Dimensionality Reduction

Big Data

一个矩形的数据集有两个维度，observations 的数量 n，和 predictors 的数量 p (简单说，n 是行数，p 是列数)。这两者都可以在将问题定义为大数据问题方面发挥作用。

In [11]:
```
print(nyc_cab_df.shape)
nyc_cab_df.head()
```

```
(1873671, 30)
```

Out[11]:

	AWND	Base	Day	Dropoff_latitude	Dropoff_longitude	Ehail_fee	Extra	Fare_amount	Lpep_dropoff_datetime	MTA_tax	...	TMIN	Tip_amount	Tolls_amou
0	4.7	B02512	1	NaN	NaN	NaN	NaN	33.863498	2014-04-01 00:24:00	NaN	...	39	NaN	Na
1	4.7	B02512	1	NaN	NaN	NaN	NaN	19.022892	2014-04-01 00:29:00	NaN	...	39	NaN	Na
2	4.7	B02512	1	NaN	NaN	NaN	NaN	25.498981	2014-04-01 00:34:00	NaN	...	39	NaN	Na
3	4.7	B02512	1	NaN	NaN	NaN	NaN	28.024628	2014-04-01 00:39:00	NaN	...	39	NaN	Na
4	4.7	B02512	1	NaN	NaN	NaN	NaN	12.083589	2014-04-01 00:40:00	NaN	...	39	NaN	Na

5 rows × 30 columns

当出现以下情况时有哪些问题:

- n 很大（p 为小到中等）
- p 很大（n 为小到中等）
- n 和 p 都很大

当 N 很大的时候，从统计角度来看，这通常不是一个大问题，从计算角度来看只是一个问题:

- 算法可能需要很长时间才能完成。
  - 估计回归模型的系数 (coefficients) 可能需要一些时间，尤其是没有闭合形式的模型（如 LASSO）
- 可以在训练集的子集上进行实验，从而解决计算方面的问题

当 P 很大时，就会出现很多问题:

- 矩阵可能无法逆（LR 中的问题）
- 可能存在多重共线性 (Multicollinearity)
- 模型容易受到过度拟合的影响

Multicollinearity: 多元回归模式下两个或多个自变量之间发生高相关性。

增加 p 一般可以用 feature interaction，如果我们经过 interaction term 的 p 很大，但 observations (或 n) 却很少，这时候我们就无法通过少量的 n 来确定大量的 p，就是训不过来了。我们管这种叫做 model unidentifiable。

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 (X_1 + X_2) + s$$

- Two-way interactions: $\binom{p}{2} = 253$
- Three-way interactions: $\binom{p}{3} = 1771$
- Etc.

The total number of all possible interaction terms (including main effects) is.

$$\sum_{k\,1\,2}^{p} \binom{p}{k} = 2^{p} \approx 8.3\text{million}$$

这种情况称为高维性 (High Dimensionality)，在执行数据分析和建模时需要考虑。

在实际操作中，我们可以：

- 增加观察次数
- 仅考虑具有重要科学意义的相互作用项 (interaction terms)
- 执行变量选择
- 执行另一种降维技术，如 PCA

Principal Components Analysis (PCA)

DIMENSIONALITY REDUCTION

生成高维空间的低维编码。

用途：

- 数据压缩/可视化
- 对噪声和不确定性的鲁棒性
- 可能更易于解释

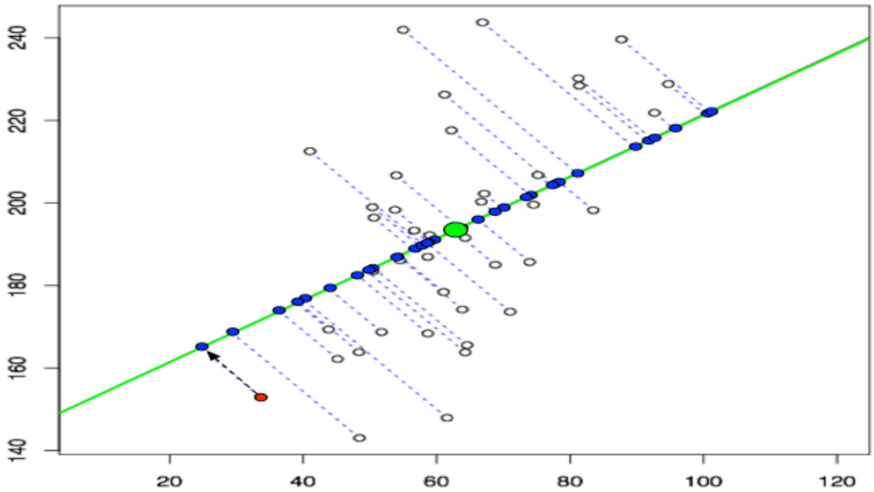
降维方法包括 2 个步骤：

- 确定一组最佳的新预测变量 Z1 , ... , Zm，for m < p
- 根据这些新的预测变量来表达数据中的每个观测值

转换后的数据将具有 m 列而不是 p 列。

PCA

主成分分析（PCA）是一种统计技术，允许识别数据集中的潜在线性模式，因此它可以用另一个具有有效较低维度的数据集来表示，而不会丢失太多信息。



IMPLEMENTATION OF PCA USING LINEAR ALGEBRA

- 取由 d+1 维组成的整个数据集（一列是一维），忽略标签，使我们的新数据集变为 d 维
- 计算整个数据集的每个维度的平均值
- 计算整个数据集的协方差矩阵（covariance matrix）
- 计算特征向量（eigenvectors）和相应的特征值（eigenvalues）
- 通过递减特征值对特征向量进行排序，并选择具有最大特征值的 k 个特征向量以形成 d × k 维矩阵 W
- 使用此 d × k 特征向量矩阵将样本转换为新的子空间

对下面的数据进行 PCA：

Student	Math	English	Art
1	90	60	90
2	90	90	30
3	60	60	60
4	60	60	90
5	30	30	30

计算整个数据集的每个维度的平均值：

$$\mathbf{A} = \begin{bmatrix} 90 & 60 & 90 \\ 90 & 90 & 30 \\ 60 & 60 & 60 \\ 60 & 60 & 90 \\ 30 & 30 & 30 \end{bmatrix}$$

Matrix A

So, The mean of matrix **A** would be

$$\bar{\mathbf{A}} = [\ 66 \quad 60 \quad 60\ ]$$

Mean of Matrix A

计算整个数据集的协方差矩阵 (covariance matrix):

$$cov(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n} (Xi - \bar{x})(Yi - \bar{y})$$

$$\begin{array}{c|ccc} & Math & English & Arts \\ \hline 1 & 90 & 60 & 90 \\ 2 & 90 & 90 & 30 \\ 3 & 60 & 60 & 60 \\ 4 & 60 & 60 & 90 \\ 5 & 30 & 30 & 30 \end{array}$$

Matrix A

$$\begin{array}{c|ccc} & Math & English & Art \\ \hline Math & 504 & 360 & 180 \\ English & 360 & 360 & 0 \\ Art & 180 & 0 & 720 \end{array}$$

Covariance Matrix of A

注：n 是 observations (或行数)，我们以算 (math, english) 为例：

$$cov(math, english) = \frac{1}{5}[(90-66)\times(60-60)+(90-66)\times(90-60)+(60-66)\times(60-60)+(60-66)\times(60-60)+(30-66)\times(30-60)]$$

得到结果 $cov(math, english) = 360$

计算特征向量 (eigenvectors) 和相应的特征值 (eigenvalues):

> *Let* **A** *be a square matrix,* **v** *a vector and* **λ** *a scalar that satisfies* **A**v = **λ**v, *then* **λ** *is called eigenvalue associated with eigenvector* **v** *of* **A**.

$$\det(A - \lambda I) = 0$$

$$\det\left(\begin{pmatrix} 504 & 360 & 180 \\ 360 & 360 & 0 \\ 180 & 0 & 720 \end{pmatrix} - \lambda\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\right) \qquad \det\begin{pmatrix} 504-\lambda & 360 & 180 \\ 360 & 360-\lambda & 0 \\ 180 & 0 & 720-\lambda \end{pmatrix}$$

$$-\lambda^3 + 1584\lambda^2 - 641520\lambda + 25660800 = 0 \qquad \lambda \approx 44.81966\ldots,\ \lambda \approx 629.11039\ldots,\ \lambda \approx 910.06995\ldots$$

Eigenvalues

eigenvectors

$$\begin{pmatrix} -3.75100\ldots \\ 4.28441\ldots \\ 1 \end{pmatrix}, \begin{pmatrix} -0.50494\ldots \\ -0.67548\ldots \\ 1 \end{pmatrix}, \begin{pmatrix} 1.05594\ldots \\ 0.69108\ldots \\ 1 \end{pmatrix}$$

$$\lambda \approx 44.81966\ldots, \lambda \approx 629.11039\ldots, \lambda \approx 910.06995\ldots$$

Eigenvalues

通过递减特征值对特征向量进行排序，并选择具有最大特征值的 k 个特征向量以形成 d × k 维矩阵 W:

So, after sorting the eigenvalues in decreasing order, we have

$$\begin{pmatrix} 910.06995 \\ 629.11039 \\ 44.81966 \end{pmatrix}$$

So, *eigenvectors* corresponding to two maximum eigenvalues are :

$$\mathbf{W} = \begin{bmatrix} 1.05594 & -0.50494 \\ 0.69108 & -0.67548 \\ 1 & 1 \end{bmatrix}$$

使用此 d × k 特征向量矩阵将样本转换为新的子空间:

将原数据和 W 相乘，得到的结果就是新的数据。

A FEWNOTES ON USING PCA

PCA 是一种无监督算法。

不会提高模型的预测能力；一般应用在可解释性很重要的情况下。

PCA 可以在非常高的维度设置下降低维度；可视化 feature 对响应的预测性。

PCA 保留了数据集内的协变 (covariation，意思就是两者有相同的变化趋势)，因此，主要保留了最大方差的轴。

NON-NEGATIVE MATRIX FACTORIZATION (NMF)

NMF 通过分解为两个非负矩阵来解释数据集。

有一些数据是多种独立数据组成的，比如多人说话的音轨、或多种乐器的合奏。NMF 可以识别构成组合数据的原始组件。

NMF 可以产生比 PCA 更多的可解释组件。

MATRIX FACTORIZATION

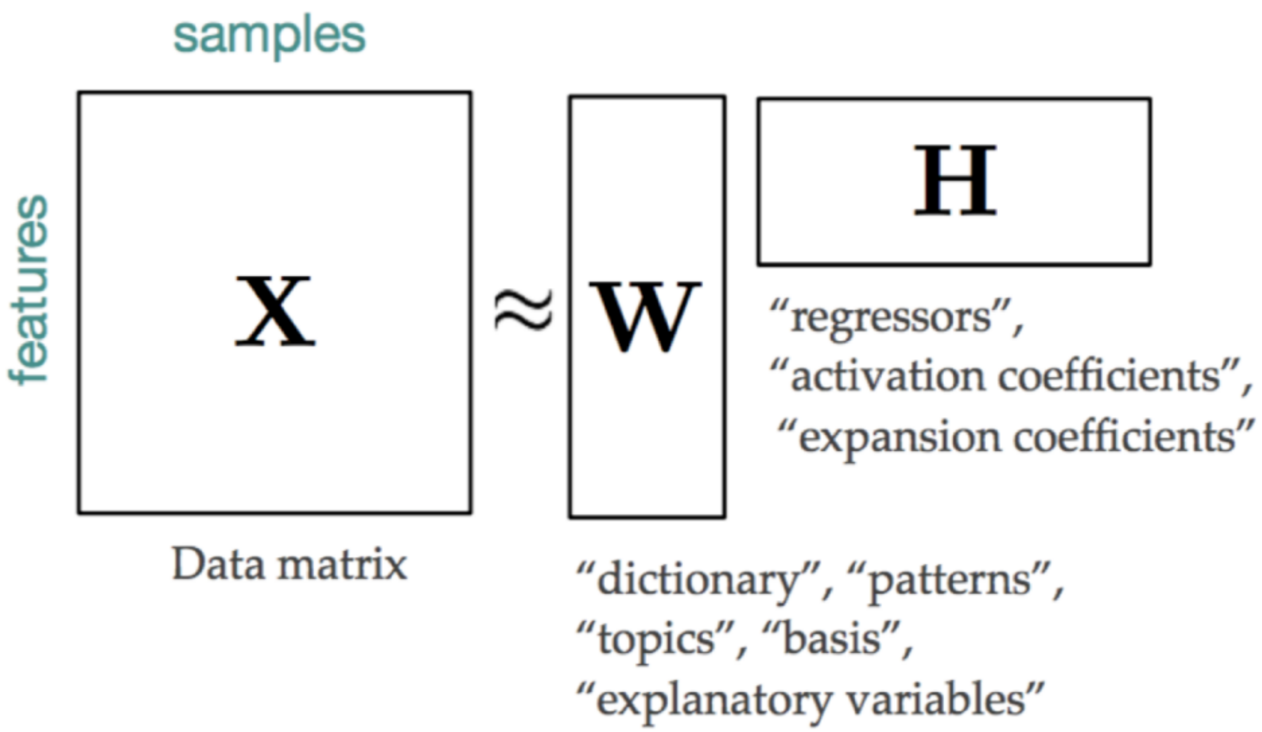
很多降维方法都用到了 matrix factorization (矩阵分解)。

基础思想：对于一个矩阵 X，找到 W 和 H 使得 W 和 H 的乘积最接近 X。

Low rank approximation to original $N \times M$ matrix:

$$\mathbf{X} \approx \mathbf{W}\mathbf{H}^T$$

where $\mathbf{W}$ is $N \times R$, $\mathbf{H}^T$ is $M \times R$, and $R \ll N$.



Generalization of many methods (e.g., SVD, QR, CUR, Truncated SVD, etc.)

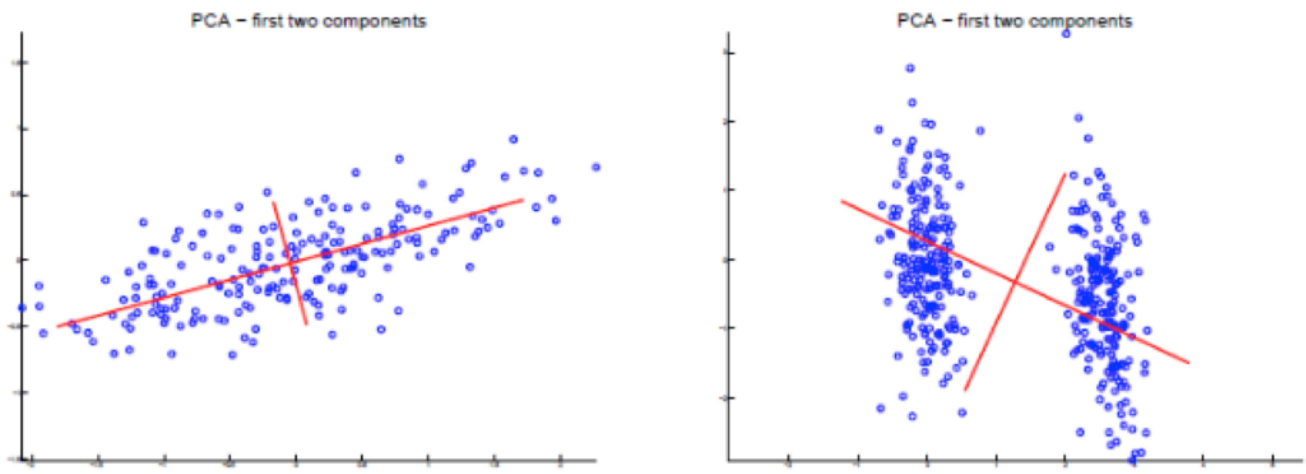
矩阵分解也是对数据进行压缩。

还原压缩数据的质量与使用 PCA 时相似，但稍差。这是预料之中的，因为 PCA 在重建方面找到了最佳方向。

NMF 通常不用于重建或编码数据，而是用于在数据中查找有趣的模式。

QA

- Assume I have some data in 2D. How to draw the first and second principal component in PCA method?



对于 2d 数据，第一个主成分将沿主要连续性方向对齐，第二个主成分将垂直于该主成分，沿最不连续的方向对齐。通常只是平均值 x / 平均值 y 处的特征向量（平均值 x / 平均值 y 确定原点）。