机器学习与数据挖掘

Machine Learning & Data Mining

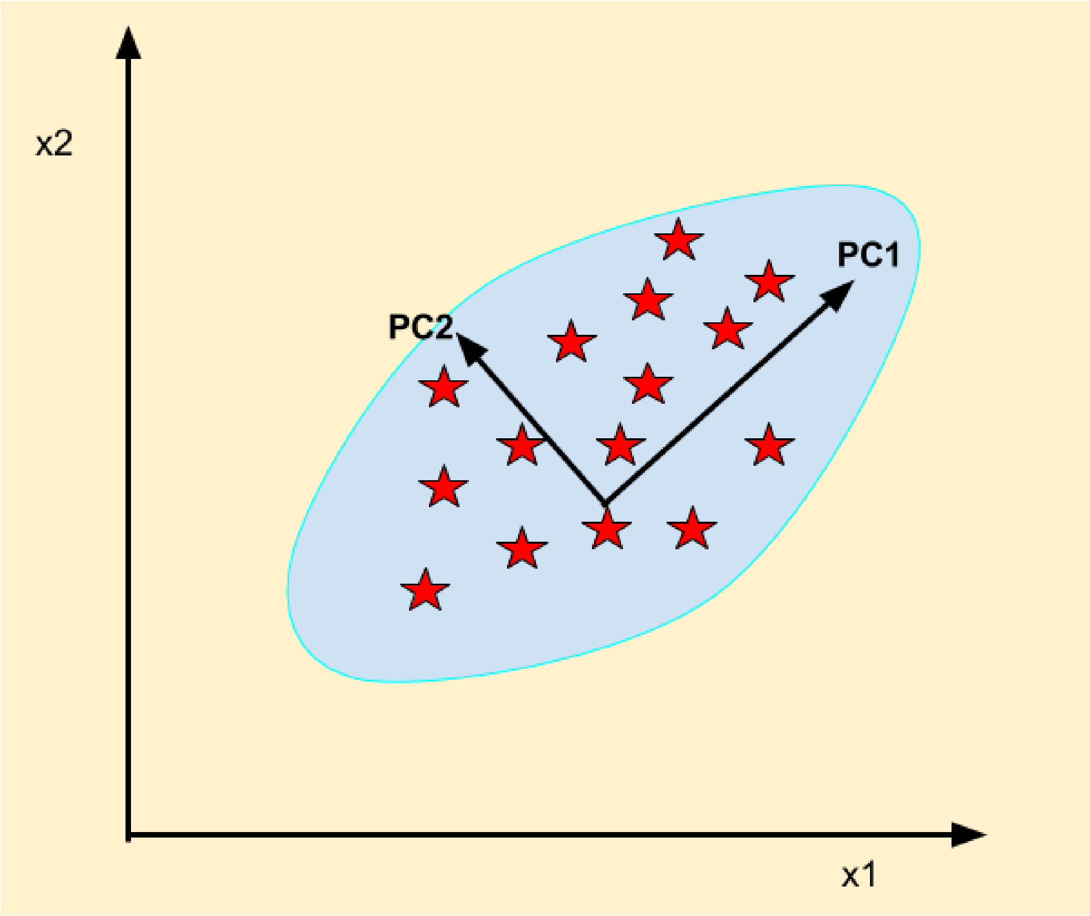
权小军 教授

中山大学数据科学与计算机学院

quanxj3@mail.sysu.edu.cn

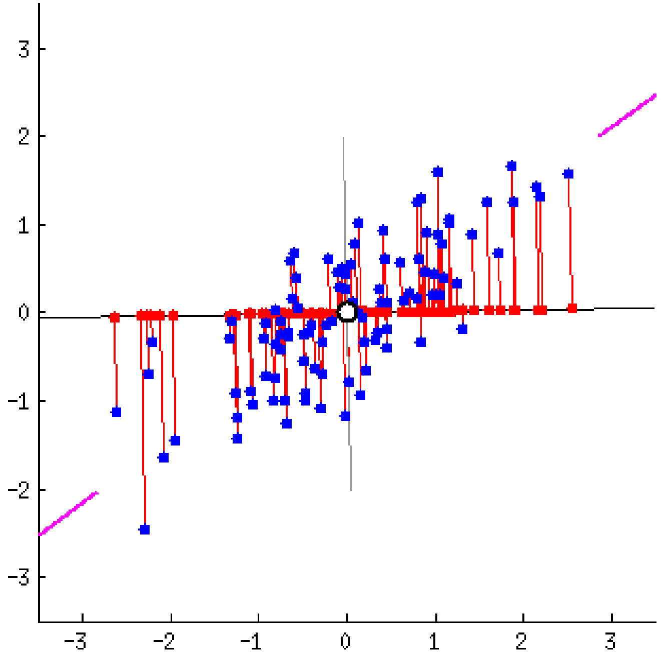
Preface

Preface



How to reduce the dimension from 2 to 1?

Preface



Lecture 14: Principal Components Analysis

Content

- 主成分分析
- 流形学习
- Autoencoders

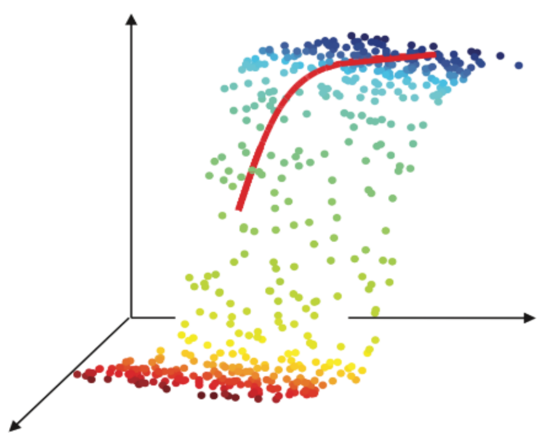
低维嵌入

- 在高维情形下出现的数据样本稀疏、距离计算困难等问题，是所有机器学习方法共同面临的严重障碍，被称为“维数灾难”。
- 缓解维数灾难的一个重要途径是降维(dimension reduction)
  - 即通过某种数学变换，将原始高维属性空间转变为一个低维“子空间”，在这个子空间中样本密度大幅度提高，距离计算也更为容易。

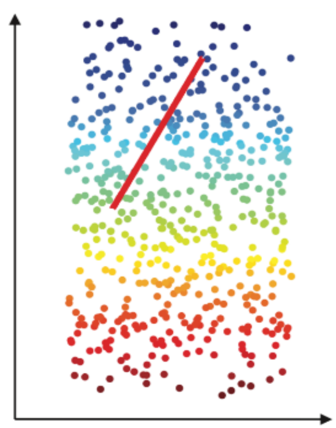
低维嵌入

为什么能进行降维?

- 数据样本虽然是高维的，但与学习任务密切相关的也许仅是某个低维分布，即高维空间中的一个低维“嵌入”（embedding），因而可以对数据进行有效的降维。

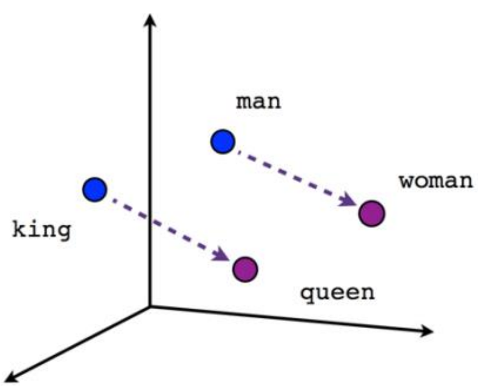


(a) 三维空间中观察到的样本点

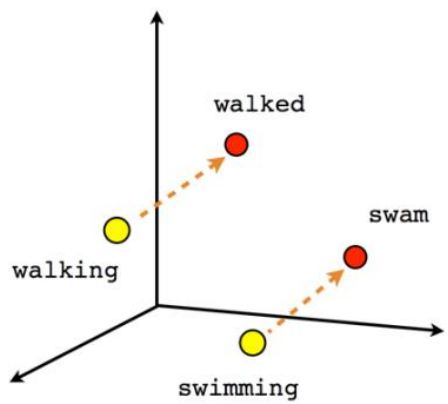


(b) 二维空间中的曲面

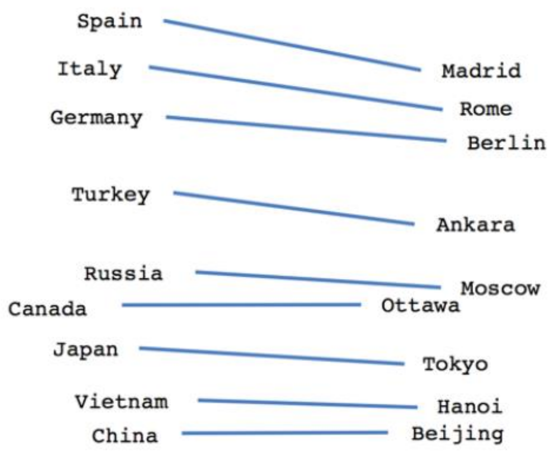
Word Embeddings



Male-Female



Verb tense



Country-Capital

线性降维方法

- 一般来说，欲获得低维子空间，最简单的是对原始高维空间进行线性变换。给定 d 维空间中的样本，

$$\mathbf{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m) \in \mathbb{R}^{d \times m}$$

变换之后得到 $d' \leq d$ 维空间中的样本

$$\mathbf{Z} = \mathbf{W}^{\mathrm{T}}\mathbf{X},$$

其中$\mathbf{W} \in \mathbb{R}^{d \times d'}$是变换矩阵,$\mathbf{Z} \in \mathbb{R}^{d' \times m}$是样本在新空间中的表达。

线性降维方法

- 变换矩阵 $\mathbf{W}$ 可视为 d' 个 d 维属性向量。换言之，$\boldsymbol{z}_i$ 是原属性向量 $\boldsymbol{x}_i$ 在新坐标系 $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'}\}$ 中的坐标向量。若 $\boldsymbol{w}_i$ 与 $\boldsymbol{w}_j (i \neq j)$ 正交，则新坐标系是一个正交坐标系，此时 $\mathbf{W}$ 为正交变换。显然，新空间中的属性是原空间中的属性的线性组合。
- 基于线性变换来进行降维的方法称为线性降维方法，对低维子空间性质的不同要求可通过对 $\mathbf{W}$ 施加不同的约束来实现。

主成分分析

主成分分析(Principal Component Analysis, PCA)

- 对于正交属性空间中的样本点，如何用一个超平面对所有样本进行恰当的表达？
- 容易想到，若存在这样的超平面，那么它应具有这样的性质：
  - 最近重构性：样本点到这个超平面的距离都足够近；
  - 最大可分性：样本点在这个超平面上的投影能尽可能分开。
- 基于最近重构性和最大可分性，能得到主成分分析的两种等价推导。

主成分分析

最近重构性

- 对样本进行中心化，$\sum_i \boldsymbol{x}_i = \mathbf{0}$，再假定投影变换后得到的新坐标系为$\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_d\}$，其中 $\boldsymbol{w}_i$ 是标准正交基向量，

$$||\boldsymbol{w}_i||_2 = 1, \boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{w}_j = 0 (i \neq j).$$

主成分分析

最近重构性

- 若丢弃新坐标系中的部分坐标，即将维度降低到 $d' < d$ ，则样本点在低维坐标系中的投影是 $\boldsymbol{z}_i = (z_{i1}; z_{i2}; \ldots; z_{id'})$， $z_{ij} = \boldsymbol{w}_j^{\mathrm{T}} \boldsymbol{x}_i$ 是 $\boldsymbol{x}_i$ 在低维坐标下第 j 维的坐标，若基于 $\boldsymbol{z}_i$ 来重构 $\boldsymbol{x}_i$，则会得到

$$\hat{\boldsymbol{x}}_i = \sum_{j=1}^{d'} z_{ij} \boldsymbol{w}_j .$$

主成分分析

最近重构性

- 考虑整个训练集，原样本点 $\boldsymbol{x}_i$ 与基于投影重构的样本点 $\hat{\boldsymbol{x}}_i$ 之间的距离为

$$
\begin{aligned}
\sum_{i=1}^{m}\left\|\sum_{j=1}^{d'} z_{ij}\boldsymbol{w}_j-\boldsymbol{x}_i\right\|_2^2 &= \sum_{i=1}^{m}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_i-2\sum_{i=1}^{m}\boldsymbol{z}_i^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\boldsymbol{x}_i+\mathrm{const} \\
&\propto -\mathrm{tr}\left(\mathbf{W}^{\mathrm{T}}\left(\sum_{i=1}^{m}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\right)\mathbf{W}\right).
\end{aligned}
$$

主成分分析

最近重构性

- 根据最近重构性应最小化上式。考虑到$\boldsymbol{w}_j$是标准正交基，$\sum_i \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}}$是协方差矩阵，有

$$\min_{\mathbf{W}} \quad -\operatorname{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W}) \quad \text{s.t.} \quad \mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}.$$

这就是主成分分析的优化目标。

主成分分析

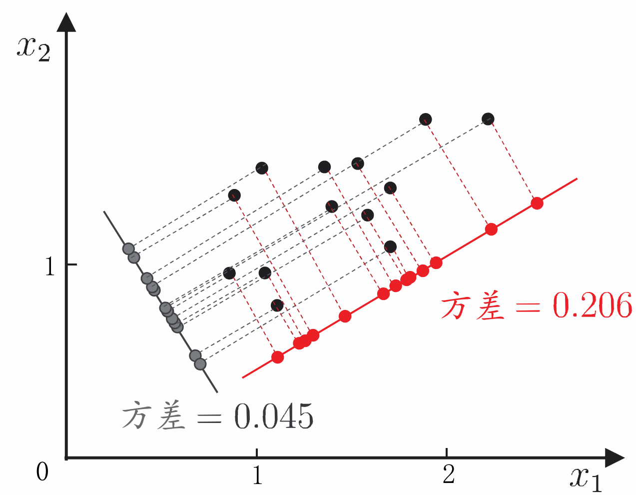
最大可分性

- 样本点$\boldsymbol{x}_i$在新空间中超平面上的投影是$\mathbf{W}^{\mathrm{T}}\boldsymbol{x}_i$，若所有样本点的投影能尽可能分开，则应该使得投影后样本点的方差最大化。若投影后样本点的方差是$\sum_i \mathbf{W}^{\mathrm{T}}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\mathbf{W}$，于是优化目标可写为

$$\max_{\mathbf{W}} \quad \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W})$$
$$\text{s.t.} \quad \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}.$$

显然与 $\min_{\mathbf{W}} \quad -\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W})$ 等价。

$$\text{s.t.} \quad \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}.$$



主成分分析

PCA的求解

$$\max_{\mathbf{W}} \quad \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W}) \qquad \min_{\mathbf{W}} \quad -\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W})$$

$$\text{s.t.} \quad \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}. \qquad \text{s.t.} \quad \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}.$$

- 对优化式使用拉格朗日乘子法可得

$$\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W} = \lambda\mathbf{W}.$$

只需对协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ 进行特征值分解，并将求得的特征值排序：$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$，再取前$d'$个特征值对应的特征向量构成$\mathbf{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'})$，这就是主成分分析的解。

主成分分析

PCA算法

输入： 样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;

低维空间维数 d'.

过程：

1: 对所有样本进行中心化: $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}_i$;

2: 计算样本的协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$;

3: 对协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ 做特征值分解;

4: 取最大的 d' 个特征值所对应的特征向量 $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'}$.

输出： 投影矩阵 $\mathbf{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'})$.

主成分分析

- 降维后低维空间的维数d'通常是由用户事先指定，或通过在d'值不同的低维空间中对k近邻分类器（或其它开销较小的学习器）进行交叉验证来选取较好的d'值。对PCA，还可从重构的角度设置一个重构阈值，例如$t = 95\%$，然后选取使下式成立的最小d'值：

$$\frac{\sum_{i=1}^{d'} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq t.$$

主成分分析

- PCA仅需保留**W**与样本的均值向量即可通过简单的向量减法和矩阵-向量乘法将新样本投影至低维空间中。如何做?

今天下午16点前将答案发送到邮箱: sysucsers@163.com

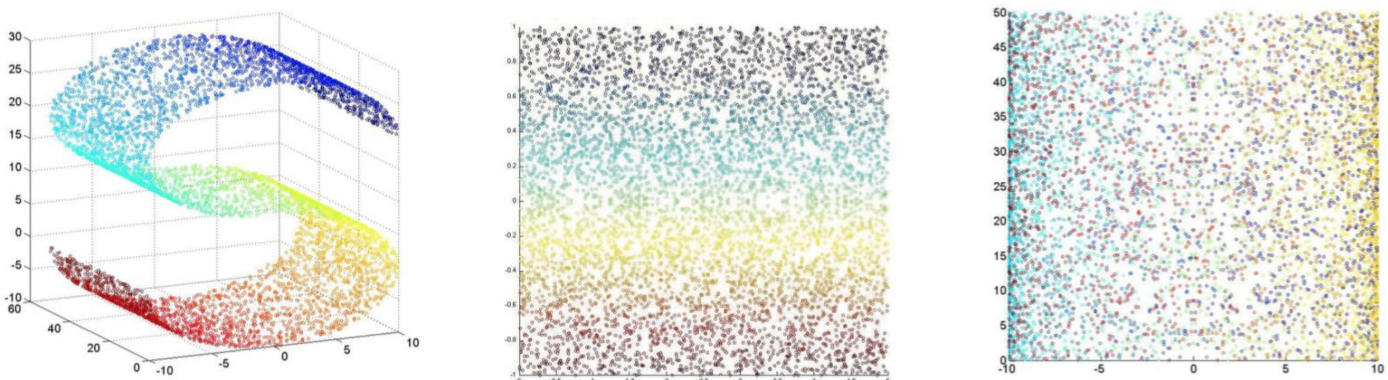
邮件标题和文件名称: 学号-姓名-Lecture 14(pca), 例如: 01111-张三- Lecture 13(pca).

主成分分析

- 降维虽然会导致信息的损失，但一方面舍弃这些信息后能使得样本的采样密度增大，另一方面，当数据受到噪声影响时，最小的特征值所对应的特征向量往往与噪声有关，舍弃可以起到去噪效果。

核化线性降维

- 线性降维方法假设从高维空间到低维空间的函数映射是线性的，然而，在不少现实任务中，可能需要非线性映射才能找到恰当的低维嵌入：



(a) 三维空间中的观察　(b) 本真二维结构　(c) PCA 降维结果

核化线性降维

核化主成分分析 (Kernelized PCA, 简称KPCA)

- 非线性降维的一种常用方法，是基于核技巧对线性降维方法进行“核化”（kernelized)。

- 假定我们将在高维特征空间中把数据投影到由$\mathbf{W}$确定的超平面上，即PCA欲求解

$$\left(\sum_{i=1}^{m} \boldsymbol{z}_i \boldsymbol{z}_i^{\mathrm{T}}\right) \mathbf{W} = \lambda \mathbf{W}.$$

- 其中$\boldsymbol{z}_i$是样本点$\boldsymbol{x}_i$在高维特征空间中的像。令 $\boldsymbol{\alpha}_i = \frac{1}{\lambda} \boldsymbol{z}_i^{\mathrm{T}} \mathbf{W}$,

$$\mathbf{W} = \frac{1}{\lambda}\left(\sum_{i=1}^{m} \boldsymbol{z}_i \boldsymbol{z}_i^{\mathrm{T}}\right) \mathbf{W} = \sum_{i=1}^{m} \boldsymbol{z}_i \frac{\boldsymbol{z}_i^{\mathrm{T}} \mathbf{W}}{\lambda} = \sum_{i=1}^{m} \boldsymbol{z}_i \boldsymbol{\alpha}_i.$$

核化线性降维

核化主成分分析 (Kernelized PCA, 简称KPCA)

- 假定$\boldsymbol{z}_i$是由原始属性空间中的样本点$\boldsymbol{x}_i$通过映射ϕ产生，即

$$\boldsymbol{z}_i = \phi(\boldsymbol{x}_i),\ i = 1, 2, \ldots, m.$$

$$\mathbf{W} = \sum_{i=1}^{m} \boldsymbol{z}_i \boldsymbol{\alpha}_i$$

- 若ϕ能被显式表达出来，则通过它将样本映射至高维空间，再在特征空间中实施PCA即可，即有

$$\left(\sum_{i=1}^{m} \phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)^{\mathrm{T}}\right)\mathbf{W} = \lambda \mathbf{W}.$$

并且

$$\mathbf{W} = \sum_{i=1}^{m} \phi(\boldsymbol{x}_i)\boldsymbol{\alpha}_i.$$

核化线性降维

核化主成分分析 (Kernelized PCA, 简称KPCA)

- 一般情形下，我们不清楚ϕ的具体形式，于是引入核函数

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^{\mathrm{T}}\phi(\boldsymbol{x}_j).$$

- 又由$\mathbf{W} = \sum_{i=1}^{m}\phi(\boldsymbol{x}_i)\boldsymbol{\alpha}_i$，代入优化式$\left(\sum_{i=1}^{m}\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)^{\mathrm{T}}\right)\mathbf{W} = \lambda\mathbf{W}$，有

$$\mathbf{KA} = \lambda\mathbf{A}.$$

其中$\mathbf{K}$为κ对应的核矩阵, $(\mathbf{K})_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j), \mathbf{A} = (\boldsymbol{\alpha}_1; \boldsymbol{\alpha}_2; \ldots; \boldsymbol{\alpha}_m)$.

- 上式为特征值分解问题，取$\mathbf{K}$最大的d'个特征值对应的特征向量得到解。

核化线性降维

核化主成分分析 (Kernelized PCA, 简称KPCA)

- 对新样本$\boldsymbol{x}$，其投影后的第 j $(j = 1, 2, \ldots, d')$ 维坐标为

$$z_j = \boldsymbol{w}_j^{\mathrm{T}} \phi(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i^j \phi(\boldsymbol{x}_i)^{\mathrm{T}} \phi(\boldsymbol{x})$$

$$= \sum_{i=1}^{m} \alpha_i^j \kappa(\boldsymbol{x}_i, \boldsymbol{x}).$$

其中$\boldsymbol{\alpha}_i$已经过规范化，α_i^j是$\boldsymbol{\alpha}_i$的第j个分量。由该式可知，为获得投影后的坐标，KPCA需对所有样本求和，因此它的计算开销较大。

Content

- 主成分分析
- 流形学习
- Autoencoders

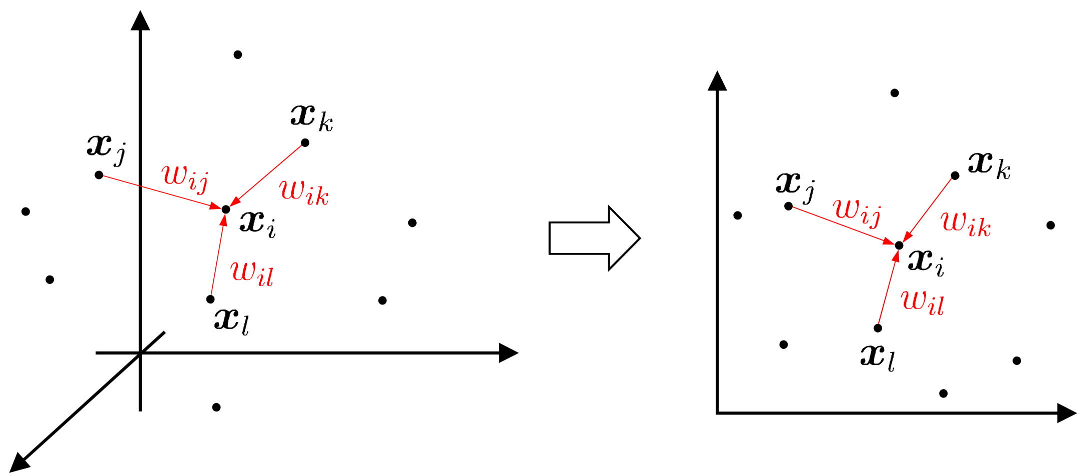
流形学习

- 流形学习(manifold learning)是一类借鉴了拓扑流形概念的降维方法。“流形”在局部具有欧氏空间的性质，能用欧氏距离来进行距离计算。
- 若低维流形嵌入到高维空间中，则数据样本在高维空间的分布虽然看上去非常复杂，但在局部上仍具有欧氏空间的性质，因此，可以容易地在局部建立降维映射关系，然后再设法将局部映射关系推广到全局。
- 当维数被降至二维或三维时，能对数据进行可视化展示，因此流形学习也可被用于可视化。

流形学习

局部线性嵌入（Locally Linear Embedding，LLE）

- 局部线性嵌入试图保持邻域内的线性关系，并使得该线性关系在降维后的空间中继续保持。



$$\boldsymbol{x}_i = w_{ij}\boldsymbol{x}_j + w_{ik}\boldsymbol{x}_k + w_{il}\boldsymbol{x}_l$$

流形学习

局部线性嵌入 (Locally Linear Embedding，LLE)

- LLE先为每个样本$\boldsymbol{x}_i$找到其近邻下标集合Q_i，然后计算出基于Q_i的中的样本点对$\boldsymbol{x}_i$进行线性重构的系数$\boldsymbol{w}_i$：

$$\min_{\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_m} \sum_{i=1}^{m} \left\| \boldsymbol{x}_i - \sum_{j \in Q_i} w_{ij} \boldsymbol{x}_j \right\|_2^2$$

$$\text{s.t.} \sum_{j \in Q_i} w_{ij} = 1,$$

其中$\boldsymbol{x}_i$和$\boldsymbol{x}_j$均为已知，令$C_{jk} = (\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{x}_k)$，$w_{ij}$有闭式解

$$w_{ij} = \frac{\sum_{k \in Q_i} C_{jk}^{-1}}{\sum_{l,s \in Q_i} C_{ls}^{-1}}.$$

流形学习

局部线性嵌入 (Locally Linear Embedding，LLE)

- LLE在低维空间中保持$\boldsymbol{w}_i$不变，于是$\boldsymbol{x}_i$对应的低维空间坐标$\boldsymbol{z}_i$可通过下式求解：

$$\min_{\mathrm{z}_1, \mathrm{z}_2, \ldots, \mathrm{z}_m} \sum_{i=1}^{m} \left\| \boldsymbol{z}_i - \sum_{j \in Q_i} w_{ij} \boldsymbol{z}_j \right\|_2^2$$

- 令$\mathbf{Z} = (\mathrm{z}_1, \mathrm{z}_2, \ldots, \mathrm{z}_m) \in \mathbb{R}^{d' \times m}, (\mathbf{W})_{ij} = w_{ij},$

$$\mathbf{M} = (\mathbf{I} - \mathbf{W})^{\mathrm{T}} (\mathbf{I} - \mathbf{W}),$$

- 则优化式可重写为下式，并通过特征值分解求解。

$$\min_{\mathbf{Z}} \operatorname{tr}(\mathbf{Z} \mathbf{M} \mathbf{Z}^{\mathrm{T}}) \qquad \text{s.t. } \mathbf{Z} \mathbf{Z}^{\mathrm{T}} = \mathbf{I}.$$

流形学习

局部线性嵌入 (Locally Linear Embedding，LLE)

输入： 样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
近邻参数 k;
低维空间维数 d'.

过程：

1: **for** $i = 1, 2, \ldots, m$ **do**
2: 确定 $\boldsymbol{x}_i$ 的 k 近邻;
3: 从式(10.27)求得 w_{ij}, $j \in Q_i$;
4: 对于 $j \notin Q_i$, 令 $w_{ij} = 0$;
5: **end for**
6: 从式(10.30)得到 $\mathbf{M}$;
7: 对 $\mathbf{M}$ 进行特征值分解;
8: **return** $\mathbf{M}$ 的最小 d' 个特征值对应的特征向量

输出： 样本集 D 在低维空间的投影 $Z = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m\}$.

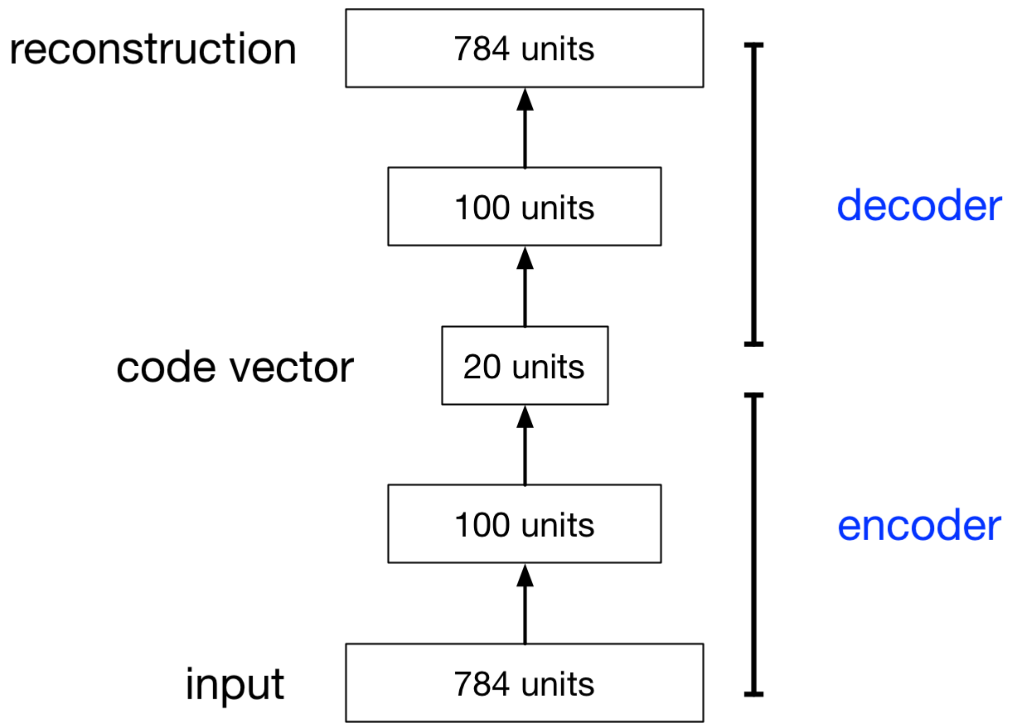
LLE 算法

Content

- 主成分分析
- 流形学习
- Autoencoders

Autoencoders

- An autoencoder is a feed-forward neural net whose job it is to take an input **x** and predict **x**.
- To make this non-trivial, we need to add a bottleneck layer whose dimension is much smaller than the input.



Linear Autoencoders

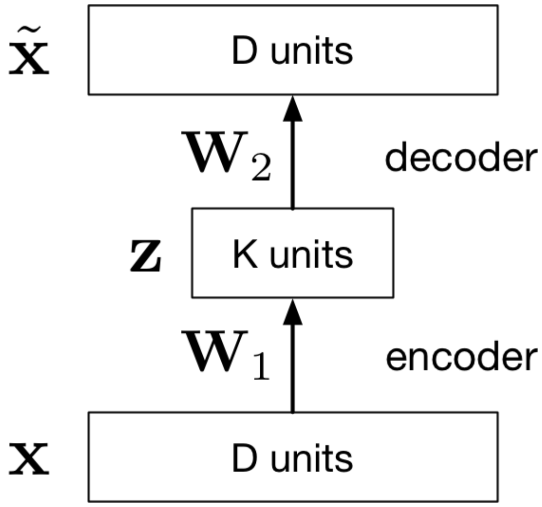
Why autoencoders?

- Map high-dimensional data to two dimensions for visualization
- Learn abstract features in an unsupervised way so you can apply them to a supervised task
  - Unlabled data can be much more plentiful than labeled data

Linear Autoencoders

- The simplest kind of autoencoder has one hidden layer, linear activations, and squared error loss.

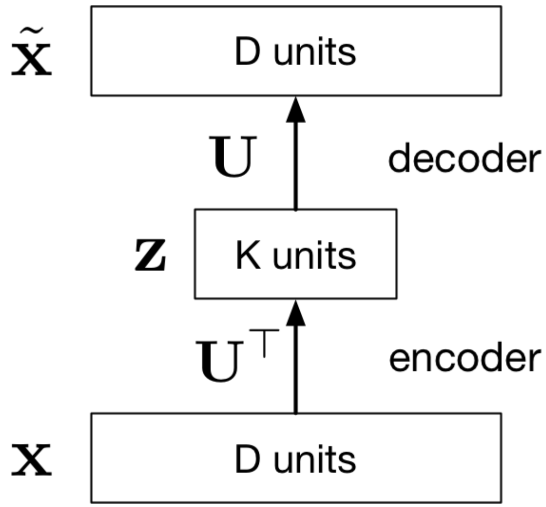
$$L(\mathbf{x}, \tilde{\mathbf{x}}) = \|\mathbf{x} - \tilde{\mathbf{x}}\|^2$$



- This network computes $\tilde{\mathbf{x}} = \mathbf{W}_2\mathbf{W}_1\mathbf{x}$, which is a linear function.
- If $K \geq D$, we can choose $\mathbf{W}_2$ and $\mathbf{W}_1$ such that $\mathbf{W}_2\mathbf{W}_1$ is the identity matrix. This isn't very interesting.
- But suppose $K < D$:
  - $\mathbf{W}_1$ maps $\mathbf{x}$ to a K-dimensional space, so it's doing dimensionality reduction.

Linear Autoencoders

- Observe that the output of the autoencoder must lie in a K -dimensional subspace spanned by the columns of $\mathbf{W}_2$.
- We saw that the best possible K -dimensional subspace in terms of reconstruction error is the PCA subspace.
- The autoencoder can achieve this by setting $\mathbf{W}_1 = \mathbf{U}^T$ and $\mathbf{W}_2 = \mathbf{U}$.
- Therefore, the optimal weights for a linear autoencoder are just the principal components!



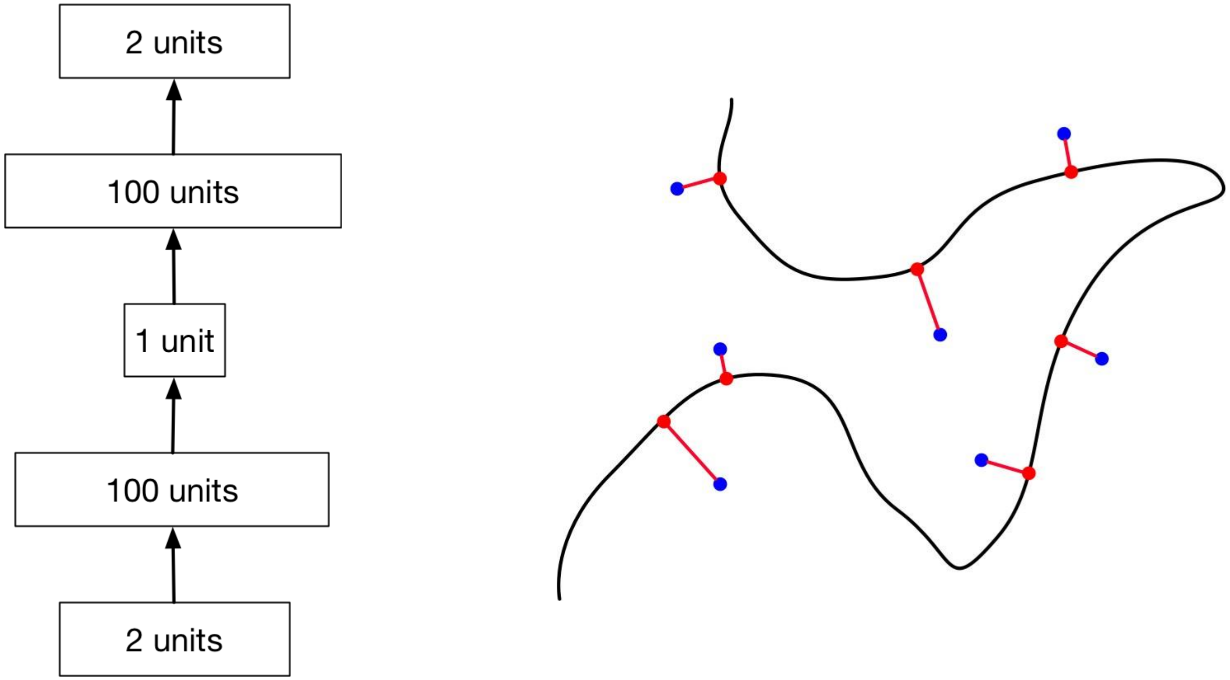
Linear Autoencoders

If linear activations are used, or only a single sigmoid hidden layer, then the optimal solution to an autoencoder is strongly related to principal component analysis (PCA).

-Wikipedia

Nonlinear Autoencoders

- Deep nonlinear autoencoders learn to project the data, not onto a subspace, but onto a nonlinear manifold.
- This manifold is the image of the decoder.
- This is a kind of nonlinear dimensionality reduction.



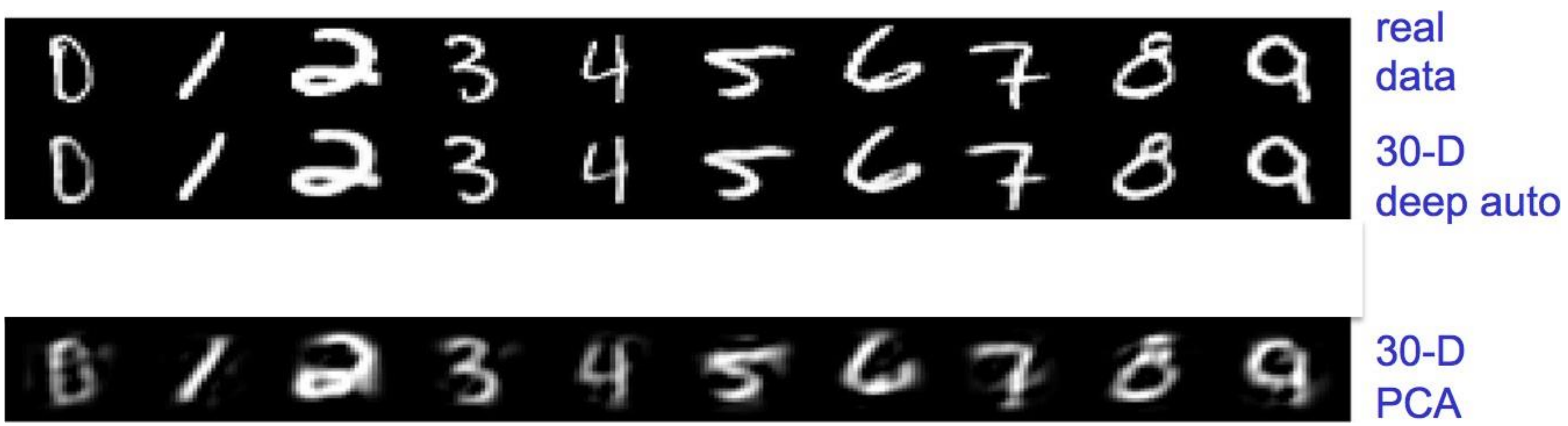
Linear Autoencoders

However, the potential of Autoencoders resides in their non-linearity, allowing the model to learn more powerful generalizations compared to PCA, and to reconstruct back the input with a significantly lower loss of information.

-Wikipedia

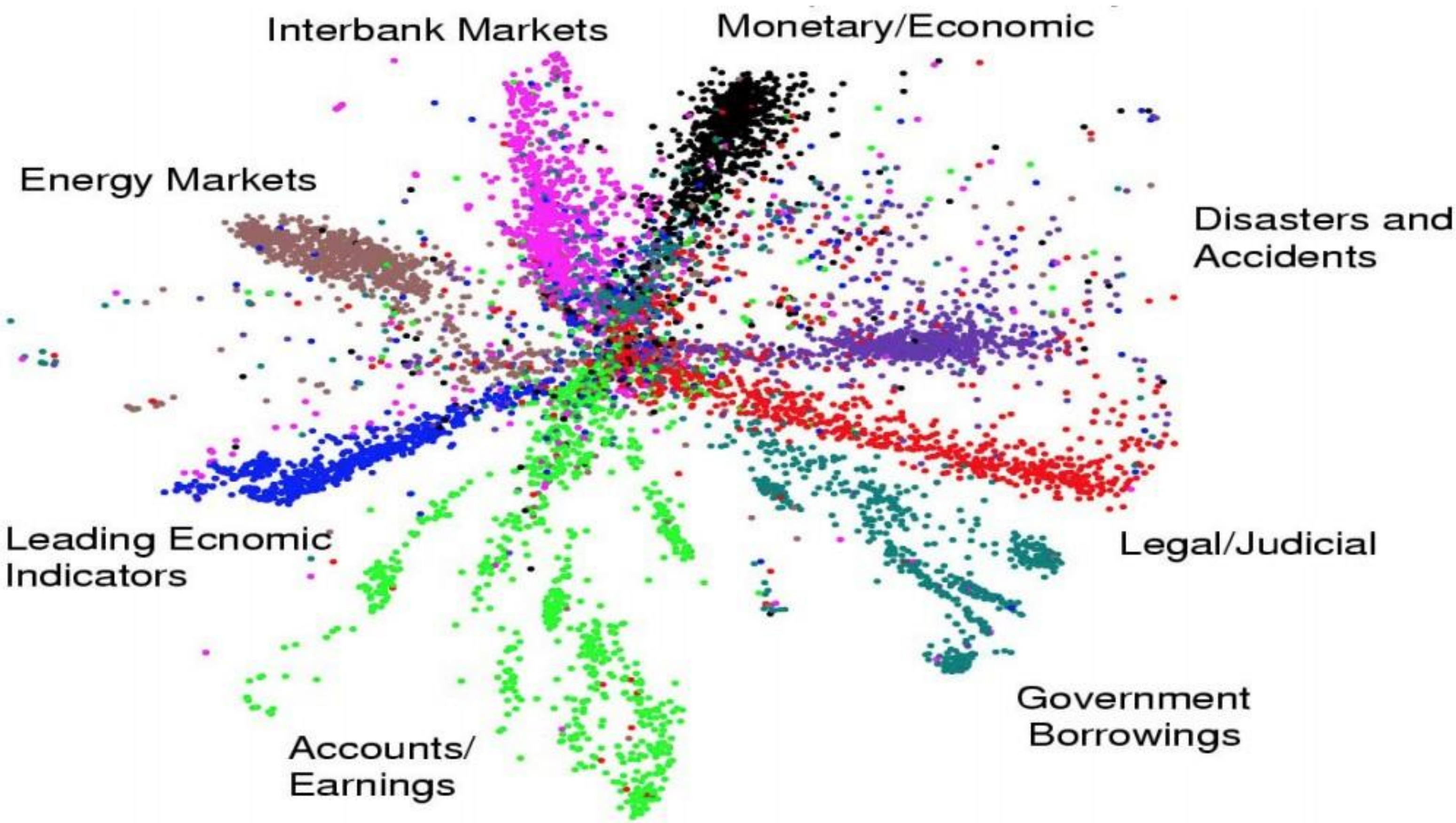
Nonlinear Autoencoders

- Nonlinear autoencoders can learn more powerful codes for a given dimensionality, compared with linear autoencoders (PCA)



Nonlinear Autoencoders

Here's a 2-dimensional autoencoder representation of newsgroup articles. They're color-coded by topic, but the algorithm wasn't given the labels.



Thank you!

权小军 中山大学数据科学与计算机学院