# 第 7 章　数据降维

在很多应用问题中向量的维数会很高。处理高维向量不仅给算法带来了挑战，而且不便于可视化，另外还会面临维数灾难（这一概念将在第 14 章中介绍）的问题。降低向量的维数是数据分析中一种常用的手段。本章将介绍最经典的线性降维方法-主分量分析，以及非线性降维技术-流形学习算法。

## 7.1　主成分分析

在有些应用中向量的维数非常高。以图像数据为例，对于高度和宽度都为 100 像素的图像，如果将所有像素值拼接起来形成一个向量，这个向量的维数是 10000。一般情况下，向量的各个分量之间可能存在相关性。直接将向量送入机器学习算法中处理效率会很低，也影响算法的精度。为了可视化显示数据，我们也需要把向量变换到低维空间中。如何降低向量的维数并且去掉各个分量之间的相关性？主成分分析就是达到这种目的方法之一。

### 7.1.1　数据降维问题

主成分分析（principal component analysis，简称 PCA）[1]是一种数据降维和去除相关性的方法，它通过线性变换将向量投影到低维空间。对向量进行投影就是对向量左乘一个矩阵，得到结果向量：

$$\mathbf{y} = \mathbf{W}\mathbf{x}$$

在这里，结果向量的维数小于原始向量的维数。降维要确保的是在低维空间中的投影能很好的近似表达原始向量，即重构误差最小化。下图 7.1 是主成分投影示意图：

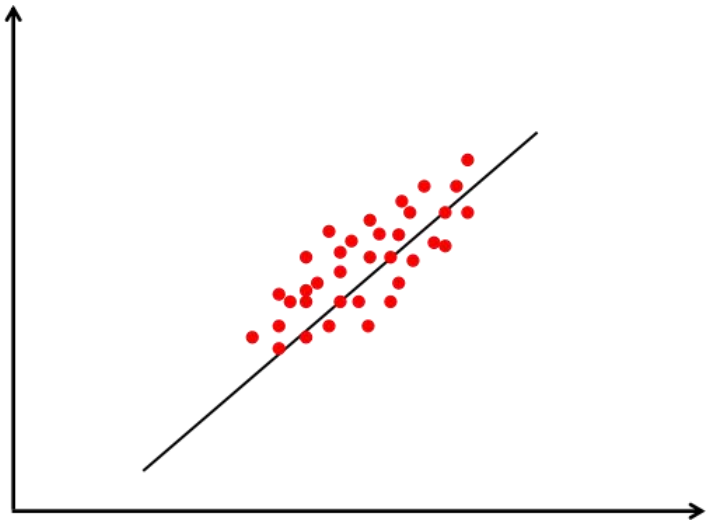

图 7.1　主成分投影示意图

在上图中样本用红色的点表示，倾斜的直线是它们的主要变化方向。将数据投影到这条

直线上即能完成数据的降维，把数据从 2 维降为 1 维。

### 7.1.2 计算投影矩阵

核心的问题是如何得到投影矩阵，和其他机器学习算法一样，它通过优化目标函数而得到。首先考虑最简单的情况，将向量投影到 1 维空间，然后推广到一般情况。假设有 $n$ 个 $d$ 维向量 $\mathbf{x}_i$，如果要用一个向量 $\mathbf{x}_0$ 来近似代替它们，这个向量取什么值的时候近似代替的误差最小？如果用均方误差作为标准，就是要最小化如下函数：

$$L\left(\mathbf{x}_0\right)=\sum_{i=1}^{n}\left\|\mathbf{x}_i-\mathbf{x}_0\right\|^2$$

显然问题的最优解是这些向量的均值：

$$\mathbf{m}=\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

证明很简单。为了求上面这个目标函数的极小值，对它求梯度并令梯度等于 $\mathbf{0}$，可以得到：

$$\nabla L\left(\mathbf{x}_0\right)=\sum_{i=1}^{n}2\left(\mathbf{x}_0-\mathbf{x}_i\right)=\mathbf{0}$$

解这个方程即可得到上面的结论。只用均值代表整个样本集过于简单，误差太大。作为改进，可以将每个向量表示成均值向量和另外一个向量的和：

$$\mathbf{x}_i=\mathbf{m}+a_i\mathbf{e}$$

其中 $\mathbf{e}$ 为单位向量，$a_i$ 是标量。上面这种表示相当于把向量投影到一维空间，坐标就是 $a_i$。当 $\mathbf{e}$ 和 $a_i$ 取什么值的时候，这种近似表达的误差最小？这相当于最小化如下误差函数：

$$L\left(a,\mathbf{e}\right)=\sum_{i=1}^{n}\left\|\mathbf{m}+a_i\mathbf{e}-\mathbf{x}_i\right\|^2$$

为了求这个函数的极小值，对 $a_i$ 求偏导数并令其为 0 可以得到：

$$2\mathbf{e}^{\mathrm{T}}\left(\mathbf{m}+a_i\mathbf{e}-\mathbf{x}_i\right)=0$$

变形后得到：

$$a_i\mathbf{e}^{\mathrm{T}}\mathbf{e}=\mathbf{e}^{\mathrm{T}}\left(\mathbf{x}_i-\mathbf{m}\right)$$

由于 $\mathbf{e}$ 是单位向量，因此 $\mathbf{e}^{\mathrm{T}}\mathbf{e}=1$，最后得到：

$$a_i=\mathbf{e}^{\mathrm{T}}\left(\mathbf{x}_i-\mathbf{m}\right)$$

这就是样本和均值的差对向量 $\mathbf{e}$ 做投影。现在的问题是 $\mathbf{e}$ 的值如何选确定。定义如下的散布矩阵：

$$\mathbf{S}=\sum_{i=1}^{n}\left(\mathbf{x}_i-\mathbf{m}\right)\left(\mathbf{x}_i-\mathbf{m}\right)^{\mathrm{T}}$$

这个矩阵是协方差矩阵的 $n$ 倍，协方差矩阵的计算公式为：

$$\mathbf{\Sigma}=\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{x}_i-\mathbf{m}\right)\left(\mathbf{x}_i-\mathbf{m}\right)^{\mathrm{T}}$$

将上面求得的 $a_i$ 代入目标函数中，得到只有变量 $\mathbf{e}$ 的函数：

$$\begin{aligned}
L(\mathbf{e}) &= \sum_{i=1}^{n}\left(\alpha_i\mathbf{e}+\mathbf{m}-\mathbf{x}_i\right)^{\mathrm{T}}\left(\alpha_i\mathbf{e}+\mathbf{m}-\mathbf{x}_i\right) \\
&= \sum_{i=1}^{n}\left(\left(\alpha_i\mathbf{e}\right)^{\mathrm{T}}\alpha_i\mathbf{e}+2\left(\alpha_i\mathbf{e}\right)^{\mathrm{T}}\left(\mathbf{m}-\mathbf{x}_i\right)+\left(\mathbf{m}-\mathbf{x}_i\right)^{\mathrm{T}}\left(\mathbf{m}-\mathbf{x}_i\right)\right) \\
&= \sum_{i=1}^{n}a_i^2-2\sum_{i=1}^{n}a_i^2+\sum_{i=1}^{n}\left(\mathbf{m}-\mathbf{x}_i\right)^{\mathrm{T}}\left(\mathbf{m}-\mathbf{x}_i\right) \\
&= -\sum_{i=1}^{n}\left(\mathbf{e}^{\mathrm{T}}\left(\mathbf{x}_i-\mathbf{m}\right)\right)^2+\sum_{i=1}^{n}\left(\mathbf{m}-\mathbf{x}_i\right)^{\mathrm{T}}\left(\mathbf{m}-\mathbf{x}_i\right) \\
&= -\sum_{i=1}^{n}\left(\mathbf{e}^{\mathrm{T}}\left(\mathbf{x}_i-\mathbf{m}\right)\left(\mathbf{x}_i-\mathbf{m}\right)^{\mathrm{T}}\mathbf{e}\right)+\sum_{i=1}^{n}\left(\mathbf{m}-\mathbf{x}_i\right)^{\mathrm{T}}\left(\mathbf{m}-\mathbf{x}_i\right) \\
&= -\mathbf{e}^{\mathrm{T}}\mathbf{S}\mathbf{e}+\sum_{i=1}^{n}\left(\mathbf{m}-\mathbf{x}_i\right)^{\mathrm{T}}\left(\mathbf{m}-\mathbf{x}_i\right)
\end{aligned}$$

上式的后半部分和 $\mathbf{e}$ 无关，由于 $\mathbf{e}$ 是单位向量，因此有 $\|\mathbf{e}\|=1$ 的约束，这可以写成 $\mathbf{e}^{\mathrm{T}}\mathbf{e}=1$。要求解的是一个带等式约束的极值问题，可以使用拉格朗日乘数法。构拉格朗日函数：

$$L(\mathbf{e},\lambda)=-\mathbf{e}^{\mathrm{T}}\mathbf{S}\mathbf{e}+\lambda\left(\mathbf{e}^{\mathrm{T}}\mathbf{e}-1\right)$$

对 $\mathbf{e}$ 求梯度并令其为 $\mathbf{0}$ 可以得到：

$$-2\mathbf{S}\mathbf{e}+2\lambda\mathbf{e}=\mathbf{0}$$

即：

$$\mathbf{S}\mathbf{e}=\lambda\mathbf{e}$$

$\lambda$ 就是散度矩阵的特征值，$\mathbf{e}$ 为它对应的特征向量，因此上面的最优化问题可以归结为矩阵的特征值和特征向量问题。矩阵 $\mathbf{S}$ 的所有特征向量给出了上面极值问题的所有极值点。矩阵 $\mathbf{S}$ 是实对称半正定矩阵，因此所有特征值非负。事实上，对于任意的非 $\mathbf{0}$ 向量 $\mathbf{x}$，有：

$$\begin{aligned}\mathbf{x}^{\mathrm{T}}\mathbf{S}\mathbf{x} &= \mathbf{x}^{\mathrm{T}}\left(\sum_{i=1}^{n}\left(\mathbf{x}_i-\mathbf{m}\right)\left(\mathbf{x}_i-\mathbf{m}\right)^{\mathrm{T}}\right)\mathbf{x}\\ &= \sum_{i=1}^{n}\mathbf{x}^{\mathrm{T}}\left(\mathbf{x}_i-\mathbf{m}\right)\left(\mathbf{x}_i-\mathbf{m}\right)^{\mathrm{T}}\mathbf{x}\\ &= \sum_{i=1}^{n}\left(\mathbf{x}^{\mathrm{T}}\left(\mathbf{x}_i-\mathbf{m}\right)\right)\left(\mathbf{x}^{\mathrm{T}}\left(\mathbf{x}_i-\mathbf{m}\right)\right)^{\mathrm{T}}\\ &\geq 0\end{aligned}$$

因此这个矩阵半正定。这里需要最大化 $\mathbf{e}^{\mathrm{T}}\mathbf{S}\mathbf{e}$ 的值，由于：

$$\mathbf{e}^{\mathrm{T}}\mathbf{S}\mathbf{e}=\lambda\mathbf{e}^{\mathrm{T}}\mathbf{e}=\lambda$$

因此 $\lambda$ 为散度矩阵最大的特征值时，$\mathbf{e}^{\mathrm{T}}\mathbf{S}\mathbf{e}$ 有极大值，目标函数取得极小值。将上述结论从一维推广到 $d'$ 维，每个向量可以表示成：

$$\mathbf{x}=\mathbf{m}+\sum_{i=1}^{d'}a_i\mathbf{e}_i$$

在这里 $\mathbf{e}_i$ 都是单位向量，并且相互正交，即寻找低维空间中的标准正交基。误差函数变成：

$$\sum_{i=1}^{n}\left\|\mathbf{m}+\sum_{j=1}^{d'}a_{ij}\mathbf{e}_j-\mathbf{x}_i\right\|^2$$

和一维情况类似，可以证明，使得该函数取最小值的 $\mathbf{e}_j$ 为散度矩阵最大的 $d'$ 个特征值对应的单位长度特征向量。即求解下面的优化问题：

$$\begin{aligned}&\min_{\mathbf{W}}-\mathrm{tr}\left(\mathbf{W}^{\mathrm{T}}\mathbf{S}\mathbf{W}\right)\\ &\mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}\end{aligned}$$

其中 tr 为矩阵的迹，$\mathbf{I}$ 为单位矩阵，该等式约束保证投影基向量是标准正交基。矩阵 $\mathbf{W}$ 的列 $\mathbf{e}_j$ 是要求解的基向量。散度矩阵是实对称矩阵，属于不同特征值的特征向量相互正交。前面已经证明这个矩阵半正定，特征值非负。这些特征向量构成一组基向量，我们可以用它们的线性组合来表达向量 $\mathbf{x}$。从另外一个角度来看，这种变换将协方差矩阵对角化，相当于去除了各分量之间的相关性。

从上面的推导过程我们可以得到计算投影矩阵的流程为：

1.计算样本集的均值向量。将所有向量减去均值，这称为白化。

2.计算样本集的协方差矩阵。

3.对方差矩阵进行特征值分解，得到所有特征值与特征向量。

4.将特征值从大到小排序，保留最大的一部分特征值对应的特征向量，以它们为行，形成投影矩阵。

具体保留多少个特征值由投影后的向量维数决定。使用协方差矩阵和使用散度矩阵是等

价的，因为后者是前者的 $n$ 倍，而矩阵 $\mathbf{A}$ 和 $n\mathbf{A}$ 有相同的特征向量。

### 7.1.3 向量降维

得到投影矩阵之后可以进行向量降维，将其投影到低维空间。向量投影的流程为：

1.将样本减掉均值向量。

2.左乘投影矩阵，得到降维后的向量。

### 7.1.4 向量重构

向量重构根据投影后的向量重构原始向量，与向量投影的作用和过程相反。向量重构的流程为：

1.输入向量左乘投影矩阵的转置矩阵。

2.加上均值向量，得到重构后的结果。

从上面的推导过程可以看到，在计算过程中没有使用样本标签值，因此主成分分析是一种无监督学习算法。除了标准算法之外它还有多个变种，如稀疏主成分分析，核主成分分析[2][8]，概率主分量分析等。

## 7.2 流形学习

主成分分析是一种线性降维技术，对于非线性数据具有局限性，而在实际应用中很多时候数据是非线性的。此时可以采用非线性降维技术，流形学习（manifold learning）是典型的代表。除此之外，第 9 章介绍的人工神经网络也能完成非线性降维任务。这些方法都使用非线性函数将原始输入向量 $\mathbf{x}$ 映射成更低维的向量 $\mathbf{y}$，向量 $\mathbf{y}$ 要保持 $\mathbf{x}$ 的某些信息：

$$\mathbf{y}=\phi(\mathbf{x})$$

流形是几何中的一个概念，它是高维空间中的几何结构，即空间中的点构成的集合，可以简单的将流形理解成二维空间的曲线，三维空间的曲面在更高维空间的推广。下图 7.2 是三维空间中的一个流形，这是一个卷曲面：

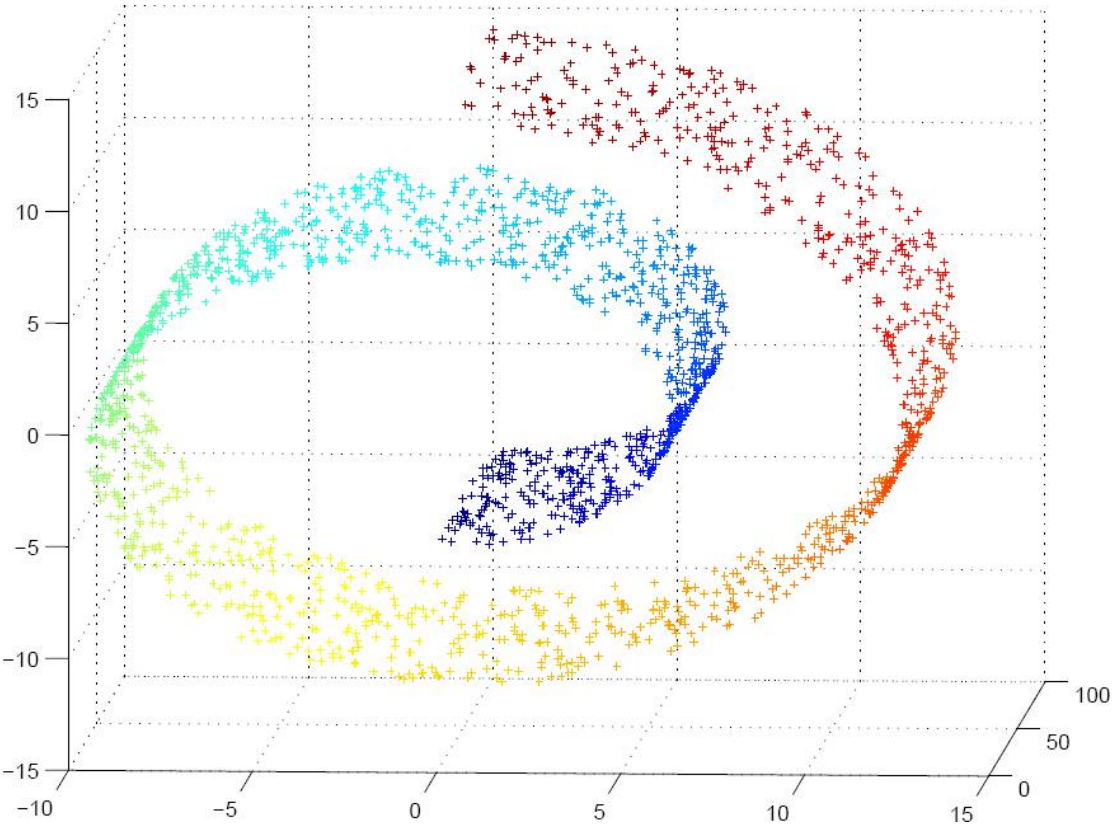


图 7.2　三维空间中的一个流形

很多应用问题的数据在高维空间中的分布具有某种几何形状，即位于一个低维的流形附近。例如同一个人的人脸图像向量在高维空间中可能是一个复杂的形状。流形学习假设原始数据在高维空间的分布位于某一更低维的流形上，基于这个假设来进行数据的分析。对于降维，要保证降维之后的数据同样满足与高维空间流形有关的几何约束关系。除此之外，流形学习还可以用实现聚类，分类以及回归算法，在后面各章中将会详细介绍。

假设有一个 $D$ 维空间中的流形 $M$ ，即 $M \subset \mathbb{R}^D$ ，流形学习降维要实现的是如下映射：

$$M \to \mathbb{R}^d$$

其中 $d \ll D$ 。即将 $D$ 维空间中流形 $M$ 上的点映射为 $d$ 维空间中的点。下面介绍几种典型的流形降维算法。

### 7.2.1　局部线性嵌入

局部线性嵌入[3]（locally linear embedding，简称 LLE）将高维数据投影到低维空间中，并保持数据点之间的局部线性关系。其核心思想是每个点都可以由与它相邻的多个点的线性组合来近似重构，投影到低维空间之后要保持这种线性重构关系，即有相同的重构系数，这也体现了它的名字。

假设数据集由 $n$ 个 $D$ 维向量 $\mathbf{x}_i$ 组成，它们分布在 $D$ 维空间中的一个流形附近。每个数据点和它的邻居位于或者接近于流形的一个局部线性片段上，即可以用邻居点的线性组合来重构，组合系数体现了局部面片的几何特性：

$$\mathbf{x}_i \approx \sum\nolimits_j w_{ij}\mathbf{x}_j$$

权重 $w_{ij}$ 为第 $j$ 个数据点对第 $i$ 个点的组合权重，这些点的线性组合被用来近似重构数据点 $i$。权重系数通过最小化下面的重构误差确定：

$$\min_{w_{ij}} \sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{j=1}^{n} w_{ij} \mathbf{x}_j \right\|^2$$

在这里还加上了两个约束条件：每个点只由它的邻居来重构，如果 $\mathbf{x}_j$ 不在 $\mathbf{x}_i$ 的邻居集合里则权重值为 0。另外限定权重矩阵的每一行元素之和为 1，即：

$$\sum_j w_{ij} = 1$$

这是一个带约束的优化问题，求解该问题可以得到权重系数。这一问题和主成分分析要求解的问题类似。可以证明，这个权重值对平移、旋转、缩放等几何变换具有不变性。

假设算法将向量从 $D$ 维空间的 $\mathbf{x}$ 映射为 $d$ 维空间的 $\mathbf{y}$。每个点在 $d$ 维空间中的坐标由下面的最优化问题确定：

$$\min_{\mathbf{y}_i} \sum_{i=1}^{n} \left\| \mathbf{y}_i - \sum_{j=1}^{n} w_{ij} \mathbf{y}_j \right\|^2$$

这里的权重和上一个优化问题的值相同，在前面已经得到，是已知量。这里优化的目标是 $\mathbf{y}_i$，此优化问题等价于求解稀疏矩阵的特征值问题。得到 $\mathbf{y}$ 之后，即完成了从 $D$ 维空间到 $d$ 维空间的非线性降维。

下图 7.3 为用 LLE 算法将手写数字图像投影到 3 维空间后的结果：

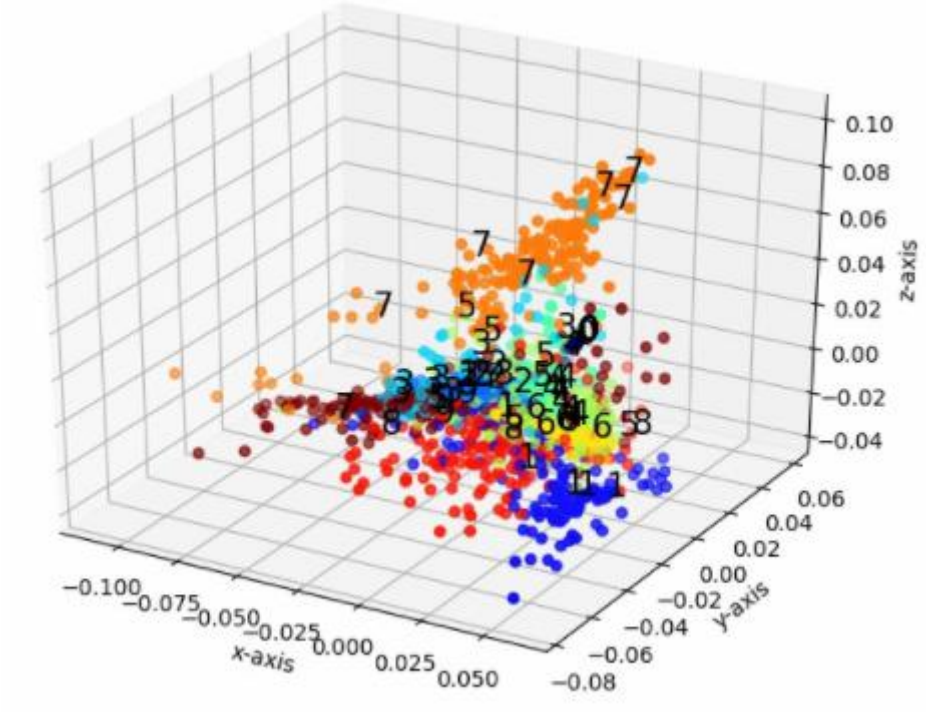


图 7.3　LLE 算法投影到 3 维空间后的结果

### 7.2.2 拉普拉斯特征映射

拉普拉斯特征映射[4]（简称 LE）是基于图论的方法。它为样本点构造带权重的图，然后计算图的拉普拉斯矩，对该矩阵进行特征值分解得到投影变换结果。这个结果对应于将样本点投影到低维空间，且保持样本点在高维空间中的相对距离信息。

图是离散数学和数据结构中的一个概念。一个图由顶点和边构成，任意两个节点之间可能都有边进行连接。边可以带有值信息，称为权重，例如两点之间的距离。下图 7.4 是一个简单的图：

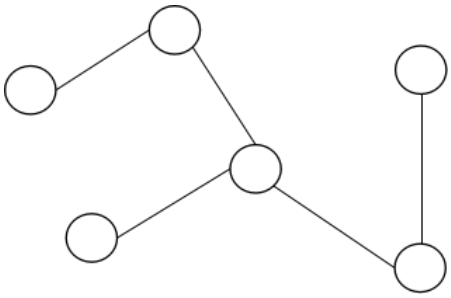

图 7.4　一个简单的无向图

图的边可以是有向的，也可以是无向的，前者称为有向图，后者称为无向图。可以将地图表示成一个图，每个地点是顶点，如果两个地点之间有路连接，则有一条边。如果这条路是单行线，则边是有向的，否则是无向的。

顶点的度定义为该顶点所关联的边的数量，对于有向图它还分为出度和入度，出度是指从一个顶点射出的边的数量，入度是连入一个节点的边的数量。无向图可以用三元组形式化的表示：

$$(V, E, w)$$

其中 $V$ 是顶点的集合， $E$ 是边的集合， $w$ 是边的权重函数，它为每条边赋予一个正的权重值。假设 $i$ 和 $j$ 为图的顶点， $w_{ij}$ 为边 $(i, j)$ 的权重，由它构成的矩阵 $\mathbf{W}$ 称为邻接矩阵。显然，无向图的邻接矩阵是一个对称矩阵。

拉普拉斯矩阵是图的一种矩阵表示，通过邻接矩阵而构造。定义顶点 $i$ 的带权重的度为与该节点相关的所有边的权重之和，即邻接矩阵每一行元素之和

$$d_i = \sum_j w_{ij}$$

定义矩阵 $\mathbf{D}$ 为一个对角矩阵，其主对角线元素为每个顶点带权重的度：

$$\begin{bmatrix} d_1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & d_n \end{bmatrix}$$

其中 $n$ 为图的顶点数。图的拉普拉斯矩阵定义为：

$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

无向图的拉普拉斯矩阵是一个对称矩阵，可以证明它是半正定矩阵。对于任意的非 $\mathbf{0}$ 向量 $\mathbf{f}$ ，有：

$$
\begin{aligned}
\mathbf{f}^{\mathrm{T}}\mathbf{L}\mathbf{f} &= \mathbf{f}^{\mathrm{T}}\mathbf{D}\mathbf{f} - \mathbf{f}^{\mathrm{T}}\mathbf{W}\mathbf{f} \\
&= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i f_j \\
&= \frac{1}{2}\left( \sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i f_j + \sum_{j=1}^{n} d_j f_j^2 \right) \\
&= \frac{1}{2}\left( \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i f_j + \sum_{j=1}^{n}\sum_{i=1}^{n} w_{ji} f_j^2 \right) \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left( w_{ij} f_i^2 - 2 w_{ij} f_i f_j + w_{ji} f_j^2 \right) \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\left( f_i - f_j \right)^2 \geq 0
\end{aligned}
$$

因此拉普拉矩阵半正定，另外还可以证明 0 是这个矩阵的特征值。

假设有一批样本点 $\mathbf{x}_1,...,\mathbf{x}_n$，它们是 $\mathbb{R}^D$ 空间的点，目标是将它们变换为更低维的 $\mathbb{R}^d$ 空间中的点 $\mathbf{y}_1,...,\mathbf{y}_n$，其中 $d \ll D$。在这里假设 $\mathrm{x}_1,...,\mathrm{x}_k \in M$，其中 $M$ 为嵌入 $\mathbb{R}^l$ 空间中的一个流形。

算法为样本点构造加权图，图的节点是每一个样本点，边为每个节点与它的邻居节点之间的相似度，每个节点只和它的邻居有连接关系。算法的目标是投影之后保持在高维空间中的距离关系，假设投影后到低维空间后的坐标为 $\mathbf{y}$，它通过最小化如下目标函数实现

$$
\min_{\mathbf{y}_i} \sum_{i=1}^{n}\sum_{j=1}^{n} \left\| \mathbf{y}_i - \mathbf{y}_j \right\|^2 w_{ij}
$$

此函数的含义是如果样本 $\mathbf{x}_i$ 和 $\mathbf{x}_j$ 的相似度很高即在高维空间中距离很近，则它们之间的边的权重 $w_{ij}$ 很大，因此投影到低维空间中后两个点要离得很近，即 $\mathbf{y}_i$ 和 $\mathbf{y}_j$ 要很接近，否则会产生一大个的损失值。根据上面证明拉普拉斯矩阵半正定时的结论，求解该目标函数等价于下面的优化问题

$$
\begin{aligned}
&\min_{\mathbf{Y}} \mathrm{tr}\left( \mathbf{Y}^{\mathrm{T}}\mathbf{L}\mathbf{Y} \right) \\
&\mathbf{Y}^{\mathrm{T}}\mathbf{D}\mathbf{Y} = \mathbf{I}
\end{aligned}
$$

其中 $\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_d \end{bmatrix}$ 为投影后的坐标按列构成的矩阵，这里加上了等式约束条件以消掉 $\mathbf{y}$ 的冗余，选用矩阵 $\mathbf{D}$ 来构造等式约束是因为其主对角线元素即节点的加权度反映了图的每个节点的重要性。通过拉格朗日乘数法可以证明，这个问题的最优解是如下广义值问题的解

$$
\mathbf{L}\mathbf{f} = \lambda\mathbf{D}\mathbf{f}
$$

可以证明这个广义特征值问题的所有特征值非负。最优解为这个广义特征值问题除去 0 之外的最小的 $d$ 个广义特征值对应的特征向量，这些向量按照列构成矩阵 $\mathbf{Y}$。下面给出拉

普拉斯特征映射算法的流程。

算法的第一步是构造图的节点的邻接关系。如果样本点 $\mathbf{x}_i$ 和样本点 $\mathbf{x}_j$ 的距离很近，则为图的节点 $i$ 和节点 $j$ 建立一条边。判断两个样本点是否解接近的方法有两种。第一种是计算二者的欧氏距离，如果距离小于某一值 $\varepsilon$ 则认为两个样本很接近

$$\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2<\varepsilon$$

其中 $\varepsilon$ 是一个人工设定的阈值。第二种方法是使用近邻规则，如果节点 $i$ 在节点 $j$ 最近的 $n$ 个邻居节点的集合中，或者节点 $j$ 在节点 $i$ 最近的 $n$ 个邻居节点的集合中，则认为二者距离很近。

第二步是计算边的权重，在这里也有两种选择。第一种方法为，如果节点 $i$ 和节点 $j$ 是联通的，则它们之间的边的权重为：

$$w_{ij}=\exp\left(-\frac{\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2}{t}\right)$$

否则 $w_{ij}=0$。其中 $t$ 是一个人工设定的大于 0 的实数。第二种方式是如果节点 $i$ 和节点 $j$ 是联通的则它们之间的边的权重为 1，否则为 0。

第三步是特征映射。假设构造的图是联通的，即任何两个节点之间都有路径可达，如果不联通，则算法分别作用于每个联通分量上。根据前面构造的图计算它的拉普拉斯矩阵，然后求解如下广义特征值和特征向量问题

$$\mathbf{Lf}=\lambda\mathbf{Df}$$

前面已经证明拉普拉斯矩阵半正定，根据它以及矩阵 $\mathbf{D}$ 的定义，可以证明广义特征值非负。假设 $\mathbf{f}_0,...,\mathbf{f}_{d-1}$ 是这个广义特征值问题的解，它们按照特征值的大小升序排列，即

$$0=\lambda_0\le...\le\lambda_{d-1}$$

去掉值为 0 的特征值 $\lambda_0$，剩下的前 $d$ 个特征值对应的特征向量即为投影结果

$$\mathbf{x}_i\to\left(\mathbf{f}_1(i),...,\mathbf{f}_d(i)\right)$$

图 7.5 是拉普拉斯特征映射对三维数据进行降维的一个例子

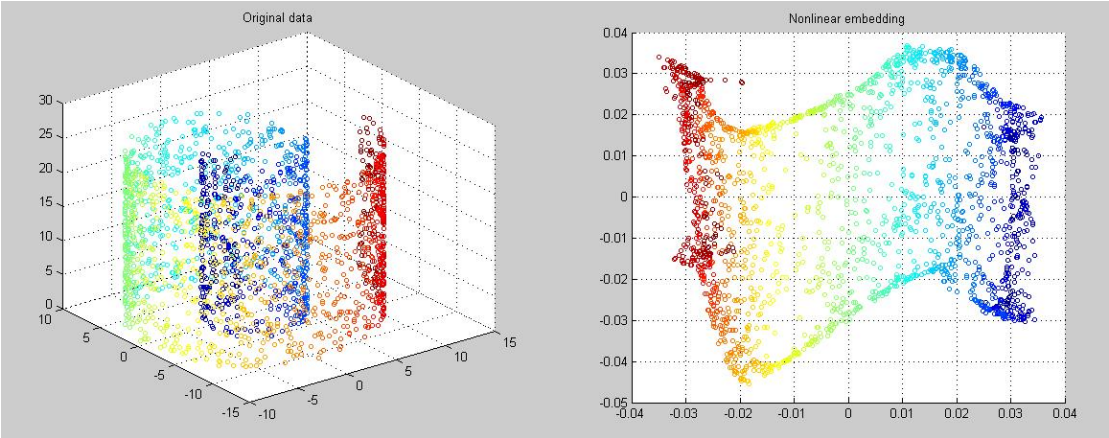

图 7.5　拉普拉斯特征映射对三维数据降维

上图中左侧为三维空间中的样本分布，右图为降维后的结果。这种变换起到的效果大致上相当于把三维空间中的曲面拉平之后铺到二维平面上，保持了样本在流形上的距离关系。

### 7.2.3　局部保持投影

局部保持投影（Locality preserving projections，简称 LPP）[5]通过最好的保持一个数据集的邻居结构信息来构造投影映射，其思路和拉普拉斯特征映射类似，区别在于不是直接得到投影结果而是求解投影矩阵。

假设有样本集 $\mathbf{x}_1,...,\mathbf{x}_m$，它们是 $\mathbb{R}^n$ 空间中的向量。这里的目标是寻找一个变换矩阵 $\mathbf{A}$，将这些样本点映射到更低维的 $\mathbb{R}^l$ 空间，得到向量 $\mathbf{y}_1,...,\mathbf{y}_m$，使得 $\mathbf{y}_i$ 能够代表 $\mathbf{x}_i$，其中 $l \ll n$：

$$\mathbf{y}_i = \mathbf{A}^{\mathrm{T}}\mathbf{x}_i$$

假设 $\mathbf{x}_1,...,\mathbf{x}_n \in M$，其中 $M$ 是 $\mathbb{R}^l$ 空间中的一个流形。

目标函数与拉普拉斯特征映射相同，定义为

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\left\|\mathbf{y}_i-\mathbf{y}_j\right\|^2 w_{ij} = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\left\|\mathbf{A}^{\mathrm{T}}\mathbf{x}_i-\mathbf{A}^{\mathrm{T}}\mathbf{x}_j\right\|^2 w_{ij}$$

所有矩阵的定义与拉普拉斯特征映射相同。投影变换矩阵为

$$\mathbf{A}=\begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & ... & \mathbf{a}_d\end{bmatrix}$$

即

$$\mathbf{y}=\mathbf{A}^{\mathrm{T}}\mathbf{x}$$

假设矩阵 $\mathbf{X}$ 为所有样本按照列构成的矩阵。根据与拉普拉斯特征映射类似的推导，这等价于求解下面的问题

$$\min_{\mathbf{a}} \mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{L}\mathbf{X}^{\mathrm{T}}\mathbf{a}$$
$$\mathbf{a}^{\mathrm{T}}\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}}\mathbf{a}=1$$

通过拉格朗日乘数法可以证明，此问题的最优解是下面广义特征值问题的解

$$\mathbf{X}\mathbf{L}\mathbf{X}^{\mathrm{T}}\mathbf{a}=\lambda\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}}\mathbf{a}$$

可以将这种做法推广到高维的情况。下面给出局部保持投影算法的流程。

算法的第一步是根据样本构造图，这和拉普拉斯特征映射的做法相同，包括确定两个顶点是否连通以及计算边的权重，在这里不再重复介绍。

第二步是特征映射，计算如下广义特征值问题

$$\mathbf{X}\mathbf{L}\mathbf{X}^{\mathrm{T}}\mathbf{a}=\lambda\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}}\mathbf{a}$$

假设上面广义特征向量问题的解为 $\mathbf{a}_1,...,\mathbf{a}_d$，它们对应的特征值满足

$$\lambda_1 < .... < \lambda_l$$

要寻找的降维变换矩阵为：

$$\mathbf{x}_i \to \mathbf{y}_i = \mathbf{A}^{\mathrm{T}}\mathbf{x}_i, \mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & .. & \mathbf{a}_l \end{bmatrix}$$

$\mathbf{A}$ 是一个 $n \times l$ 的矩阵。对向量左乘矩阵 $\mathbf{A}^{\mathrm{T}}$ 即可完成数据的降维。

### 7.2.4 等距映射

等距映射（Isomap）[6]使用了微分几何中测地线的思想，它希望数据在向低维空间映射之后能够保持流形上的测地线距离。

测地线源自于大地测量学，是地球上任意两点之间在球面上的最短路径。在三维空间中两点之间的最短距离是它们之间线段的长度，但如果要沿着地球表面走，最短距离就是测地线的长度，因为我们不能从地球内部穿过去。这里的测地线就是球面上两点之间大圆上劣弧的长度。算法计算任意两个样本之间的测地距离，然后根据这个距离构造距离矩阵。最后通过距离矩阵求解优化问题完成数据的降维，降维之后的数据保留了原始数据点之间的距离信息。

在这里测地线距离通过图构造，是图的两个节点之间的最短距离。算法的第一步构造样本集的邻居图，这和前面介绍的两种方法相同。如果两个数据点之间的距离小于指定阈值或者其中一个节点在另外一个节点的邻居集合中，则两个节点是联通的。假设有 $N$ 个样本，则邻居图有 $N$ 个节点。邻居图的节点 $i$ 和 $j$ 之间边的权重为它们之间的距离 $w_{ij}$，距离的计算公式可以有多种选择。

第二步计算图中任意两点之间的最短路径长度，可以通过经典的 Dijkstra 算法实现。假设最短路径长度为 $d_G\left(i,j\right)$，由它构造如下矩阵：

$$\mathbf{D}_G = \left\{d_G\left(i,j\right)\right\}$$

其元素是所有节点对之间的最短路径长度。算法的第三步根据矩阵 $\mathrm{D}_G$ 构造 $d$ 维嵌入 $\mathbf{y}$，这通过求解如下最优化问题实现：

$$\min_{\mathbf{y}} \sum_{i=1}^{N}\sum_{j=1}^{N}\left(d_G\left(i,j\right) - \left\|\mathbf{y}_i - \mathbf{y}_j\right\|\right)^2$$

这个问题的解 $\mathbf{y}$ 即为降维之后的向量。这个目标函数的意义是向量降维之后任意两点之间的距离要尽量的接近在原始空间中这两点之间的最短路径长度，因此可以认为降维尽量保留了数据点之间的测地距离信息。下图 7.6 为等距映射将手写数字图像投影到 3 维空间后的结果：

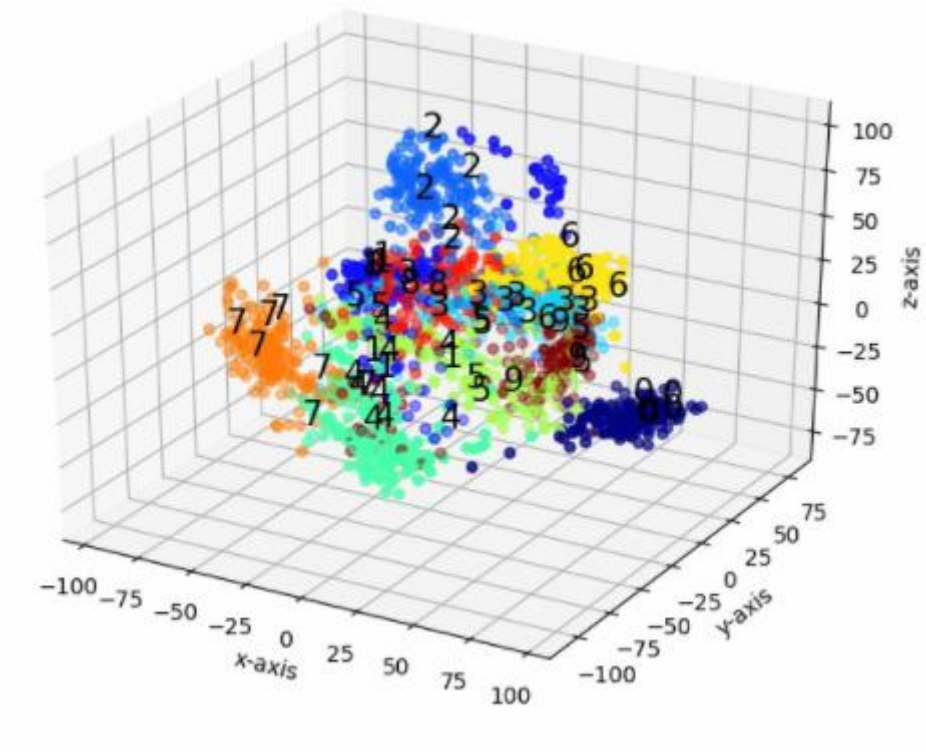


图 7.6　等距映射的投影结果

### 7.2.5　随机近邻嵌入

随机近邻嵌入（stochastic neighbor embedding，简称 SNE）[10]基于如下思想：在高维空间中距离很近的点投影到低维空间中之后也要保持这种近邻关系，在这里距离通过概率体现。假设在高维空间中有两个点样本点 $\mathbf{x}_i$ 和 $\mathbf{x}_j$，$\mathbf{x}_j$ 以 $p_{j|i}$ 的概率作为 $\mathbf{x}_i$ 的邻居，将样本之间的欧氏距离转化成概率值，借助于正态分布，此概率的计算公式为

$$p_{j|i}=\frac{\exp\left(-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-\left\|\mathbf{x}_i-\mathbf{x}_k\right\|^2/2\sigma_i^2\right)}$$

其中 $\sigma_i$ 表示以 $\mathbf{x}_i$ 为中心的正态分布的标准差，这个概率的计算公式类似于第 11.4 节将要介绍的 softmax 回归。上式中除以分母的值是为了将所有值归一化成概率。由于不关心一个点与它自身的相似度，因此 $p_{j|i}=0$。投影到低维空间之后仍然要保持这个概率关系。假设 $\mathbf{x}_i$ 和 $\mathbf{x}_j$ 投影之后对应的点为 $\mathbf{y}_i$ 和 $\mathbf{y}_j$，在低维空间中对应的近邻概率记为 $q_{j|i}$，计算公式与上面的相同，但标准差统一设为 $1/\sqrt{2}$，即

$$q_{j|i}=\frac{\exp\left(-\left\|\mathbf{y}_i-\mathbf{y}_j\right\|^2\right)}{\sum_{k\neq i}\exp\left(-\left\|\mathbf{y}_i-\mathbf{y}_k\right\|^2\right)}$$

上面定义的是点 $\mathbf{x}_i$ 与它的一个邻居点的概率关系，如果考虑所有其他点，这些概率值构成一个离散型概率分布 $P_i$，是所有样本点成为 $\mathbf{x}_i$ 的邻居的概率。在低维空间中对应的概率分布为 $Q_i$，投影的目标是这两个概率分布尽可能接近，因此需要衡量两个概率分布之间的相似度或距离。在机器学习中一般用 KL（Kullback-Leibler）散度衡量两个概率分布之间的距离，在生成对抗网络、变分自动编码器中都有它的应用。假设 $x$ 为离散型随机变量，$p(x)$ 和 $q(x)$ 是它的两个概率分布，KL 散度定义为

$$KL\left(p\|q\right)=\sum_{x}p\left(x\right)\log\frac{p\left(x\right)}{q\left(x\right)}$$

KL 散度不具有对称性，即一般情况下 $KL\left(p\|q\right)\neq KL\left(q\|p\right)$。KL 散度是非负的，如果两个概率分布完全相同，有极小值 0。对于连续型随机变量，则将求和换成定积分。

由此得到投影的目标为最小化如下函数

$$L\left(\mathbf{y}_i\right)=\sum_{i=1}^{l}KL\left(P_i\left|Q_i\right.\right)=\sum_{i=1}^{l}\sum_{j=1}^{l}p_{j|i}\log\frac{p_{j|i}}{q_{j|i}}$$

这里对所有样本点的 KL 散度求和，$l$ 为样本数。把概率的计算公式代入 KL 散度，可以将目标函数写成所有 $\mathbf{y}_i$ 的函数。目标函数对 $\mathbf{y}_i$ 的梯度为

$$\nabla_{\mathbf{y}_i}L=2\sum_{j}\left(\mathbf{y}_i-\mathbf{y}_j\right)\left(p_{i|j}-q_{i|j}+p_{j|i}-q_{j|i}\right)$$

计算出梯度之后可用梯度下降法迭代，得到的最优 $\mathbf{y}_i$ 值即为 $\mathbf{x}_i$ 投影后的结果。

### 7.2.5 t-分布随机近邻嵌入

虽然 SNE 有较好的效果，但训练时难以优化，而且容易导致拥挤问题（crowding problem）。t 分布随机近邻嵌入（t-distributed Stochastic Neighbor Embedding，简称 t-SNE）[11]是对 SNE 的改进。t-SNE 采用了对称的概率计算公式，另外在低维空间中计算样本点之间的概率时使用 t 分布代替了正态分布。

在 SNE 中 $p_{i|j}$ 和 $p_{j|i}$ 是不相等的，因此概率值不对称。可以用两个样本点的联合概率替代它们之间的条件概率解决此问题。在高维空间中两个样本点的联合概率定义为

$$p_{ij}=\frac{\exp\left(-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2/2\sigma^2\right)}{\sum_{k\neq l}\exp\left(-\left\|\mathbf{x}_k-\mathbf{x}_l\right\|^2/2\sigma^2\right)}$$

显然这个定义是对称的，即 $p_{ij}=p_{ji}$。同样的，低维空间中两个点的联合概率为

$$q_{ij}=\frac{\exp\left(-\left\|\mathbf{y}_i-\mathbf{y}_j\right\|^2\right)}{\sum_{k\neq i}\exp\left(-\left\|\mathbf{y}_k-\mathbf{y}_l\right\|^2\right)}$$

目标函数采用 KL 散度，定义为

$$L\left(\mathbf{y}_i\right)=D_{\mathrm{KL}}\left(P\middle|Q\right)=\sum_{i=1}^{l}\sum_{j=1}^{l}p_{ij}\log\frac{p_{ij}}{q_{ij}}$$

但这样定义联合概率会存在异常值问题。如果某一个样本 $\mathbf{x}_i$ 是异常点即离其他点很远，则所有的 $\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2$ 都很大，因此与 $\mathbf{x}_i$ 有关的 $p_{ij}$ 很小，从而导致低维空间中的 $\mathbf{y}_i$ 对目标函数影响很小。解决方法是重新定义高维空间中的联合概率，具体为

$$p_{ij}=\frac{p_{j|i}+p_{i|j}}{2n}$$

这样能确保对所有的 $\mathbf{x}_i$ 有 $\sum_j p_{ij}>\frac{1}{2n}$，因此每个样本点都对目标函数有显著的贡献。目标函数对 $\mathbf{y}_i$ 的梯度为

$$\nabla_{\mathbf{y}_i}L=4\sum_j\left(p_{ij}-q_{ij}\right)\left(\mathbf{y}_i-\mathbf{y}_j\right)$$

这种方法称为对称 SNE。

对称 SNE 虽然对 SNE 做了改进，但还存在拥挤问题，各类样本降维后聚集在一起而缺乏区分度。解决方法是用 t 分布替代高斯分布，计算低维空间中的概率值。相比于正态分布，t 分布更长尾。如果在低维空间中使用 t 分布，则在高维空间中距离近的点，在低维空间中距离也要近；但在高维空间中距离远的点，在低维空间中距离要更远。因此可以有效的拉大各个类之间的距离。使用 t 分布之后，低维空间中的概率计算公式为

$$q_{ij}=\frac{\left(1+\left\|\mathbf{y}_i-\mathbf{y}_j\right\|^2\right)^{-1}}{\sum_{k\neq l}\left(1+\left\|\mathbf{y}_k-\mathbf{y}_l\right\|^2\right)^{-1}}$$

目标函数同样采用 KL 散度。目标函数对的 $\mathbf{y}_i$ 梯度为

$$\nabla_{\mathbf{y}_i}L=4\sum_j\left(p_{ij}-q_{ij}\right)\left(\mathbf{y}_i-\mathbf{y}_j\right)\left(1+\left\|\mathbf{y}_i-\mathbf{y}_j\right\|^2\right)^{-1}$$

同样的，求得梯度之后可以用梯度下降法进行迭代从而得到 $\mathbf{y}_i$ 的最优解。

## 7.3 实验程序

下面通过实验程序介绍主成分分析的使用。程序基于 sklearn，使用 iris 数据集。程序将 4 维的特征向量投影到 2 维平面，根据样本的标签值将样本点显示成不同的颜色。在创建和

训练 PCA 模型时需要指定降维后的维数，这里为 2。程序源代码可以通过左侧二维码获取。

程序运行结果如下图 7.7 所示。

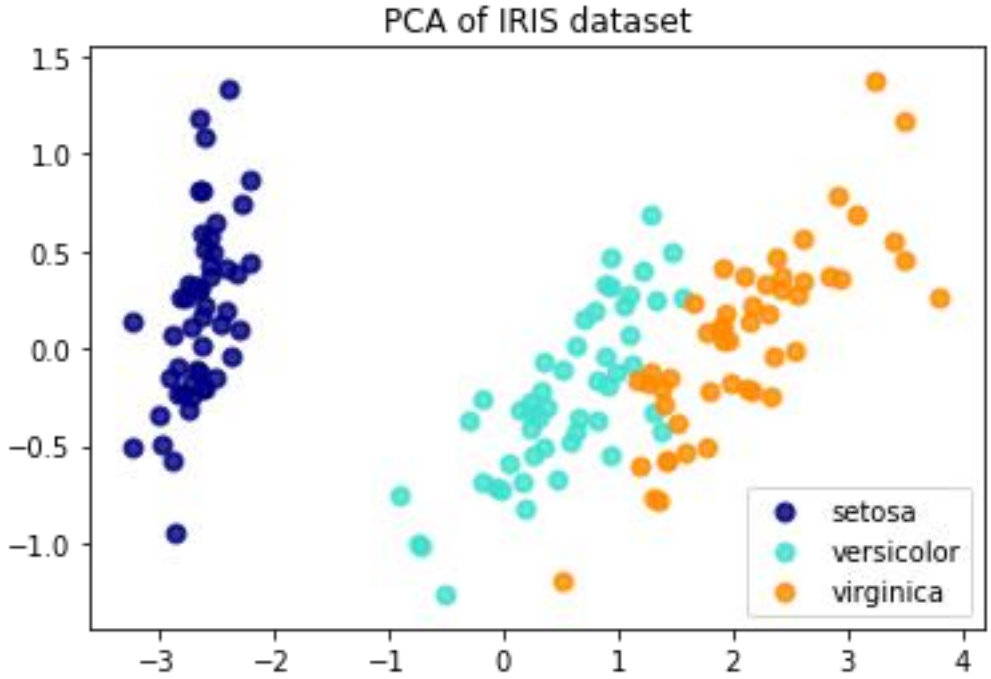


图 7.7　PCA 对 iris 数据集的降维结果

除 PCA 之外，sklearn 还支持本章介绍的其他降维算法。

## 7.4　应用

主成分分析被大量的用于科学与工程数据分析中需要数据降维的地方，是一种通用性非常好的算法。在人脸识别早期它被直接用于人脸识别问题[7]，将在下一章中详细介绍。流形学习在高维复杂数据集上得到了更好的表现，如人脸图像[9]和其他图像的分类问题。

## 参 考 文 献

[1] Ian T. Jolliffe. Principal Component Analysis. Springer Verlag, New York, 1986.

[2] Sebastian Mika,Bernhard Scholkopf,Alexander J Smola,Klausrobert Muller,Matthias Scholz Gun. Kernel PCA and de-noising in feature spaces. neural information processing systems, 1999.

[3] Roweis, Sam T and Saul, Lawrence K. Nonlinear dimensionality reduction by locally linear embedding. Science, 290(5500). 2000: 2323-2326.

[4] Belkin, Mikhail and Niyogi, Partha. Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation. 15(6). 2003:1373-1396.

[5] He Xiaofei and Niyogi, Partha. Locality preserving projections. NIPS. 2003:234-241.

[6] Tenenbaum, Joshua B and De Silva, Vin and Langford, John C. A global geometric framework for nonlinear dimensionality reduction. Science, 290(5500). 2000: 2319-2323.

[7] Matthew Turk,Alex Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 1991

[8] Scholkopf, B.,Smola,A.,Mulller,K.-P. Nonlinear component analysis as a kernel eigenvalue problem.

Neural Computation, 10(5), 1299-1319, 1998.

[9] He, Xiaofei, et al. Face recognition using Laplacianfaces. Pattern Analysis and Machine Intelligence, IEEE Transactions on 27.3 (2005): 328-340.

[10] Geoffrey E Hinton, Sam T Roweis. Stochastic Neighbor Embedding. neural information processing systems, 2002.

[11] Maaten, Laurens van der, and Geoffrey Hinton. Visualizing data using t-SNE. Journal of machine learning research 9.Nov (2008): 2579-2605.