T$，从而进行如下计算：

$$\begin{aligned} C_Y &= \frac{1}{n-1} PAP^T \\ &= \frac{1}{n-1} P(P^T DP)P^T \\ &= \frac{1}{n-1} (PP^T)D(PP^T) \\ &= \frac{1}{n-1} (PP^{-1})D(PP^{-1}) \end{aligned}$$

$$C_Y = \frac{1}{n-1} D$$

可知此时的 P 就是我们需要求得变换基。至此我们可以得到 PCA 的结果：

- X 的主元即是 XX^T 的特征向量，也就是矩阵 P 的行向量。
- 矩阵 C_Y 对角线上第 i 个元素是数据 X 在方向 p_i 的方差。

我们可以得到 PCA 求解的一般步骤：

1）采集数据形成 $m \times n$ 的矩阵。m 为观测变量个数，n 为采样点个数。

2）在每个观测变量（矩阵行向量）上减去该观测变量的平均值得到矩阵 X 。

3）对 XX^T 进行特征分解，求取特征向量以及所对应的特征根。

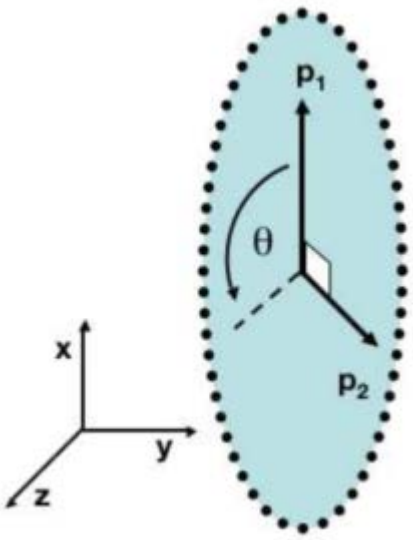
总结和讨论

- PCA 技术的一大好处是对数据进行降维的处理。我们可以对新求出的“主元”向量的重要性进行排序，根据需要取前面最重要的部分，将后面的维数省去，可以达到降维从而简化模型或是对数据进行压缩的效果。同时最大程度的保持了原有数据的信息。

  在前文的例子中，经过 PCA 处理后的数据只剩下了一维，也就是弹簧运动的那一维，从而去除了冗余的变量，揭示了实验数据背后的物理原理。

- PCA 技术的一个很大的优点是，它是完全无参数限制的。在 PCA 的计算过程中完全不需要人为的设定参数或是根据任何经验模型对计算进行干预，最后的结果只与数据相关，与用户是独立的。

  但是，这一点同时也可以看作是缺点。如果用户对观测对象有一定的先验知识，掌握了数据的一些特征，却无法通过参数化等方法对处理过程进行干预，可能会得不到预期的效果，效率也不高。



图表 4：黑色点表示采样数据，排列成转盘的形状。

该数据的主元是 (P_1, P_2) 或是旋转角 θ。

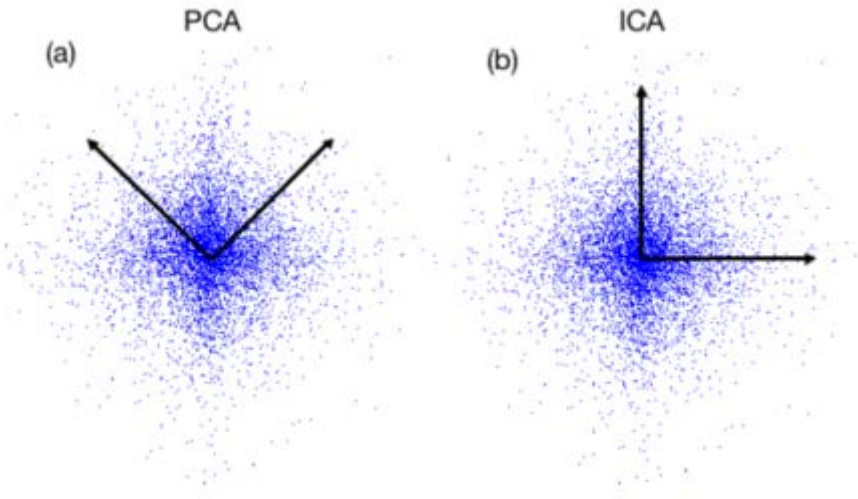
如图表 4中的例子，PCA 找出的主元将是 (P_1, P_2) 。但是这显然不是最优和最简化的主元。(P_1, P_2) 之间存在着非线性的关系。根据先验的知识可知旋转角 θ 是最优的主元。则在这种情况下，PCA 就会失

效。但是，如果加入先验的知识，对数据进行某种划归，就可以将数据转化为以θ为线性的空间中。这类根据先验知识对数据预先进行非线性转换的方法就成为 *kernel*-PCA，它扩展了 PCA 能够处理的问题的范围，又可以结合一些先验约束，是比较流行的方法。

- 有时数据的分布并不是满足高斯分布。如图表5所示，在非高斯分布的情况下，PCA 方法得出的主元可能并不是最优的。在寻找主元时不能将方差作为衡量重要性的标准。要根据数据的分布情况选择合适的描述完全分布的变量，然后根据概率分布式

$$P(y_1, y_2) = P(y_1)P(y_2)$$

来计算两个向量上数据分布的相关性。等价的，保持主元间的正交假设，寻找的主元同样要使 $P(y_1, y_2) = 0$。这一类方法被称为独立主元分解(ICA)。



图表 5：数据的分布并不满足高斯分布，呈明显的十字星状。
这种情况下，方差最大的方向并不是最优主元方向。

- PCA 方法和线形代数中的奇异值分解(SVD)方法有内在的联系，一定意义上来说，PCA 的解法是 SVD 的一种变形和弱化。对于$m \times n$的矩阵X，通过奇异值分解可以直接得到如下形式：

$$X = U\Sigma V^T$$

其中U是一个$m \times m$的矩阵，V是一个$n \times n$的矩阵，而Σ是$m \times n$的对角阵。形式如下：

$$\Sigma = \begin{vmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{vmatrix}$$

其中$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$，是原矩阵的奇异值。由简单推导可知，如果对奇异值分解加以约束：U 的向量必须正交，则矩阵U 即为 PCA 的特征值分解中的E，则说明 PCA 并不一定需要求取 XX^T，也可以直接对原数据矩阵 X 进行 SVD 奇异值分解即可得到特征向量矩阵，也就是主元向量。

计算机视学领域的应用

PCA 方法是一个具有很高普适性的方法，被广泛应用于多个领域。这里要特别介绍的是它在计算机视觉领域的应用，包括如何对图像进行处理以及在人脸识别方面的特别作用。

A. 数据表示

如果要将 PCA 方法应用于视觉领域，最基本的问题就是图像的表达。如果是一幅 $N \times N$ 大小的图像，它的数据将被表达为一个 N^2 维的向量：

$$X = (x_1 \quad x_2 \quad \ldots \quad x_{N^2})^T$$

在这里图像的结构将被打乱，每一个像素点被看作是一维，最直接的方法就是将图像的像素一行行的头尾相接成一个一维向量。还必须要注意的是，每一维上的数据对应于对应像素的亮度、灰度或是色彩值，但是需要划归到同一纬度上。

B. 模式识别

假设数据源是一系列的20幅图像，每幅图像都是 $N \times N$ 大小，那么它们都可以表示为一个 N^2 维的向量。将它们排成一个矩阵：

$$\text{ImagesMatrix} = (\text{ImageVec1} \quad \text{ImageVec2} \quad \ldots \quad \text{ImageVec20})$$

然后对它们进行 PCA 处理，找出主元。

为什么这样做呢？据人脸识别的例子来说，数据源是20幅不同的人脸图像，PCA 方法的实质是寻找这些图像中的相似的维度，因为人脸的结构有极大的相似性（特别是同一个人的人脸图像），则使用 PCA 方法就可以很容易的提取出人脸的内在结构，也及时所谓“模式”，如果有新的图像需要与原有图像比较，就可以在变换后的主元维度上进行比较，则可衡量新图与原有数据集的相似度如何。

对这样的一组人脸图像进行处理，提取其中最重要的主元，即可大致描述人脸的结构信息，称作“特征脸”(*EigenFace*)。这就是人脸识别中的重要方法“特征脸方法”的理论根据。近些年来，基于对一般 PCA 方法的改进，结合 ICA、*kernel*-PCA 等方法，在主元分析中加入关于人脸图像的先验知识，则能得到更好的效果。

C. 图像信息压缩

使用 PCA 方法进行图像压缩，又被称为 *Hotelling* 算法，或者 *Karhunenand Leove(KL)*变换。这是视觉领域内图像处理的经典算法之一。具体算法与上述过程相同，使用 PCA 方法处理一个图像序列，提取其中的主元。然后根据主元的排序去除其中次要的分量，然后变换回原空间，则图像序列因为维数降低得到很大的压缩。例如上例中取出次要的5个维度，则图像就被压缩了1/4。但是这种有损的压缩方法同时又保持了其中最“重要”的信息，是一种非常重要且有效的算法。

参考文献

[1] Lindsay I Smith. (2002) “A tutorial on Principal Components Analysis” http://csnet.otago.ac.nz/cosc453/student_ tutorials/principal_components.pdf

[2] Jonathon Shlens. (2005) “A Tutorial on Principal Component Analysis” http://www.snl.salk.edu/~shlens/pub/notes/pca.pdf

[3] Will, Todd (1999) “Introduction to the Singular Value Decomposition” Davidson College. http://www.davidson.edu/academic/math/will/svd/index.html

[4] Bell, Anthony and Sejnowski, Terry. (1997) “The Independent Components of Natural Scenes are EdgeFilters.” Vision Research 37(23), 3327-3338.

[5] T.F. Cootes and C.J.Taylor (2004) “Statistical Models of Appearance for Computer Vision” http://www.isbe.man.ac.uk/~bim/Models/app_models.pdf

[6] 张翠平 苏光大 (2000)“人脸识别技术综述”《中国图像图形学报》第五卷 A 版第11期

[7] 何国辉 甘俊英 (2006)“PCA 类内平均脸法在人脸识别中的应用研究”《计算机应用研究》2006年第三期

[8] 牛丽平 付仲良 魏文利 (2006)“人脸识别技术研究”《电脑开发与应用》2006年第五期

[9] Wikipedia “principal components analysis”词条解释 From Answers.com

补充

主成分分析（Principal components analysis）-最大方差解释

在这一篇之前的内容是《Factor Analysis》，由于非常理论，打算学完整个课程后再写。在写这篇之前，我阅读了 PCA、SVD 和 LDA。这几个模型相近，却都有自己的特点。本篇打算先介绍 PCA，至于他们之间的关系，只能是边学边体会了。PCA 以前也叫做 Principal factor analysis。

1. 问题

真实的训练数据总是存在各种各样的问题：

1、比如拿到一个汽车的样本，里面既有以“千米/每小时”度量的最大速度特征，也有“英里/小时”的最大速度特征，显然这两个特征有一个多余。

2、拿到一个数学系的本科生期末考试成绩单，里面有三列，一列是对数学的兴趣程度，一列是复习时间，还有一列是考试成绩。我们知道要学好数学，需要有浓厚的兴趣，所以第二项与第一项强相关，第三项和第二项也是强相关。那是不是可以合并第一项和第二项呢？

3、拿到一个样本，特征非常多，而样例特别少，这样用回归去直接拟合非常困难，容易过度拟合。比如北京的房价：假设房子的特征是（大小、位置、朝向、是否学区房、建造年代、是否二手、层数、所在层数），搞了这么多特征，结果只有不到十个房子的样例。要拟合房子特征->房价的这么多特征，就会造成过度拟合。

4、这个与第二个有点类似，假设在 IR 中我们建立的文档-词项矩阵中，有两个词项为“learn”和“study”，在传统的向量空间模型中，认为两者独立。然而从语义的角度来讲，两者是相似的，而且两者出现频率也类似，是不是可以合成为一个特征呢？

5、 在信号传输过程中，由于信道不是理想的，信道另一端收到的信号会有噪音扰动，那么怎么滤去这些噪音呢？

回顾我们之前介绍的《模型选择和规则化》，里面谈到的特征选择的问题。但在那篇中要剔除的特征主要是和类标签无关的特征。比如“学生的名字”就和他的“成绩”无关，使用的是互信息的方法。

而这里的特征很多是和类标签有关的，但里面存在噪声或者冗余。在这种情况下，需要一种特征降维的方法来减少特征数，减少噪音和冗余，减少过度拟合的可能性。

下面探讨一种称作主成分分析（PCA）的方法来解决部分上述问题。PCA 的思想是将 n 维特征映射到 k 维上（k<n），这 k 维是全新的正交特征。这 k 维特征称为主元，是重新构造出来的 k 维特征，而不是简单地从 n 维特征中去除其余 n-k 维特征。

2. PCA 计算过程

首先介绍 PCA 的计算过程：

假设我们得到的 2 维数据如下：

Data =

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

行代表了样例，列代表特征，这里有 10 个样例，每个样例两个特征。可以这样认为，有 10 篇文档，x 是 10 篇文档中“learn”出现的 TF-IDF，y 是 10 篇文档中“study”出现的 TF-IDF。也可以认为有 10 辆汽车，x 是千米/小时的速度，y 是英里/小时的速度，等等。

第一步分别求 x 和 y 的平均值，然后对于所有的样例，都减去对应的均值。这里 x 的均值是 1.81，y 的均值是 1.91，那么一个样例减去均值后即为（0.69,0.49），得到

DataAdjust =

x	y
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01

第二步，求特征协方差矩阵，如果数据是 3 维，那么协方差矩阵是

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

这里只有 x 和 y，求解得

$$cov = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

对角线上分别是 x 和 y 的方差，非对角线上是协方差。协方差大于 0 表示 x 和 y 若有一个增，另一个也增；小于 0 表示一个增，一个减；协方差为 0 时，两者独立。协方差绝对值越大，两者对彼此的影响越大，反之越小。

第三步，求协方差的特征值和特征向量，得到

$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

上面是两个特征值，下面是对应的特征向量，特征值 0.0490833989 对应特征向量为 $(-0.735178656, 0.677873399)^T$，这里的特征向量都归一化为单位向量。

第四步，将特征值按照从大到小的顺序排序，选择其中最大的 k 个，然后将其对应的 k 个特征向量分别作为列向量组成特征向量矩阵。

这里特征值只有两个，我们选择其中最大的那个，这里是 1.28402771，对应的特征向量是 $(-0.677873399, -0.735178656)^T$。

第五步，将样本点投影到选取的特征向量上。假设样例数为 m，特征数为 n，减去均值后的样本矩阵为 DataAdjust(m*n)，协方差矩阵是 n*n，选取的 k 个特征向量组成的矩阵为 EigenVectors(n*k)。那么投影后的数据 FinalData 为

$$FinalData(m*k) = DataAdjust(m*n) \times EigenVectors(n*k)$$

这里是

FinalData(10*1) = DataAdjust(10*2 矩阵)×特征向量$(-0.677873399, -0.735178656)^T$

得到结果是

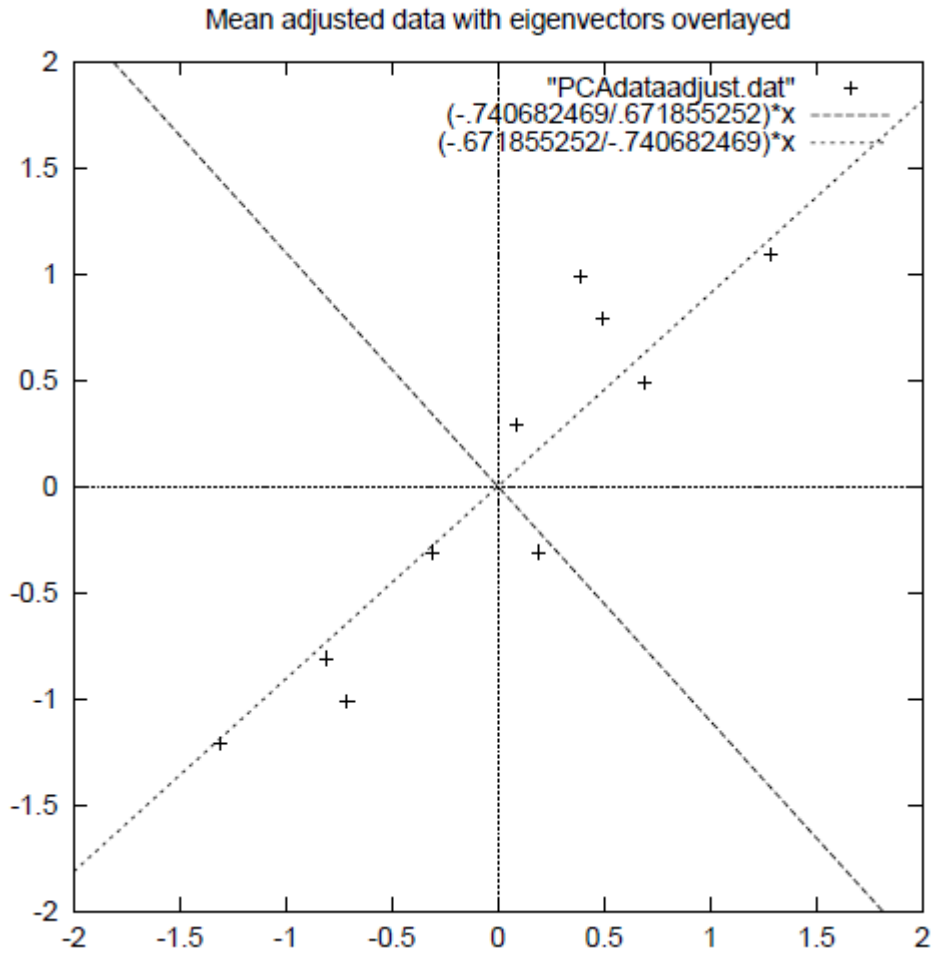
Transformed Data (Single eigenvector)

x
-.827970186
1.77758033
-.992197494
-.274210416
-1.67580142
-.912949103
.0991094375
1.14457216
.438046137
1.22382056

这样，就将原始样例的 n 维特征变成了 k 维，这 k 维就是原始特征在 k 维上的投影。

上面的数据可以认为是 learn 和 study 特征融合为一个新的特征叫做 LS 特征，该特征基本上代表了这两个特征。

上述过程有个图描述：

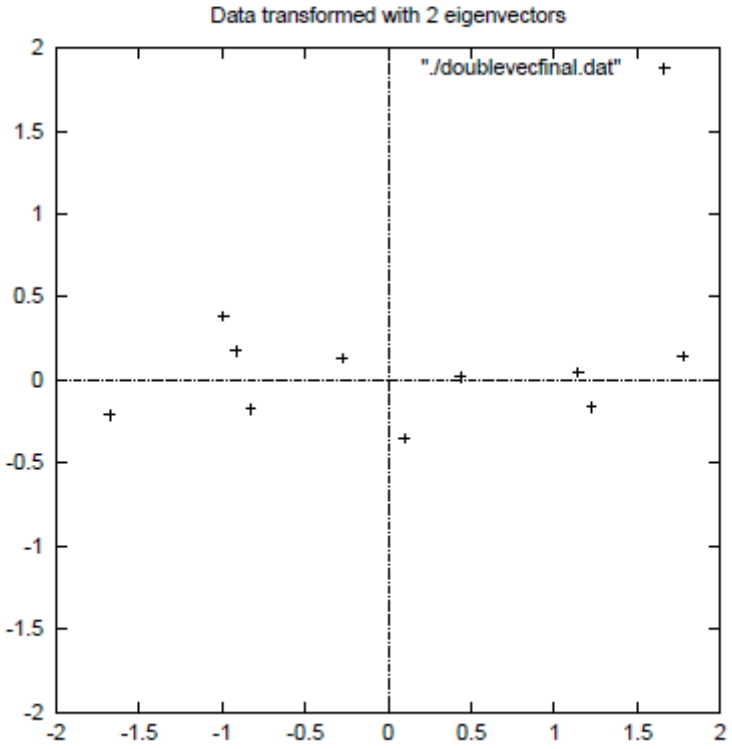


正号表示预处理后的样本点，斜着的两条线就分别是正交的特征向量（由于协方差矩阵是对称的，因此其特征向量正交），最后一步的矩阵乘法就是将原始样本点分别往特征向量对应的轴上做投影。

如果取的 k=2，那么结果是

Transformed Data=

x	y
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287



这就是经过 PCA 处理后的样本数据，水平轴（上面举例为 LS 特征）基本上可以代表全部样本点。整个过程看起来就像将坐标系做了旋转，当然二维可以图形化表示，高维就不行了。上面的如果 k=1，那么只会留下这里的水平轴，轴上是所有点在该轴的投影。

这样 PCA 的过程基本结束。在第一步减均值之后，其实应该还有一步对特征做方差归一化。比如一个特征是汽车速度（0 到 100），一个是汽车的座位数（2 到 6），显然第二个的方差比第一个小。因此，如果样本特征中存在这种情况，那么在第一步之后，求每个特征的标准差σ，然后对每个样例在该特征下的数据除以σ。

归纳一下，使用我们之前熟悉的表示方法，在求协方差之前的步骤是：

1. Let $\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$.
2. Replace each $x^{(i)}$ with $x^{(i)} - \mu$.
3. Let $\sigma_j^2 = \frac{1}{m}\sum_i (x_j^{(i)})^2$
4. Replace each $x_j^{(i)}$ with $x_j^{(i)}/\sigma_j$.

其中$\mathbf{x}^{(i)}$是样例，共 m 个，每个样例 n 个特征，也就是说$\mathbf{x}^{(i)}$是 n 维向量。$x_j^{(i)}$是第 i 个样例的第 j 个特征。μ是样例均值。σ_j是第 j 个特征的标准差。

整个 PCA 过程貌似及其简单，就是求协方差的特征值和特征向量，然后做数据转换。但是有没有觉得很神奇，为什么求协方差的特征向量就是最理想的 k 维向量？其背后隐藏的意义是什么？整个 PCA 的意义是什么？

3. PCA 理论基础

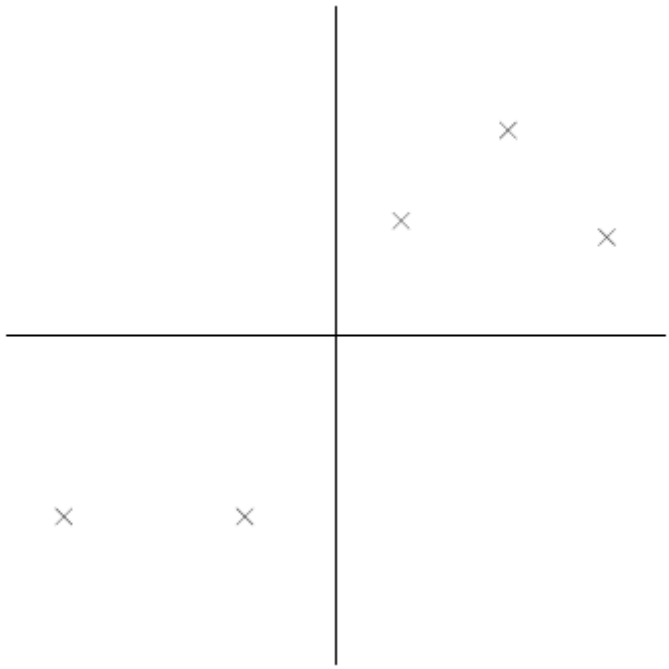
要解释为什么协方差矩阵的特征向量就是 k 维理想特征，我看到的有三个理论：分别是最大方差理论、最小错误理论和坐标轴相关度理论。这里简单探讨前两种，最后一种在讨论 PCA 意义时简单概述。

3.1 最大方差理论

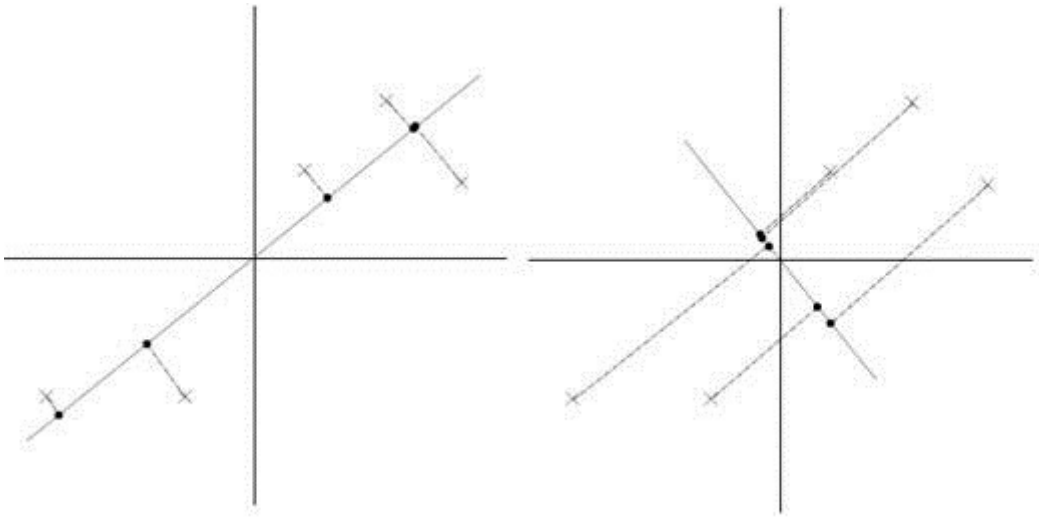
在信号处理中认为信号具有较大的方差，噪声有较小的方差，信噪比就是信号与噪声的方差比，越大越好。如前面的图，样本在横轴上的投影方差较大，在纵轴上的投影方差较小，那么认为纵轴上的投影是由噪声引起的。

因此我们认为，最好的 k 维特征是将 n 维样本点转换为 k 维后，每一维上的样本方差都很大。

比如下图有 5 个样本点：（已经做过预处理，均值为 0，特征方差归一）

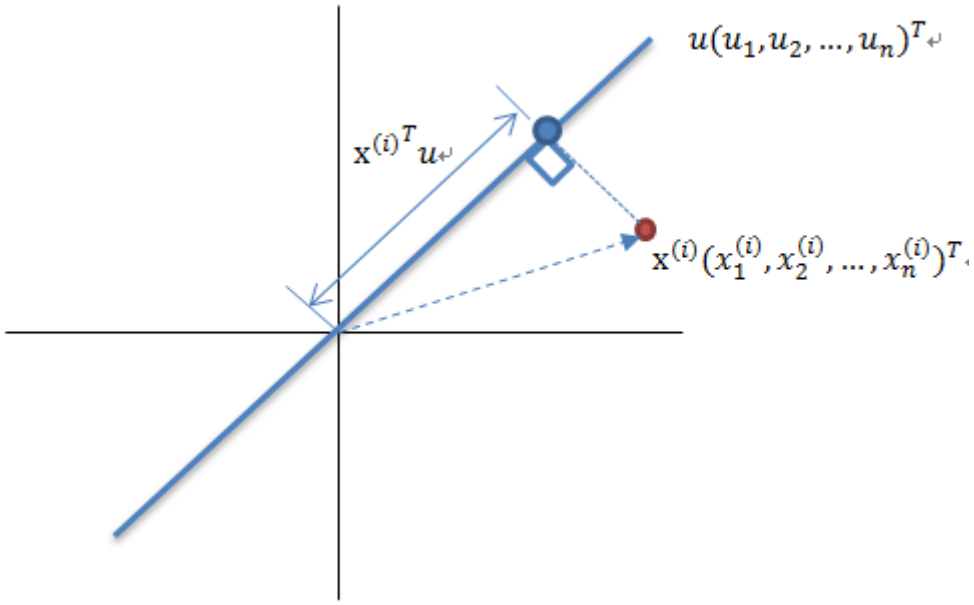


下面将样本投影到某一维上，这里用一条过原点的直线表示（前处理的过程实质是将原点移到样本点的中心点）。



假设我们选择两条不同的直线做投影，那么左右两条中哪个好呢？根据我们之前的方差最大化理论，左边的好，因为投影后的样本点之间方差最大。

这里先解释一下投影的概念：



红色点表示样例$\mathbf{x}^{(i)}$，蓝色点表示$\mathbf{x}^{(i)}$在 u 上的投影，u 是直线的斜率也是直线的方向向量，而且是单位向量。蓝色点是$\mathbf{x}^{(i)}$在 u 上的投影点，离原点的距离是$<\mathbf{x}^{(i)},u>$（即$\mathbf{x}^{(i)T}u$或者$\mathbf{u}^T\mathbf{x}^{(i)}$）由于这些样本点（样例）的每一维特征均值都为 0，因此投影到 u 上的样本点（只有一个到原点的距离值）的均值仍然是 0。

回到上面左右图中的左图，我们要求的是最佳的 u，使得投影后的样本点方差最大。

由于投影后均值为 0，因此方差为：

$$\begin{aligned}\frac{1}{m}\sum_{i=1}^{m}(x^{(i)^T}u)^2 &= \frac{1}{m}\sum_{i=1}^{m}u^T x^{(i)}x^{(i)^T}u \\ &= u^T\left(\frac{1}{m}\sum_{i=1}^{m}x^{(i)}x^{(i)^T}\right)u.\end{aligned}$$

中间那部分很熟悉啊，不就是样本特征的协方差矩阵么（$\mathbf{x^{(i)}}$的均值为 0，一般协方差矩阵都除以 m-1，这里用 m）。

用$\boldsymbol{\lambda}$来表示$\frac{1}{m}\sum_{i=1}^{m}(x^{(i)^T}u)^2$，$\boldsymbol{\Sigma}$表示$\frac{1}{m}\sum_{i=1}^{m}x^{(i)}x^{(i)^T}$，那么上式写作

$$\boldsymbol{\lambda = u^T \Sigma u}$$

由于 u 是单位向量，即$\boldsymbol{u^T u = 1}$，上式两边都左乘 u 得，$\boldsymbol{u\lambda = \lambda u = uu^T \Sigma u = \Sigma u}$

即$\boldsymbol{\Sigma u = \lambda u}$

We got it！$\boldsymbol{\lambda}$就是$\boldsymbol{\Sigma}$的特征值，u 是特征向量。最佳的投影直线是特征值$\boldsymbol{\lambda}$最大时对应的特征向量，其次是$\boldsymbol{\lambda}$第二大对应的特征向量，依次类推。

因此，我们只需要对协方差矩阵进行特征值分解，得到的前 k 大特征值对应的特征向量就是最佳的 k 维新特征，而且这 k 维新特征是正交的。得到前 k 个 u 以后，样例$\mathbf{x^{(i)}}$通过以下变换可以得到新的样本。

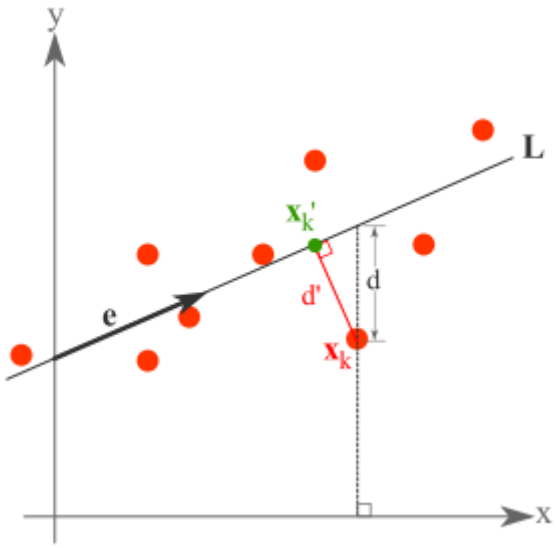
$$y^{(i)} = \begin{bmatrix} u_1^T x^{(i)} \\ u_2^T x^{(i)} \\ \vdots \\ u_k^T x^{(i)} \end{bmatrix} \in \mathbb{R}^k.$$

其中的第 j 维就是$\mathbf{x^{(i)}}$在$\mathbf{u_j}$上的投影。

通过选取最大的 k 个 u，使得方差较小的特征（如噪声）被丢弃。

这是其中一种对 PCA 的解释，第二种是错误最小化，放在下一篇介绍。

3.2 最小平方误差理论



假设有这样的二维样本点（红色点），回顾我们前面探讨的是求一条直线，使得样本点投影到直线上的点的方差最大。本质是求直线，那么度量直线求的好不好，不仅仅只有方差最大化的方法。再回想我们最开始学习的线性回归等，目的也是求一个线性函数使得直线能够最佳拟合样本点，那么我们能不能认为最佳的直线就是回归后的直线呢？回归时我们的最小二乘法度量的是样本点到直线的坐标轴距离。比如这个问题中，特征是 x，类标签是 y。回归时最小二乘法度量的是距离 d。如果使用回归方法来度量最佳直线，那么就是直接在原始样本上做回归了，跟特征选择就没什么关系了。

因此，我们打算选用另外一种评价直线好坏的方法，使用点到直线的距离 d’来度量。

现在有 n 个样本点$(x_1,x_2,...,x_n)$，每个样本点为 m 维（这节内容中使用的符号与上面的不太一致，需要重新理解符号的意义）。将样本点x_k在直线上的投影记为x_k'，那么我们就是要最小化

$$\sum_{k=1}^{n}||(x_k'-x_k)||^2$$

这个公式称作最小平方误差（Least Squared Error）。

而确定一条直线，一般只需要确定一个点，并且确定方向即可。

第一步确定点：

假设要在空间中找一点x_0来代表这 n 个样本点，“代表”这个词不是量化的，因此要量化的话，我们就是要找一个 m 维的点x_0，使得

$$J_0(x_0)=\sum_{k=1}^{n}\|x_0-x_k\|^2, \qquad (1)$$

最小。其中$J_0(x_0)$是平方错误评价函数（squared-error criterion function），假设 **m** 为 n 个样本点的均值：

$$m=\frac{1}{n}\sum_{k=1}^{n}x_k, \qquad (2)$$

那么平方错误可以写作：

$$
\begin{aligned}
J_0(x_0) &= \sum_{k=1}^{n} \|(x_0 - m) - (x_k - m)\|^2 \\
&= \sum_{k=1}^{n} \|x_0 - m\|^2 - 2\sum_{k=1}^{n}(x_0 - m)^t(x_k - m) + \sum_{k=1}^{n} \|x_k - m\|^2 \\
&= \sum_{k=1}^{n} \|x_0 - m\|^2 - 2(x_0 - m)^t \sum_{k=1}^{n}(x_k - m) + \sum_{k=1}^{n} \|x_k - m\|^2 \\
&= \sum_{k=1}^{n} \|x_0 - m\|^2 + \underbrace{\sum_{k=1}^{n} \|x_k - m\|^2}_{independent of x_0}. \qquad (3)
\end{aligned}
$$

后项与x_0无关，看做常量，而$J_0(x_0) \geq 0$，因此最小化$J_0(x_0)$时，

$$x_0 = \mathbf{m}$$

x_0是样本点均值。

第二步确定方向：

我们从x_0拉出要求的直线（这条直线要过点 **m**），假设直线的方向是单位向量 **e**。那么直线上任意一点，比如x_k'就可以用点 **m** 和 **e** 来表示

$$\mathbf{x}_k' = \mathbf{m} + a_k \boldsymbol{e}$$

其中a_k是x_k'到点 **m** 的距离。

我们重新定义最小平方误差：

$$J_1(a_1, \ldots, a_n, e) = \sum_{k=1}^{n} \|(x_k' - x_k)\|^2 = \sum_{k=1}^{n} \|((m + a_k e) - x_k)\|^2, \qquad (5)$$

这里的 k 只是相当于 *i*。J_1就是最小平方误差函数，其中的未知参数是$a_1, a_2, \ldots, a_n$和 **e**。

实际上是求J_1的最小值。首先将上式展开：

$$
\begin{aligned}
J_1(a_1, \ldots, a_n, e) &= \sum_{k=1}^{n} \|((m + a_k e) - x_k)\|^2 = \sum_{k=1}^{n} \|(a_k e - (x_k - m))\|^2 \\
&= \sum_{k=1}^{n} a_k^2 \|e\|^2 - 2\sum_{k=1}^{n} a_k e^t(x_k - m) + \sum_{k=1}^{n} \|x_k - m\|^2. \qquad (6)
\end{aligned}
$$

我们首先固定 **e**，将其看做是常量，$\|e\|^2 = 1$，然后对a_k进行求导，得

$$a_k = e^t(x_k - m). \qquad (8)$$

这个结果意思是说，如果知道了 **e**，那么将$\mathbf{x_k'-m}$与 **e** 做内积，就可以知道了x_k在 **e** 上的投影离 **m** 的长度距离，不过这个结果不用求都知道。

然后是固定$\mathbf{a_k}$，对 **e** 求偏导数，我们先将公式（8）代入J_1，得

$$\begin{aligned} J_1(\boldsymbol{e}) &= \sum_{k=1}^{n} a_k^2\|\boldsymbol{e}\|^2 - 2\sum_{k=1}^{n} a_k^2 + \sum_{k=1}^{n}\|\boldsymbol{x}_k - \boldsymbol{m}\|^2 \\ &= -\sum_{k=1}^{n}[\boldsymbol{e}^t(\boldsymbol{x}_k - m)]^2 + \sum_{k=1}^{n}\|\boldsymbol{x}_k - \boldsymbol{m}\|^2 \\ &= -\sum_{k=1}^{n}\boldsymbol{e}^t(\boldsymbol{x}_k - \boldsymbol{m})(\boldsymbol{x}_k - \boldsymbol{m})^t\boldsymbol{e} + \sum_{k=1}^{n}\|\boldsymbol{x}_k - \boldsymbol{m}\|^2 \\ &= -\boldsymbol{e}^t\boldsymbol{S}\boldsymbol{e} + \sum_{k=1}^{n}\|\boldsymbol{x}_k - \boldsymbol{m}\|^2. \end{aligned} \tag{9}$$

其中$\boldsymbol{S} = \sum_{k=1}^{n}\boldsymbol{e}^t(\boldsymbol{x}_k - \boldsymbol{m})(\boldsymbol{x}_k - \boldsymbol{m})^t\boldsymbol{e}$，与协方差矩阵类似，只是缺少个分母 n-1，我们称之为**散列矩阵**（scatter matrix）。

然后可以对 **e** 求偏导数，但是 **e** 需要首先满足$\|\mathbf{e}\|^2 = 1$，引入拉格朗日乘子λ，来使$\mathbf{e}^T\boldsymbol{S}\boldsymbol{e}$最大（$J_1$最小），令

$$u = \boldsymbol{e}^t\boldsymbol{S}\boldsymbol{e} - \lambda(\boldsymbol{e}^t\boldsymbol{e} - 1) \tag{10}$$

求偏导

$$\frac{\partial u}{\partial \boldsymbol{e}} = 2\boldsymbol{S}\boldsymbol{e} - 2\lambda\boldsymbol{e}, \tag{11}$$

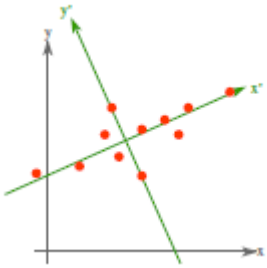
这里存在对向量求导数的技巧，方法这里不多做介绍。可以去看一些关于矩阵微积分的资料，这里求导时可以将$\mathbf{e}^T\boldsymbol{S}\boldsymbol{e}$看作是$\mathbf{S}\mathbf{e}^2$，将$\mathbf{e}^T\boldsymbol{e}$看做是$\mathbf{e}^2$。

导数等于 0 时，得

$$\boldsymbol{S}\boldsymbol{e} = \lambda\boldsymbol{e}. \tag{12}$$

两边除以 n-1 就变成了，对协方差矩阵求特征值向量了。

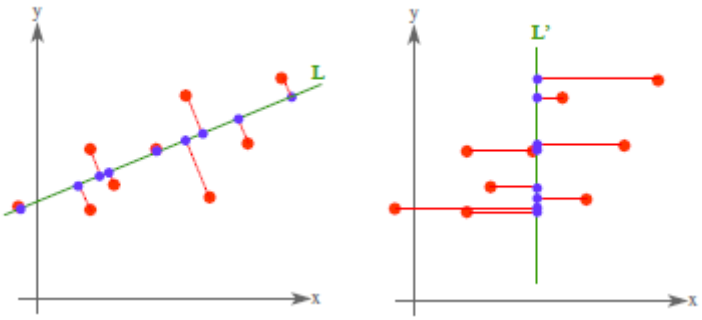
从不同的思路出发，最后得到同一个结果，对协方差矩阵求特征向量，求得后特征向量上就成为了新的坐标，如下图：



这时候点都聚集在新的坐标轴周围，因为我们使用的最小平方误差的意义就在此。

4. PCA 理论意义

PCA 将 n 个特征降维到 k 个，可以用来进行数据压缩，如果 100 维的向量最后可以用 10 维来表示，那么压缩率为 90%。同样图像处理领域的 KL 变换使用 PCA 做图像压缩。但 PCA 要保证降维后，还要保证数据的特性损失最小。再看回顾一下 PCA 的效果。经过 PCA 处理后，二维数据投影到一维上可以有以下几种情况：



我们认为左图好，一方面是投影后方差最大，一方面是点到直线的距离平方和最小，而且直线过样本点的中心点。为什么右边的投影效果比较差？直觉是因为坐标轴之间相关，以至于去掉一个坐标轴，就会使得坐标点无法被单独一个坐标轴确定。

PCA 得到的 k 个坐标轴实际上是 k 个特征向量，由于协方差矩阵对称，因此 k 个特征向量正交。看下面的计算过程。

假设我们还是用$\mathbf{x}^{(i)}(x_1^{(i)},x_2^{(i)},\dots,x_n^{(i)})^T$来表示样例，m 个样例，n 个特征。特征向量为 **e**，$e_1^{(i)}$表示第 i 个特征向量的第 1 维。那么原始样本特征方程可以用下面式子来表示：

前面两个矩阵乘积就是协方差矩阵Σ（除以 m 后），原始的样本矩阵 A 是第二个矩阵 m*n。

$$\begin{bmatrix} | & | & | & | \\ \mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \dots & \mathbf{x}^{(m)} \\ | & | & | & | \end{bmatrix}\begin{bmatrix} - & \mathbf{x}^{(1)T} & - \\ - & \mathbf{x}^{(2)T} & - \\ - & \vdots & - \\ - & \mathbf{x}^{(m)T} & - \end{bmatrix}\begin{bmatrix} e_1^{(i)} \\ e_2^{(i)} \\ \vdots \\ e_n^{(i)} \end{bmatrix} = \lambda_i \begin{bmatrix} e_1^{(i)} \\ e_2^{(i)} \\ \vdots \\ e_n^{(i)} \end{bmatrix}$$

上式可以简写为$A^TAe = \lambda e$

我们最后得到的投影结果是AE，E 是 k 个特征向量组成的矩阵，展开如下：

$$\begin{bmatrix} - & x^{(1)T} & - \\ - & x^{(2)T} & - \\ - & \vdots & - \\ - & x^{(m)T} & - \end{bmatrix} \begin{bmatrix} e_1^{(1)} & e_1^{(2)} & \cdots & e_1^{(k)} \\ e_2^{(1)} & e_2^{(2)} & \cdots & e_2^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ e_n^{(1)} & e_n^{(2)} & \cdots & e_n^{(k)} \end{bmatrix}$$

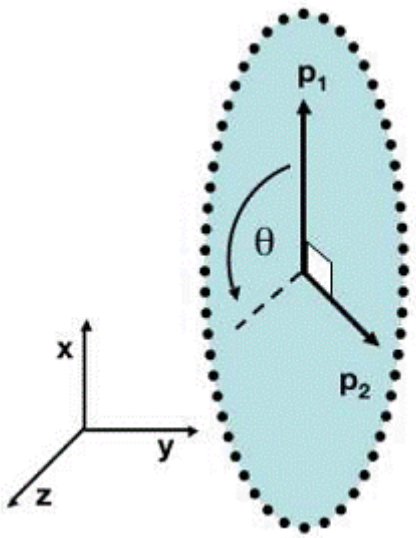
得到的新的样例矩阵就是 m 个样例到 k 个特征向量的投影，也是这 k 个特征向量的线性组合。e 之间是正交的。从矩阵乘法中可以看出，PCA 所做的变换是将原始样本点（n 维），投影到 k 个正交的坐标系中去，丢弃其他维度的信息。举个例子，假设宇宙是 n 维的（霍金说是 13 维的），我们得到银河系中每个星星的坐标（相对于银河系中心的 n 维向量），然而我们想用二维坐标去逼近这些样本点，假设算出来的协方差矩阵的特征向量分别是图中的水平和竖直方向，那么我们建议以银河系中心为原点的 x 和 y 坐标轴，所有的星星都投影到 x 和 y 上，得到下面的图片。然而我们丢弃了每个星星离我们的远近距离等信息。

5. 总结与讨论

这一部分来自 http://www.cad.zju.edu.cn/home/chenlu/pca.htm

PCA 技术的一大好处是对数据进行降维的处理。我们可以对新求出的“主元”向量的重要性进行排序，根据需要取前面最重要的部分，将后面的维数省去，可以达到降维从而简化模型或是对数据进行压缩的效果。同时最大程度的保持了原有数据的信息。

PCA 技术的一个很大的优点是，它是完全无参数限制的。在 PCA 的计算过程中完全不需要人为的设定参数或是根据任何经验模型对计算进行干预，最后的结果只与数据相关，与用户是独立的。
但是，这一点同时也可以看作是缺点。如果用户对观测对象有一定的先验知识，掌握了数据的一些特征，却无法通过参数化等方法对处理过程进行干预，可能会得不到预期的效果，效率也不高。



图表 4：黑色点表示采样数据，排列成转盘的形状。

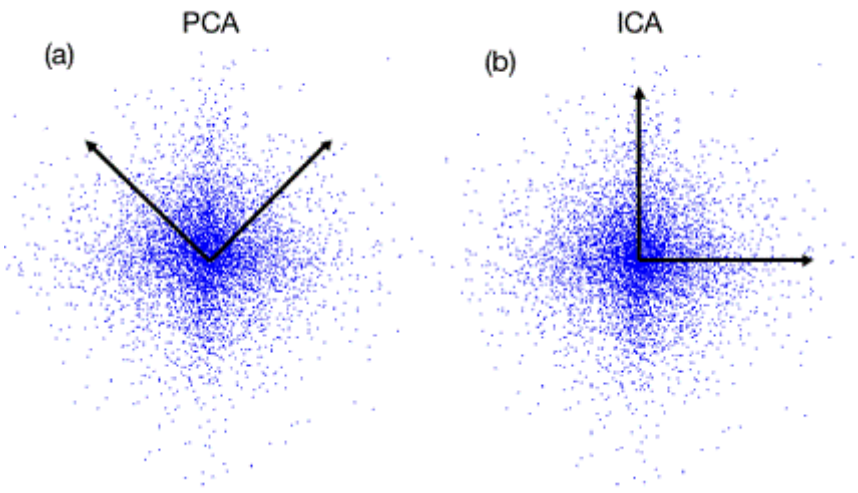
容易想象，该数据的主元是(P_1, P_2)或是旋转角θ。

如图表 4 中的例子，PCA 找出的主元将是(P_1, P_2)。但是这显然不是最优和最简化的主元。(P_1, P_2)之间存在着非线性的关系。根据先验的知识可知旋转角θ是最优的主元（类比极坐标）。则在这种情况下，PCA 就会失效。但是，如果加入先验的知识，对数据进行某种划归，就可以将数据转化为以θ为线性的空间中。这类根据先验知识对数据预先进行非线性转换的方法就成为 *kernel*-PCA，它扩展了 PCA 能够处理的问题的范围，又可以结合一些先验约束，是比较流行的方法。

有时数据的分布并不是满足高斯分布。如图表 5 所示，在非高斯分布的情况下，PCA 方法得出的主元可能并不是最优的。在寻找主元时不能将方差作为衡量重要性的标准。要根据数据的分布情况选择合适的描述完全分布的变量，然后根据概率分布式

$$P(y_1, y_2) = P(y_1)P(y_2)$$

来计算两个向量上数据分布的相关性。等价的，保持主元间的正交假设，寻找的主元同样要使$P(y_1, y_2) = 0$。这一类方法被称为独立主元分解(ICA)。



图表 5：数据的分布并不满足高斯分布，呈明显的十字星状。
这种情况下，方差最大的方向并不是最优主元方向。

另外 PCA 还可以用于预测矩阵中缺失的元素。

独立成分分析（Independent Component Analysis）

1. 问题：

1、上节提到的 PCA 是一种数据降维的方法，但是只对符合高斯分布的样本点比较有效，那么对于其他分布的样本，有没有主元分解的方法呢？

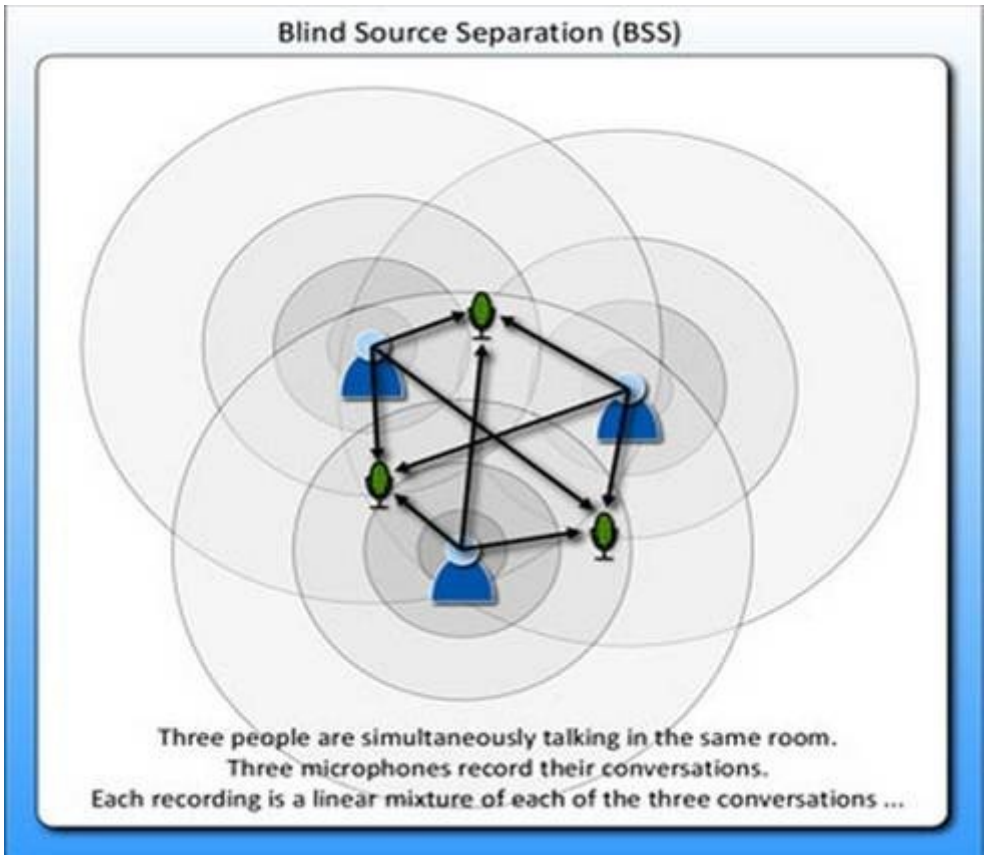
2、经典的鸡尾酒宴会问题（cocktail party problem）。假设在 party 中有 n 个人，他们可以同时说话，我们也在房间中一些角落里共放置了 n 个声音接收器（Microphone）用来记录声音。宴会过后，我们从 n 个麦克风中得到了一组数据$\{\mathbf{x}^{(i)}(x_1^{(i)},x_2^{(i)},\ldots,x_n^{(i)});i=1,\ldots,m\}$，i 表示采样的时间顺序，也就是说共得到了 m 组采样，每一组采样都是 n 维的。我们的目标是单单从这 m 组采样数据中分辨出每个人说话的信号。

将第二个问题细化一下，有 n 个信号源$\mathbf{s}\ (s_1,s_2,\ldots,s_n)^T$，$\mathbf{s}\in\mathbb{R}^n$，每一维都是一个人的声音信号，每个人发出的声音信号独立。A 是一个未知的混合矩阵（mixing matrix），用来组合叠加信号 s，那么

$$\mathbf{x}=\mathbf{As}$$

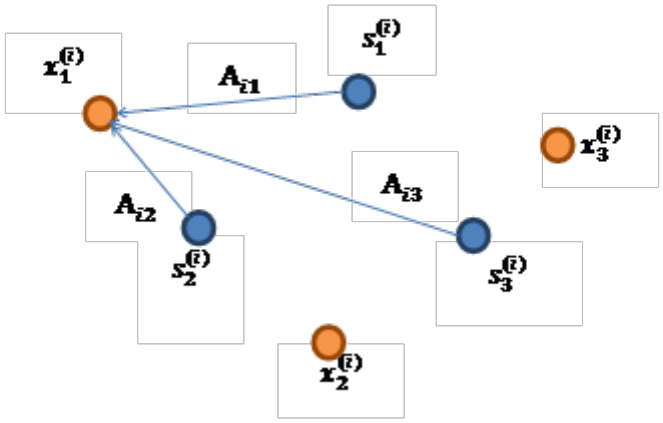
x 的意义在上文解释过，这里的 x 不是一个向量，是一个矩阵。其中每个列向量是$\mathbf{x}^{(i)}$，$x^{(i)}=As^{(i)}$

表示成图就是



这张图来自

http://amouraux.webnode.com/research-interests/research-interests-erp-analysis/blind-source-separation-bss-of-erps-using-independent-component-analysis-ica/



$\mathbf{x}^{(i)}$的每个分量都由$\mathbf{s}^{(i)}$的分量线性表示。A 和 s 都是未知的，x 是已知的，我们要想办法根据 x 来推出 s。这个过程也称作为盲信号分离。

令$W = A^{-1}$，那么$s^{(i)} = \mathrm{A}^{-1}x^{(i)} = Wx^{(i)}$

将 W 表示成

$$W = \begin{bmatrix} — & w_1^T & — \\ & \vdots & \\ — & w_n^T & — \end{bmatrix}.$$

其中$w_i \in \mathbb{R}^n$，其实就是将w_i写成行向量形式。那么得到：

$$s_j^{(i)} = w_j^T x^{(i)}$$

2. ICA 的不确定性（ICA ambiguities）

由于 w 和 s 都不确定，那么在没有先验知识的情况下，无法同时确定这两个相关参数。比如上面的公式 s=wx。当 w 扩大两倍时，s 只需要同时扩大两倍即可，等式仍然满足，因此无法得到唯一的 s。同时如果将人的编号打乱，变成另外一个顺序，如上图的蓝色节点的编号变为 3,2,1，那么只需要调换 A 的列向量顺序即可，因此也无法单独确定 s。这两种情况称为原信号不确定。

还有一种 ICA 不适用的情况，那就是信号不能是高斯分布的。假设只有两个人发出的声音信号符合多值正态分布，$s \sim N(0, I)$，I 是 2*2 的单位矩阵，s 的概率密度函数就不用说了吧，以均值 0 为中心，投影面是椭圆的山峰状（参见多值高斯分布）。因为$x = As$，因此，x 也是高斯分布的，均值为 0，协方差为

$E[xx^T] = E[Ass^TA^T] = AA^T$。

令 R 是正交阵$(RR^T = R^TR = I)$，$A' = AR$。如果将 A 替换成 A’。那么$x' = A's$。s 分布没变，因此 x’仍然是均值为 0，协方差$E[x'(x')^T] = E[A'ss^T(A')^T] = E[ARss^T(AR)^T] = ARR^TA^T = AA^T$。

因此，不管混合矩阵是 A 还是 A’，x 的分布情况是一样的，那么就无法确定混合矩阵，也就无法确定原信号。

3. 密度函数和线性变换

在讨论 ICA 具体算法之前，我们先来回顾一下概率和线性代数里的知识。

假设我们的随机变量 s 有概率密度函数$p_s(s)$（连续值是概率密度函数，离散值是概率）。为了简单，我们再假设 s 是实数，还有一个随机变量 x=As，A 和 x 都是实数。令p_x是 x 的概率密度，那么怎么求p_x？

令$W = A^{-1}$，首先将式子变换成$s = Wx$，然后得到$p_x(x) = p_s(Ws)$，求解完毕。可惜这种方法是错误的。比如 s 符合均匀分布的话（$s \sim \text{Uniform}[0,1]$），那么 s 的概率密度是$p_s(s) = 1\{0 \le s \le 1\}$，现在令 A=2，即 x=2s，也就是说 x 在[0,2]上均匀分布，可知$p_x(x) = 0.5$。然而，前面的推导会得到$p_x(x) = p_s(0.5s) = 1$。正确的公式应该是

$$p_x(x) = p_s(Wx)|W|$$

推导方法

$$F_X(x) = P(X \le x) = P(AS \le x) = P(S \le Wx) = F_S(Wx)$$

$$p_x(x) = F_X'(x) = F_S'(Wx) = p_s(Wx)|W|$$

更一般地，如果 s 是向量，A 可逆的方阵，那么上式子仍然成立。

4. ICA 算法

ICA 算法归功于 Bell 和 Sejnowski，这里使用最大似然估计来解释算法，原始的论文中使用的是一个复杂的方法 Infomax principal。

我们假定每个s_i有概率密度p_s，那么给定时刻原信号的联合分布就是

$$p(s) = \prod_{i=1}^{n} p_s(s_i)$$

这个公式代表一个假设前提：每个人发出的声音信号各自独立。有了 p(s)，我们可以求得 p(x)

$$p(x) = p_s(Wx)|W| = |W| \prod_{i=1}^{n} p_s(w_i^T x)$$

左边是每个采样信号 x（n 维向量）的概率，右边是每个原信号概率的乘积的|W|倍。

前面提到过，如果没有先验知识，我们无法求得 W 和 s。因此我们需要知道$p_s(s_i)$，我们打算选取一个概率密度函数赋给 s，但是我们不能选取高斯分布的密度函数。在概率论里我们知道密度函数 p(x)由累计分布函数（cdf）F(x)求导得到。F(x)要满足两个性质是：单调递增和在[0,1]。我们发现 sigmoid 函数很适合，定义域负无穷到正无穷，值域 0 到 1，缓慢递增。我们假定 s 的累积分布函数符合 sigmoid 函数

$$g(s) = \frac{1}{1+e^{-s}}$$

求导后

$$p_s(s) = g'(s) = \frac{e^s}{(1+e^s)^2}$$

这就是 s 的密度函数。这里 s 是实数。

如果我们预先知道 s 的分布函数，那就不用假设了，但是在缺失的情况下，sigmoid 函数能够在大多数问题上取得不错的效果。由于上式中$p_s(s)$是个对称函数，因此 E[s]=0（s 的均值为 0），那么 E[x]=E[As]=0，x 的均值也是 0。

知道了$p_s(s)$，就剩下 W 了。给定采样后的训练样本$\{x^{(i)}(x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}); i = 1, \ldots, m\}$，样本对数似然估计如下：

使用前面得到的 x 的概率密度函数，得

$$\ell(W) = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \log g'(w_j^T x^{(i)}) + \log |W| \right).$$

大括号里面是$p(x^{(i)})$。

接下来就是对 W 求导了，这里牵涉一个问题是对行列式|W|进行求导的方法，属于矩阵微积分。这里先给出结果，在文章最后再给出推导公式。

$$\nabla_W |W| = |W|(W^{-1})^T$$

最终得到的求导后公式如下，$\log g'(s)$的导数为$1 - 2g(s)$（可以自己验证）：

$$W := W + \alpha \left(\begin{bmatrix} 1 - 2g(w_1^T x^{(i)}) \\ 1 - 2g(w_2^T x^{(i)}) \\ \vdots \\ 1 - 2g(w_n^T x^{(i)}) \end{bmatrix} x^{(i)T} + (W^T)^{-1} \right),$$

其中α是梯度上升速率，人为指定。

当迭代求出 W 后，便可得到$s^{(i)} = Wx^{(i)}$来还原出原始信号。

注意：我们计算最大似然估计时，假设了$x^{(i)}$与$x^{(j)}$之间是独立的，然而对于语音信号或者其他具有时间连续依赖特性（比如温度）上，这个假设不能成立。但是在数据足够多时，假设独立对效果影响不大，同时如果事先打乱样例，并运行随机梯度上升算法，那么能够加快收敛速度。

回顾一下鸡尾酒宴会问题，s 是人发出的信号，是连续值，不同时间点的 s 不同，每个人发出的信号之间独立（s_i和s_j之间独立）。s 的累计概率分布函数是 sigmoid 函数，但是所有人发出声音信号都符合这个分布。A（W 的逆阵）代表了 s 相对于 x 的位置变化，x 是 s 和 A 变化后的结果。

5. 实例

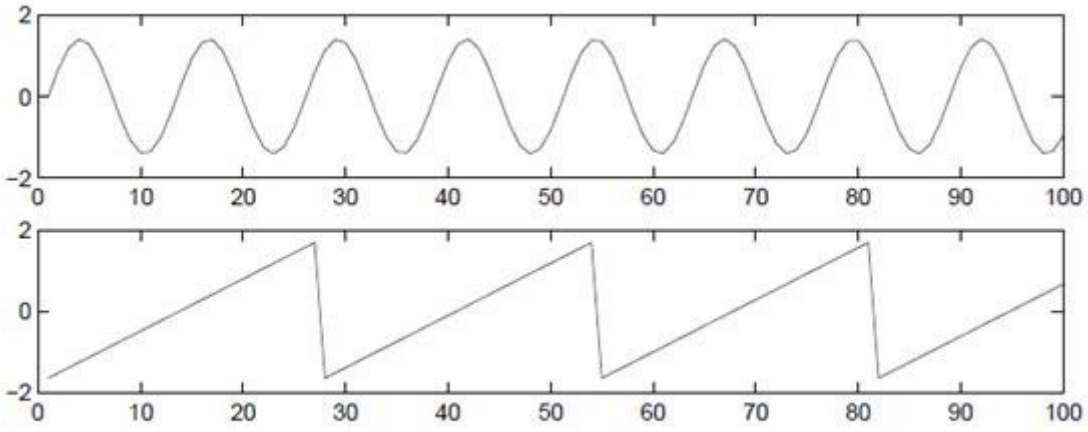


Figure 1: The original signals.

s=2 时的原始信号

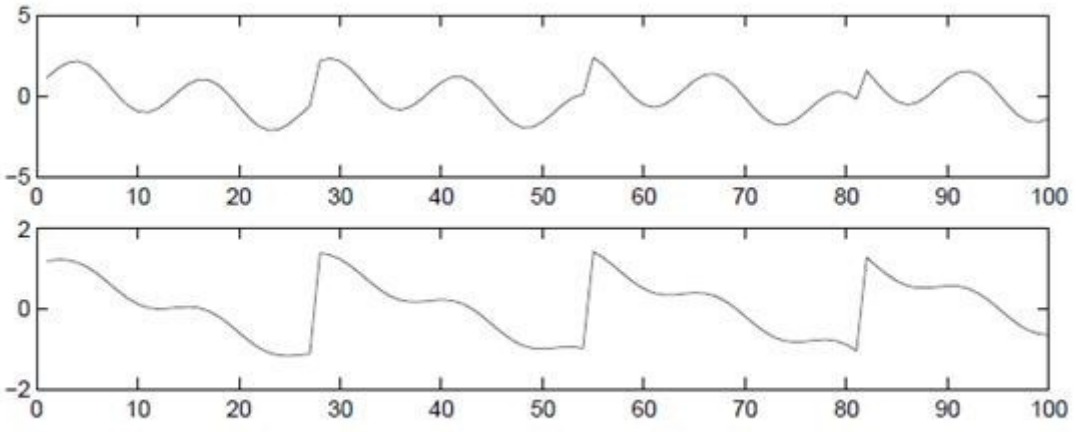
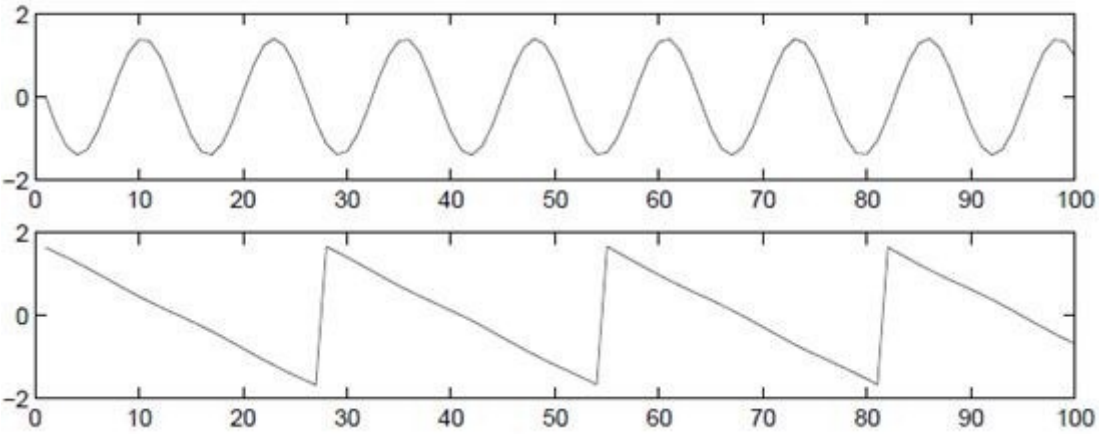


Figure 2: The observed mixtures of the source signals in Fig. 1.

观察到的 x 信号



使用 ICA 还原后的 s 信号

6. 行列式的梯度

对行列式求导，设矩阵 A 是 n×n 的，我们知道行列式与代数余子式有关，

$$|A| = \sum_{i=1}^{n} (-1)^{i+j} A_{ij} |A_{\setminus i,\setminus j}| \quad \text{(for any } j \in 1, \ldots, n)$$

$A_{\setminus i,\setminus j}$是去掉第 i 行第 j 列后的余子式，那么对$A_{k,l}$求导得

$$\frac{\partial}{\partial A_{k\ell}} |A| = \frac{\partial}{\partial A_{k\ell}} \sum_{i=1}^{n} (-1)^{i+j} A_{ij} |A_{\setminus i,\setminus j}| = (-1)^{k+\ell} |A_{\setminus k,\setminus \ell}| = (\operatorname{adj}(A))_{\ell k}.$$

adj(A)跟我们线性代数中学的A^*是一个意思，因此

$$\nabla_A |A| = (\operatorname{adj}(A))^T = |A| A^{-T}.$$