Overview

unsupervised learning可以分为两大类，dimension reduction和generation

Clustering

先根据input之间的相似度建立一颗树，再用一条线切一刀，如果使用红色的线切，那么可以分为2个大类

Distributed Representation

Dimension Reduction

用2维就可以表示这些特征，并不需要用到3D

这些3只有倾斜的角度不一样，用1D即可表示

主要有两种方法，feature selection和PCA

PCA

如果reduce to 1D，我们使用$z_1 = w^1 \cdot x$，使得x投影到w_1上，即达到了降维的目的，那么我们如何来评价降维的好坏呢?

我们可以使用降维之后数据的variance来评价，variance越大越好

如果reduce to 2D，那么现在就需要投影到两个不同的方向(w^1, w^2)上，再来与x做inner product，得到z_1, z_2，再分别计算这两者的variance；其中w^1, w^2要满足一定的条件，即$w^1 \cdot w^2 = 0$，两者是垂直的，可以保证是不同的方向

那么W就是一个正交矩阵，向量之间相互正交，且向量模长都是1

Formula

由于$a^T b$是一个scalar，所以可以直接加上转置符号

$$\begin{aligned}(a \cdot b)^2 &= a^T b a^T b \\ &= a^T b (a^T b)^T\end{aligned}$$

其中$Ex = \bar{x}$，协方差$Cov(x)$为

$$\begin{aligned}Cov(x) &= \frac{1}{N}\sum (x - Ex)(x - Ex)^T \\ &= \frac{1}{N}\sum (x - \bar{x})(x - \bar{x})^T\end{aligned}$$

令协方差为S，那么我们现在的问题是 maximizing $(w^1)^T S w^1$，限制条件是

$$||w^1||_2 = (w^1)^T w^1 = 1$$

先找到w^1，可以maximizing $(w^1)^T S w^1$，这里使用了拉格朗日乘数法，再对w^1求偏微分，得

$$Sw^1 = \alpha w^1$$

即w^1为S的特征向量 eigenvector

w^1为矩阵S的特征向量，对应的特征值λ_1是最大的

再找到w^2，对w^2求偏微分，得

$$Sw^2 - \alpha w^2 - \beta w^1 = 0$$

两边同时乘上$(w^1)^T$，得

$$(w^1)^T S w^2 - \alpha (w^1)^T w^2 - \beta (w^1)^T w^1 = 0$$

代入$(w^1)^T w^1 = 1, (w^2)^T w^1 = 0,$

$$\begin{aligned} &(w^1)^T S w^2 - \alpha (w^1)^T w^2 - \beta (w^1)^T w^1 \\ &= (w^1)^T S w^2 - \beta \end{aligned}$$

代入$Sw^1 = \lambda w^1,$

$$\begin{aligned} &(w^1)^T S w^2 \\ &= ((w^1)^T S w^2)^T = (w^2)^T S^T w^1 \\ &= (w^2)^T S w^1 = \lambda (w^2)^T w^1 = 0 \end{aligned}$$

那么

$$0 - \alpha\, 0 - \beta\, 1 = 0 \quad \rightarrow \beta = 0$$

代入$Sw^2 - \alpha w^2 - \beta w^1 = 0$，可得

$$Sw^2 - \alpha w^2 = 0 \quad \rightarrow Sw^2 = \alpha w^2$$

可得出w^2为矩阵S的特征向量，对应的特征值λ_2是第二大的

Decorrelation

$$W = \begin{pmatrix} (w_1)^T \\ (w_2)^T \\ \vdots \\ (w_K)^T \end{pmatrix}$$

$(w^1)^T$表示W的第一行，且$(w^1)^T w^1 = 1, (w^2)^T w^1 = 0$，因此$Ww^1 = e_1, \ldots, Ww^K = e_K,$

可得出$Cov(z)$是一个对角矩阵，只有正对角线上有元素

Another Point of View

下图中的7可以由三个部分组成，即u^1, u^3, u^5

那么我们目标就是找到这K个component，使得$||(x-\bar{x})-\hat{x}||_2$达到最小值

x可以分为$x^1, x^2, \ldots$，对应的c_1也可以分为$c_1^1, c_1^2, \ldots$，这样就形成了三个矩阵

那么我们怎么来最小化矩阵之间的最小差值呢?

下图中的U对应PCA中的权重矩阵W，为前文求出来的K个特征向量，为K个component，$\sum, V$表示C矩阵

对于矩阵XX^T的K个最大的特征值，矩阵U的每一列表示就表示这些特征值所对应的K个特征向量

做SVD求解出来的U矩阵，就是协方差矩阵$Cov(z)$所对应的特征向量，也就是PCA得出来的解

其中c_k可以用另外一种形式表达出来，$c_k=(x-\bar{x})\cdot w^k$，·表示做inner product

如果此时K=2，那么$c_1=\sum_{i=1}^{3}(x-\bar{x})\cdot w_i^k$，可以用neural network的形式表达出来，

那么c_1乘上w_i^1，就可以得到output为$\hat{x}_i$，

对于c_2也有类似的结果

对于network的output为$\hat{x}_i$，应与$x-\bar{x}$之间的error最小化

那么我们就可以把这个结构看成是具有一个hidden layer的network，其output和input应该越接近越好，这就可以叫做**Autoencoder**

Q：既然是neural network，那么我们可以用gradient descent来得到和PCA一样的最优解吗?

A：用PCA求解出来的w是相互正交的，可以让reconstruction error最小化，但gradient descent求解出来的w并不能保证这一点，而且并不能使这个reconstruction error比PCA方法更小

如果是在linear的情况下，使用PCA比较好，用network就会很麻烦；但network可以是deep的，可以中间有很多个hidden layer，这被称为**Deep Autoencoder**

Weakness of PCA

- unsupervised，输入的data是没有label的，PCA会找一种方式使得降维之后data的variance最大，就可能出现左上的结果，这时两者的class是不一样的，如果还是继续投影到红线上，就会出现两个class的data相互交错的局面；
- Linear，对于立体的data，如果还是继续pca，得到的结果也会非常不理想

Pokémon

MNIST