主成分分析（Principal components analysis）

JerryLead
csxulijie@gmail.com

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：

	x	y
	2.5	2.4
	0.5	0.7
	2.2	2.9
	1.9	2.2
Data =	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

	x	y
	.69	.49
	-1.31	-1.21
	.39	.99
	.09	.29
DataAdjust =	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

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

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

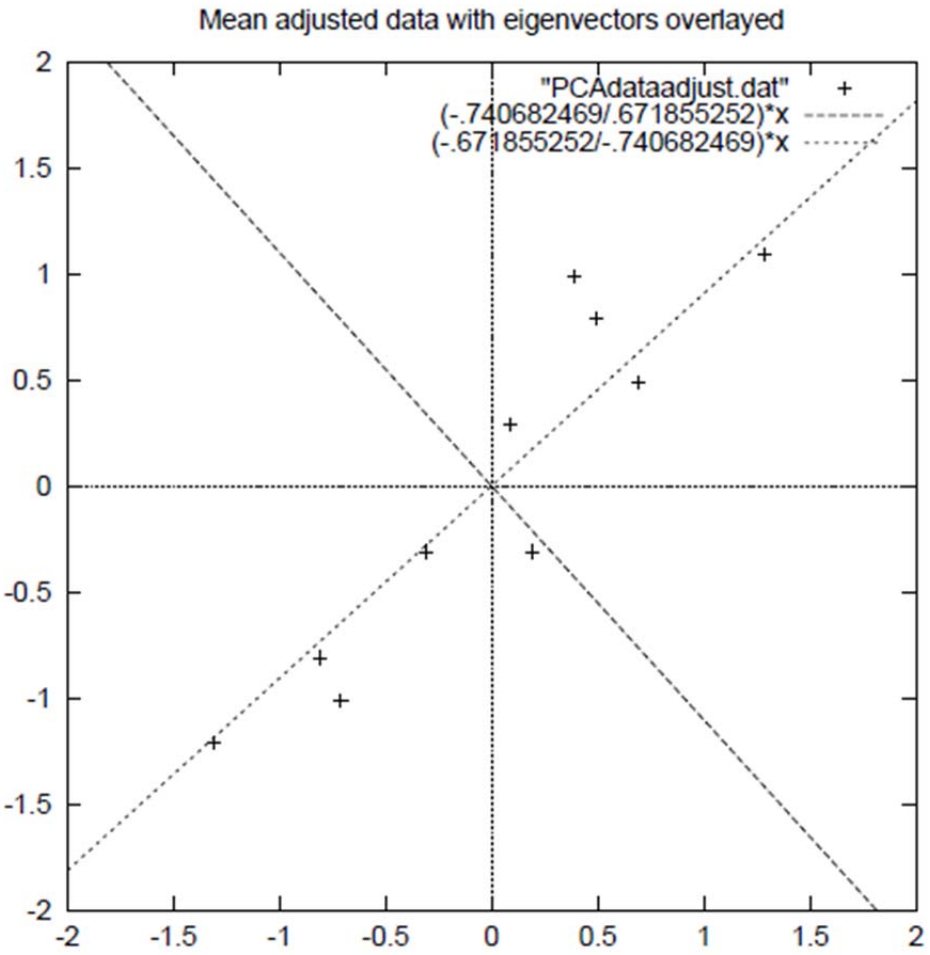
Transformed Data (Single eigenvector)

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：

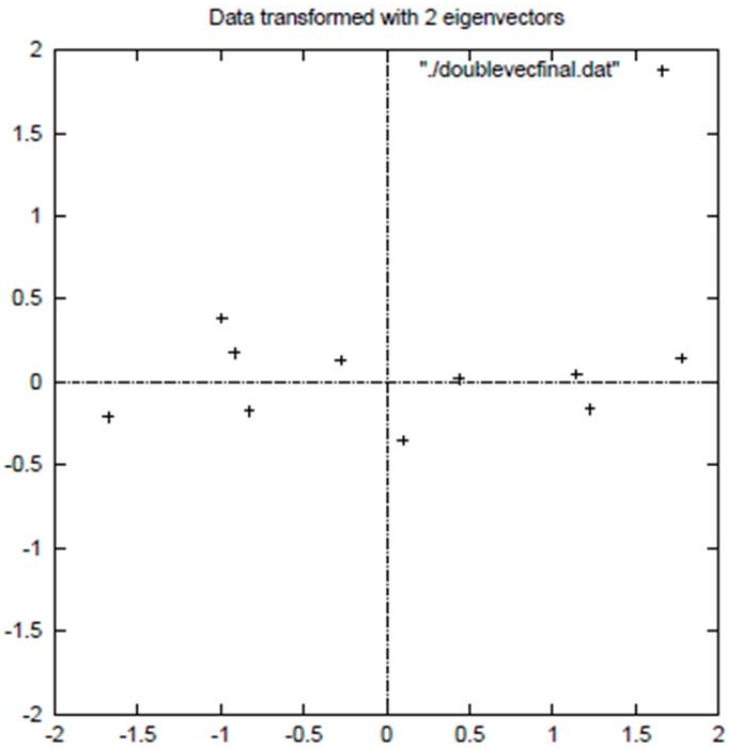


正号表示预处理后的样本点，斜着的两条线就分别是正交的特征向量（由于协方差矩阵是对称的，因此其特征向量正交），最后一步的矩阵乘法就是将原始样本点分别往特征向量对应的轴上做投影。

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$x^{(i)}$是样例，共 m 个，每个样例 n 个特征，也就是说$x^{(i)}$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

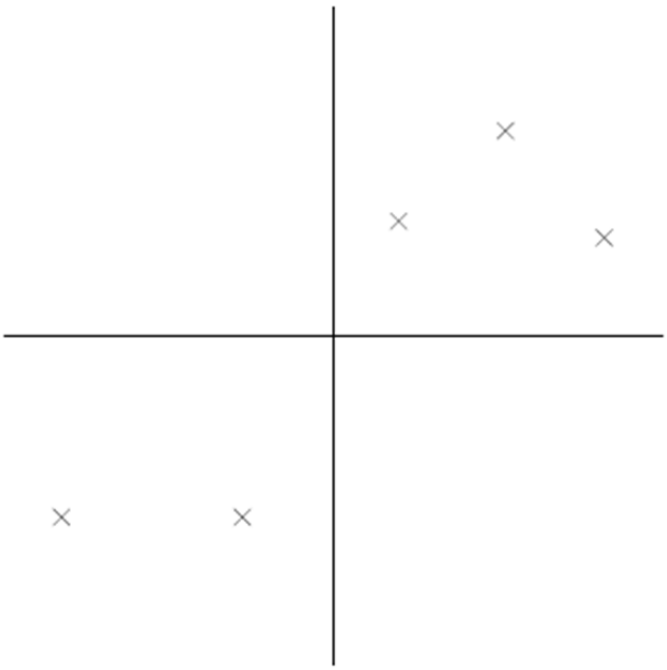
要解释为什么协方差矩阵的特征向量就是 k 维理想特征，我看到的有三个理论：分别是最大方差理论、最小错误理论和坐标轴相关度理论。这里简单探讨前两种，最后一种在讨论 PCA 意义时简单概述。

3.1 最大方差理论

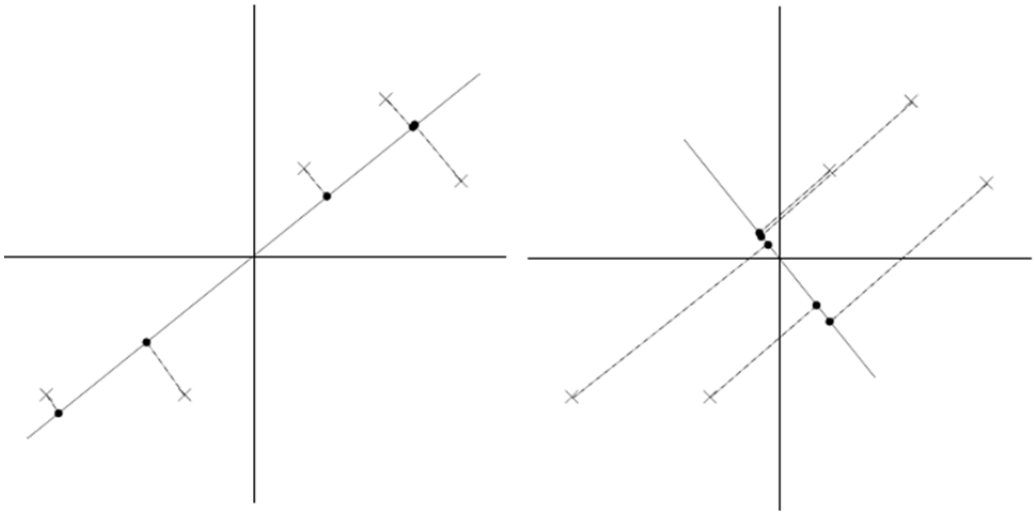
在信号处理中认为信号具有较大的方差，噪声有较小的方差，信噪比就是信号与噪声的方差比，越大越好。如前面的图，样本在横轴上的投影方差较大，在纵轴上的投影方差较小，那么认为纵轴上的投影是由噪声引起的。

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）

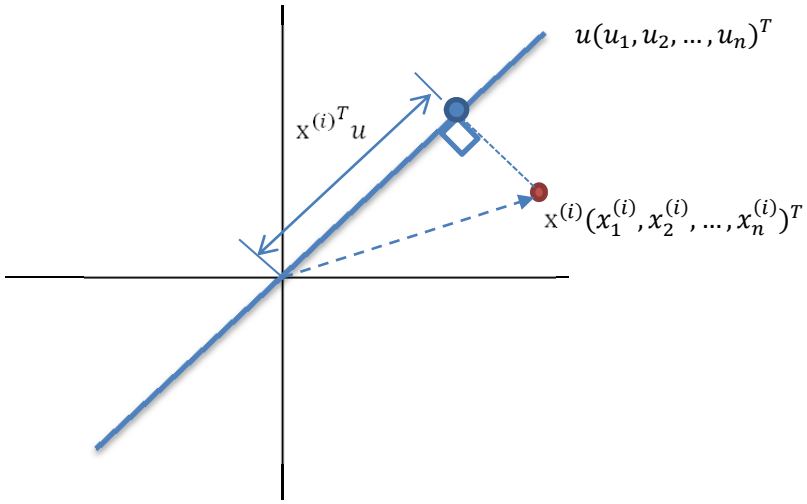


下面将样本投影到某一维上，这里用一条过原点的直线表示（前处理的过程实质是将原点移到样本点的中心点）。



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$x^{(i)}$，蓝色点表示$x^{(i)}$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$x^{(i)}$在 u 上的投影点，离原点的距离是$< x^{(i)}, u >$（即$x^{(i)T}u$或者$u^T x^{(i)}$）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（$x^{(i)}$的均值为 0，一般协方差矩阵都除以 m-1，这里用 m）。

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

$\lambda = u^T\Sigma u$

由于 u 是单位向量，即$u^Tu = 1$，上式两边都左乘 u 得，

$u\lambda = \lambda u = uu^T\Sigma u = \Sigma u$

即$\Sigma u = \lambda u$

We got it！λ就是Σ的特征值，u 是特征向量。最佳的投影直线是特征值λ最大时对应的特征向量，其次是λ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$x^{(i)}$通过以下变换可以得到新的样本。

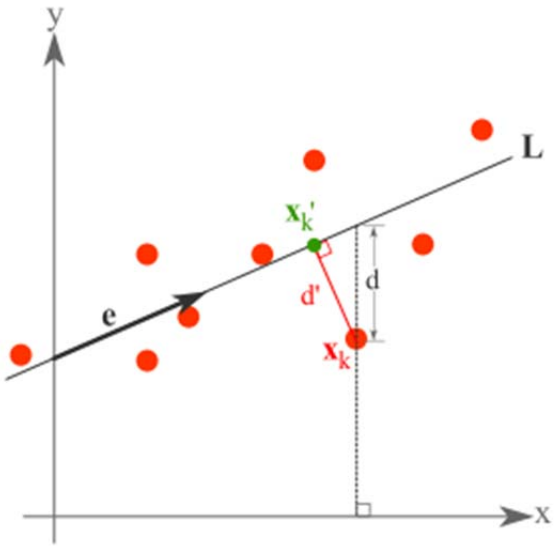
$$y^{(i)} = \begin{bmatrix} u_1^T x^{(i)} \\ u_2^T x^{(i)} \\ \vdots \\ u_k^T x^{(i)} \end{bmatrix} \in \mathbb{R}^k.$$

其中的第 j 维就是$x^{(i)}$在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。

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$(x_1, x_2, \ldots, x_n)$，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

$$J_0(\boldsymbol{x}_0) = \sum_{k=1}^{n} \|\boldsymbol{x}_0 - \boldsymbol{x}_k\|^2, \qquad (1)$$

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：

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

那么平方错误可以写作：

$$\begin{aligned} J_0(\boldsymbol{x}_0) &= \sum_{k=1}^{n} \|(\boldsymbol{x}_0 - \boldsymbol{m}) - (\boldsymbol{x}_k - \boldsymbol{m})\|^2 \\ &= \sum_{k=1}^{n} \|\boldsymbol{x}_0 - \boldsymbol{m}\|^2 - 2\sum_{k=1}^{n}(\boldsymbol{x}_0 - \boldsymbol{m})^t(\boldsymbol{x}_k - \boldsymbol{m}) + \sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2 \\ &= \sum_{k=1}^{n} \|\boldsymbol{x}_0 - \boldsymbol{m}\|^2 - 2(\boldsymbol{x}_0 - \boldsymbol{m})^t\sum_{k=1}^{n}(\boldsymbol{x}_k - \boldsymbol{m}) + \sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2 \\ &= \sum_{k=1}^{n} \|\boldsymbol{x}_0 - \boldsymbol{m}\|^2 + \underbrace{\sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2}_{independent of \boldsymbol{x}_0}. \end{aligned} \qquad (3)$$

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。

第一步确定方向：

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

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

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：

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

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：

$$J_1(a_1,\ldots,a_n,e)=\sum_{k=1}^{n}\|((\boldsymbol{m}+a_k\boldsymbol{e})-\boldsymbol{x}_k)\|^2=\sum_{k=1}^{n}\|(a_k\boldsymbol{e}-(\boldsymbol{x}_k-\boldsymbol{m}))\|^2$$
$$=\sum_{k=1}^{n}a_k^2\|\boldsymbol{e}\|^2-2\sum_{k=1}^{n}a_k\boldsymbol{e}^t(\boldsymbol{x}_k-\boldsymbol{m})+\sum_{k=1}^{n}\|\boldsymbol{x}_k-\boldsymbol{m}\|^2. \quad (6)$$

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

$$a_k=\boldsymbol{e}^t(\boldsymbol{x}_k-\boldsymbol{m}). \quad (8)$$

这个结果意思是说，如果知道了 **e**，那么将$x_k'-\mathbf{m}$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a_k，对 **e** 求偏导数，我们先将公式（8）代入J_1，得

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

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$||e||^2=1$，引入拉格朗日乘子λ，来使$e^T Se$最大（J_1最小），令

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

求偏导

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

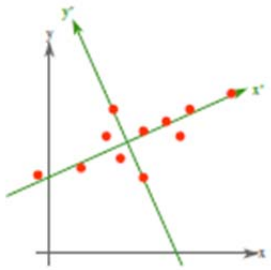
这里存在对向量求导数的技巧，方法这里不多做介绍。可以去看一些关于矩阵微积分的资料，这里求导时可以将$e^T Se$看作是Se^2，将$e^T e$看做是e^2。

导数等于 0 时，得

$$\boldsymbol{S}\boldsymbol{e}=\lambda\boldsymbol{e}. \quad (12)$$

两边除以 n-1 就变成了，对协方差矩阵求特征值向量了。

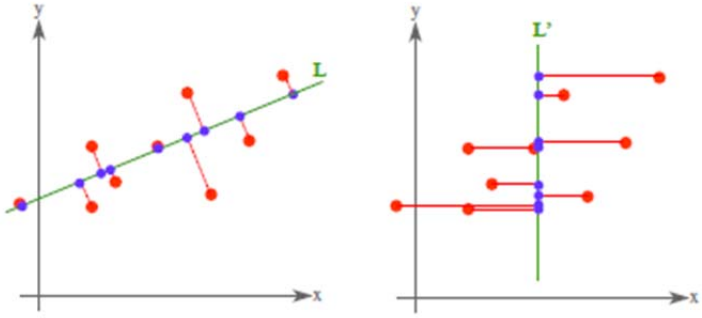
从不同的思路出发，最后得到同一个结果，对协方差矩阵求特征向量，求得后特征向量上就成为了新的坐标，如下图：



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$\mathrm{x}^{(i)}(x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)})^T$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。

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

上式可以简写为$\mathrm{A}^T Ae = \lambda e$

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：

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

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。

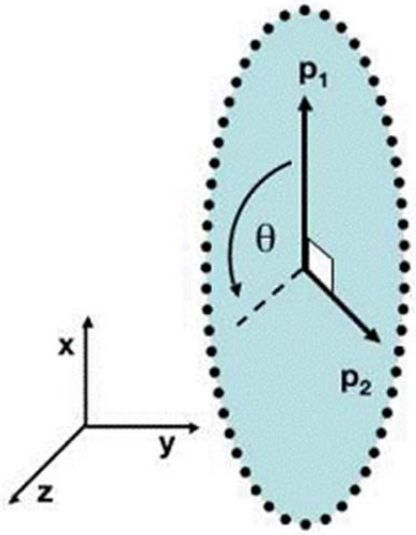
5. 总结与讨论

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。

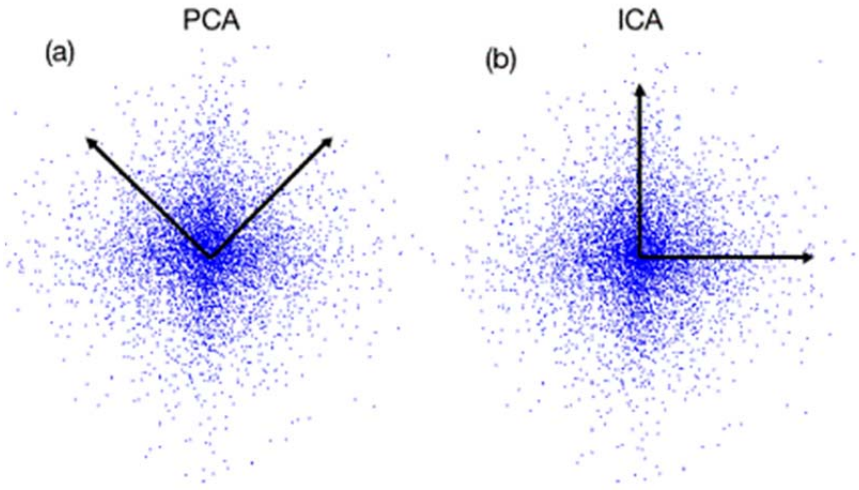
容易想象，该数据的主元是(P_1, P_2)或是旋转角θ。

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。

6. 其他参考文献

A *tutorial* on Principal Components Analysis LI Smith – 2002
A *Tutorial* on Principal Component Analysis J Shlens
http://www.cmlab.csie.ntu.edu.tw/~cyy/learning/tutorials/PCAMissingData.pdf
http://www.cad.zju.edu.cn/home/chenlu/pca.htm