CS131 Lecture12：人脸识别与降维

by：斯坦福大学计算机科学系

github: https://github.com/zhaoxiongjun/CS131_notes_zh-CN （包含中英文版课件及相关课程视频）

1 概述和动机

1.1 概述

降维是减少分析或预测模型中使用的特征数量的过程。这提高了基于计算机视觉和机器学习的方法的性能，使我们能够以更有效的方式表示数据。有几种常用的降维方法。本课程的两种主要方法是奇异值分解（SVD）和主成分分析（PCA）。

1.2 动机

降维有以下优点：

1. 可以实现简化计算。在许多数据集中，大多数变量都可以用相对较少的输入变量及其线性组合来解释。针对这些关键部件采用降维的方法，可以在不损失数据粒度的前提下降低计算成本。
2. 减少“维度诅咒”的影响。在第 11 课中，我们了解到，当我们增加一个特征空间的维度时，用相同密度“填充”该空间所需的数据点数量会呈指数级爆炸。也就是说，机器学习算法中使用的维数越多，学习所需的示例越多，算法分析相同数量的数据点所需的时间越长。通过降维，我们可以减轻这种“维数诅咒”的影响。
3. 压缩数据。通过降低图像的维数，我们可以大大降低数据存储需求。
   在这种情况下，使用 svd 这样的程序，每个数据点的计算成本可以减少许多数量级。

2 奇异值分解

2.1 概述

直观地说，奇异值分解（svd）是一个过程，它允许一个人在一个新的子特征空间中表示数据，从而捕获数据中的大多数变化；这是通过“旋转原始特征空间的轴”来实现的，以形成线性梳状的新轴。原始轴/特征（如客户的年龄、收入、性别等）的确定。这些“新轴”是有用的，因为它们根据每个方向对数据方差的贡献，系统地分解数据点的方差（数据点分布的范围有多广）：
这个过程的结果是在特征空间中按从最大方差到最小的顺序排列的“方向”列表。方差最大的方向被称为“主成分”（数据变化的方向）；通过关注这些维度上的数据分布，可以捕获原始特征空间中表示的大部分信息，而无需处理原始空间中的大量维度（但请参见下面关于特征选择和维度缩减之间的区别）。

2.2 奇异值分解的技术细节

SVD 将任何矩阵 A 表示为三个矩阵的乘积：$A = U\sum V^T$，其中 U 和 V 是旋转矩阵，$\sum$是对角缩放矩阵。例如：

$$\overset{U}{\begin{bmatrix} -.39 & -.92 \\ -.92 & .39 \end{bmatrix}} \times \overset{\Sigma}{\begin{bmatrix} 9.51 & 0 & 0 \\ 0 & .77 & 0 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} = \overset{A}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}}$$

对于许多读者来说，会写代码：[U，S，V]= numpy.linalg.svd（A）来提取 svd 值就足够了。然而，如何计算 svd 的基础对于以后的主题是有用的。计算机通常通过以下步骤计算 SVD：

- 计算AA^T的特征向量。这些向量是 U 的列向量。特征值的平方根是奇异值。
- 计算A^TA的特征向量。这些向量是 V 的列向量。

由于奇异值分解依赖于特征向量计算，而特征向量计算通常是快速的，因此奇异值分解可以很快形成，即使对于大型矩阵也是如此。

2.3 奇异值分解的应用

奇异值分解最常用的应用之一是矩阵逆的计算。如果任意矩阵 A 可以通过：$A = U\sum V^T$分解，则 a 的逆可以定义为：$A^+ = V^T\sum^{-1}U$。虽然该逆是一个近似值，但它允许计算许多非平方矩阵的逆。Macausland(2014)讨论了这个逆的数学基础，它以其创造者命名为 Moore Penrose 逆[3]。不出所料，利用这种方法可以解决许多矩阵问题。

奇异值分解也可以用来计算矩阵的主成分。主成分在各种数据分析和机器学习例程中被大量使用，因此 svd 通常是许多程序中的核心例程。

3 主成分分析

3.1 主成分是什么?

继续上面的 SVD 示例，注意 U 的第 1 列按∑的第一个值缩放。

$$\overset{U\Sigma}{\begin{bmatrix} -3.67 & -.71 & 0 \\ -8.8 & .30 & 0 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} \qquad \overset{A_{partial}}{\begin{bmatrix} 1.6 & 2.1 & 2.6 \\ 3.8 & 5.0 & 6.2 \end{bmatrix}}$$

然后，得到的向量 U∑按V^T的第 1 行缩放，以产生对 A 的列的贡献，该贡献被表示为部分。

$$\overset{U\Sigma}{\begin{bmatrix} -3.67 & -.71 & 0 \\ -8.8 & .30 & 0 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} \qquad \overset{A_{partial}}{\begin{bmatrix} 1.6 & 2.1 & 2.6 \\ 3.8 & 5.0 & 6.2 \end{bmatrix}}$$

$$+ \overset{U\Sigma}{\begin{bmatrix} -3.67 & -.71 & 0 \\ -8.8 & .30 & 0 \end{bmatrix}} \times \overset{V^T}{\begin{bmatrix} -.42 & -.57 & -.70 \\ .81 & .11 & -.58 \\ .41 & -.82 & .41 \end{bmatrix}} \qquad \overset{A_{partial}}{\begin{bmatrix} -.6 & -.1 & .4 \\ .2 & 0 & -.2 \end{bmatrix}}$$

$$= \qquad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

在这个过程中，我们将矩阵 A 建立为 U 的列的线性组合。如上所示，如果我们使用 U 的所有列，我们将完美地重建 A。然而，在实际数据中，我们只能使用 U 的前几列来获得 A 的良好近似值。这是由于的属性引起的。∑是一个对角线矩阵，其中最大的值在左上角，而对角线上的其他值随着您向右移动而减小。因此，u 的前几列对 A 的贡献最大。U 的前几列称为主成分。

然而，并不是所有的矩阵都能像前面的例子那样容易地压缩。评价其可行性的一种方法是主成分分析法。从一个高层次的角度来看，我们想看看是否有可能删除对最终图像贡献不大的维度。我们通过分析协方差矩阵来实现这一点。虽然协方差的值没有那么重要，但协方差的符号确实重要，正的表示正相关，负的表示负相关。协方差为零表示二者相互独立。

3.2 进行主成分分析

可以使用 sklearn 包：sklearn.decomposition.pca 执行主成分分析。然而，前面提到 SVD 可以用来进行主成分分析。非正规方法概述如下：

1. 将数据格式化为 m*n 矩阵，其中 m 表示样本数，p 表示与单个样本对应的特征或变量数。
2. 通过减去平均值并除以 X 中每列（特征）的标准差，使矩阵 X 居中
3. 使用 svd 对角化 X
4. 特征向量是主方向，轴上的投影是分量。这最终意味着我们要计算 XV
5. 由于 V 特征向量,因此正交,$\mathrm{XV} = \mathrm{U}\sum V^T V = US$
6. （5）意味着我们只需要 U 的列，这两个矩阵都由 SVD 表示。

3.3 主成分的应用

PCA 在图像压缩中得到了广泛的应用。图像矩阵中捕获的大部分信息可以使用低阶矩阵来提取。这使得大图像可以压缩而不会有明显的质量损失。仅使用前 16 个主分量的基于 PCA 的压缩的示例如下所示：

图 1:左:原始图像,右:压缩图像

只要前 16 个主分量，就可以重建与原始图像非常相似的图像。由于上述图像的 PCA 使用的尺寸而导致的相对误差如下所示：

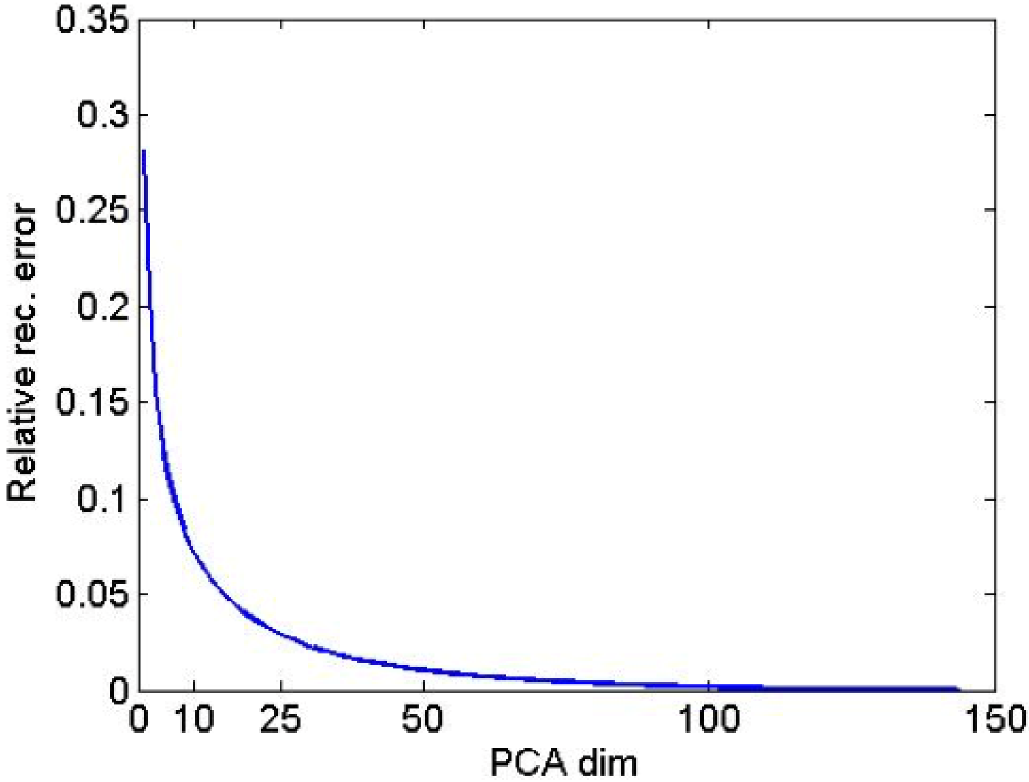

图 2：相对误差作为主成分分析维度的函数

网络搜索引擎也利用主成分分析。互联网上有数以十亿计的页面可能与提供的搜索短语有着非平凡的关系。谷歌（google）、必应（bing）和雅虎（yahoo）等公司通常只考虑这个搜索矩阵的一小部分来缩小搜索空间，而这个搜索矩阵可以使用 PCA 提取[2]。这对于及时有效的搜索是至关重要的，它体现了 SVD 的力量。