机器学习
Machine Learning

第10讲：无监督学习-2

Part-2 PCA降维

ICA（独立分量分析）

1. PCA方法降维

Principal Component Analysis：PCA

数据样本集（为了用下标表示主分量方便，用(n)表示样本序列（或改为上标亦可），即

$$\left\{\boldsymbol{x}(n), n=1,\cdots,T\right\}$$

设任何一个样本是M维向量

$$\boldsymbol{x}(n)=[x_1(n),x_2(n),\cdots,x_M(n)]^T$$

可以用一个K<M向量

$$\hat{\boldsymbol{y}}(n)=\left[y_1(n),y_2(n),\cdots,y_K(n)\right]^T$$

进行逼近表示，并对训练集外向量有好的泛化性

PCA-特征分解

为处理简单，假设 $E\left[\boldsymbol{x}(n)\right]=\mathbf{0}$

在零均值假设下，自相关矩阵代替协方差矩阵，即

$$\mathbf{R}_{\mathbf{x}}=\Sigma_{\mathbf{x}}$$

对于训练集，自相关矩阵计算为

$$\mathbf{R}_{\mathbf{x}}=\frac{1}{T}\sum_{i=1}^{T}\boldsymbol{x}(i)\boldsymbol{x}^{\mathrm{T}}(i)=\frac{1}{T}\mathbf{X}^{\mathrm{T}}\mathbf{X}$$ （<u>**或扩展到**</u> $\frac{1}{T}\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}$ ）

做特征分解，得所有特征值集：$\left\{\lambda_1,\lambda_2,\cdots,\lambda_M\right\}$ $\lambda_i\geq 0$

相应特征向量集： $\left\{\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_M\right\}$

PCA-特征分解（续）

取特征向量 $\boldsymbol{q}_k$ 是归一化的，并构成特征矩阵

$$\mathbf{Q} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_M]$$

并且 $\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{Q}\mathbf{Q}^{\mathrm{T}} = \mathbf{I}$

对于训练集中或同生成概率所产生的向量 $\boldsymbol{x}$
定义向量 $\boldsymbol{y}$ 为

$$\boldsymbol{y} = \mathbf{Q}^T \boldsymbol{x}$$

则有：$\mathbf{Q}\boldsymbol{y} = \mathbf{Q}\mathbf{Q}^T\boldsymbol{x} = \boldsymbol{x}$

这里 $\boldsymbol{y}, \boldsymbol{x}$ 是一对正反变换对（称为KL变换）

PCA-特征分解（续）

利用 $$\mathbf{R}_{\mathbf{x}} = \mathbf{Q}\Lambda\mathbf{Q}^T$$

可得 $$E\left[\boldsymbol{y}\boldsymbol{y}^T\right] = diag(\lambda_1, \lambda_2, \cdots, \lambda_M) = \Lambda$$

并且 $$E\left[\left|y_i\right|^2\right] = \lambda_i$$

和 $$E\left[\left\|\boldsymbol{x}\right\|_2^2\right] = E\left[\sum_{i=1}^{M}\left|x_i\right|^2\right] = E\left[\left\|\boldsymbol{y}\right\|_2^2\right] = \sum_{i=1}^{M}\lambda_i$$

PCA表示

如果让特征值按大小排序 $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$

仅取 $\boldsymbol{y}$ 的前K<M个系数 $\hat{\boldsymbol{y}} = \left[y_1, y_2, \cdots, y_K\right]^T$

即 $\hat{\boldsymbol{y}} = \mathbf{Q}_{\mathbf{K}}^{\mathbf{T}} \boldsymbol{x}$ 或 $y_i = \boldsymbol{q}_i^T \boldsymbol{x}, \quad i = 1, 2, \cdots, K$

这里 $\mathbf{Q}_{\mathbf{K}} = \left[\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_K\right]$ 是 $M \times K$ 维矩阵且 $\mathbf{Q}_{\mathbf{K}}^{\mathbf{T}} \mathbf{Q}_{\mathbf{K}} = \mathbf{I}$ （$K \times K$）

通过 $\hat{\boldsymbol{y}}$ ，可得 $\boldsymbol{x}$ 的近似表示 $\hat{\boldsymbol{x}}$（注 $\hat{\boldsymbol{x}}$ 仍是M维向量）

$$\hat{\boldsymbol{x}} = \left[\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_K\right] \hat{\boldsymbol{y}} = \mathbf{Q}_{\mathbf{K}} \hat{\boldsymbol{y}} = \sum_{i=1}^{K} y_i \boldsymbol{q}_i$$

通过K维向量 $\hat{\boldsymbol{y}}$ 近似表示 $\boldsymbol{x}$ ，称为 $\boldsymbol{x}$ 的PCA表示

PCA表示（续）

PCA表示的能量

$$E\left[\|\hat{\boldsymbol{x}}\|_2^2\right] = E\left[\sum_{i=1}^{M} |\hat{x}_i|^2\right] = E\left[\|\hat{\boldsymbol{y}}\|_2^2\right] = \sum_{i=1}^{K} \lambda_i$$

定义误差向量

$$\boldsymbol{e} = \boldsymbol{x} - \hat{\boldsymbol{x}}$$

误差向量评估

$$E\left[\|\boldsymbol{e}\|_2^2\right] = \sum_{i=K+1}^{M} \lambda_i$$

PCA的解释

通过训练样本集合 $\{\boldsymbol{x}(n),\ n=1, 2\cdots, \mathrm{T}\}$

得到K个主分量 $\boldsymbol{q}_i \quad i=1,2,\cdots,K$

对于 $\boldsymbol{x}$ （来自训练集或一个泛化样本），近似表示

$$\hat{\boldsymbol{x}} = \left[\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_K\right]\hat{\boldsymbol{y}} = \sum_{i=1}^{K} y_i \boldsymbol{q}_i$$

用K维向量 $\hat{\boldsymbol{y}}$ 表示 $\boldsymbol{x}$，其中 y_i 是 $\boldsymbol{x}$ 在 $\boldsymbol{q}_i$ 的投影

<u>主分量分析的要点是用低维向量表示高维向量，是一种对高维数据的有效降维方法</u>

2. PCA的在线算法

给出训练样本 $\{\boldsymbol{x}(n),\ n=1, 2\cdots, \mathrm{T}\}$

通过在线迭代得到主分量

1. 推导第一个主分量 $\boldsymbol{q}_1$

迭代中，迭代 $\boldsymbol{w}_1(n)=\left[w_{11}(n), w_{12}(n), \cdots, w_{1k}(n), \cdots, w_{1M}(n)\right]^T$

$n \rightarrow \infty$ 时 $\boldsymbol{w}_1(n) \rightarrow \boldsymbol{q}_1$

第一个主分量系数 $y_1(n)=\boldsymbol{q}_1^T \boldsymbol{x}(n)$

迭代时用： $y_1(n)=\boldsymbol{w}_1^T(n)\boldsymbol{x}(n)=\boldsymbol{x}^T(n)\boldsymbol{w}_1(n)$

优化的目标函数为 $E\left[y_1^2(n)\right]=E\left[\left|\boldsymbol{w}_1^T(n)\boldsymbol{x}(n)\right|^2\right]$

PCA的在线算法（续）

在线算法中，目标函数采用

$$J(n)=y_1^2(n)=\left|\boldsymbol{w}_1^T(n)\boldsymbol{x}(n)\right|^2=\boldsymbol{w}_1^T(n)\boldsymbol{x}(n)\boldsymbol{x}^T(n)\boldsymbol{w}_1(n)$$

用SGD算法，权向量更新为

$$\boldsymbol{w}_1(n+1)=\boldsymbol{w}_1(n)+\frac{1}{2}\mu\frac{\partial J(n)}{\partial \boldsymbol{w}_1(n)}=\boldsymbol{w}_1(n)+\mu\boldsymbol{x}(n)\boldsymbol{x}^T(n)\boldsymbol{w}_1(n)$$

$$=\boldsymbol{w}_1(n)+\mu\boldsymbol{x}(n)y_1(n)$$

注：在神经网络的文献中，这类将权系数的更新表示为输入与输出乘积的形式，称为Hebb学习规则，这类算法称为Hebb学习算法

PCA的在线算法（续）

由于PCA中，固定 $\|\boldsymbol{q}_1\|_2 = 1$

故迭代中，每一步固定： $\|\boldsymbol{w}_1(n)\|_2 = 1$

迭代算法修改为：

$$\boldsymbol{w}_1(n+1) = \frac{\boldsymbol{w}_1(n) + \mu \boldsymbol{x}(n) y_1(n)}{\|\boldsymbol{w}_1(n) + \mu \boldsymbol{x}(n) y_1(n)\|_2}$$

上式可近似为（留作练习）

$$\begin{aligned}\boldsymbol{w}_1(n+1) &= \boldsymbol{w}_1(n) + \mu y_1(n)\left(\boldsymbol{x}(n) - y_1(n)\boldsymbol{w}_1(n)\right) \\ &= \boldsymbol{w}_1(n) - \Delta \boldsymbol{w}_1(n)\end{aligned}$$

其中 $\Delta \boldsymbol{w}_1(n) = \mu y_1(n)\boldsymbol{x}(n) - \mu y_1^2(n)\boldsymbol{w}_1(n)$

PCA在线算法--广义Hebb算法（Generalized Hebbian Algorithm，GHA）

推广到有K个主分量

初始化：$\boldsymbol{w}_j(0)$， $j=1,2,\cdots,K$ ，取小的随机数，构成 K 随机向量，分别赋予 $\boldsymbol{w}_j(0)$ 令$n=1$

循环起始：对 $j=1,2,\cdots,K$ 计算

$$y_j(n)=\boldsymbol{w}_j^T(n)\boldsymbol{x}(n)$$

$$\Delta\boldsymbol{w}_j(n)=\mu y_j(n)\left(\boldsymbol{x}(n)-\sum_{k=1}^{j}y_k(n)\boldsymbol{w}_k(n)\right)$$

$$\boldsymbol{w}_j(n+1)=\boldsymbol{w}_j(n)+\Delta\boldsymbol{w}_j(n)$$

令 $n=n+1$，取 $\boldsymbol{x}(n)$ ，返回循环起始，直到结束

3. 盲源分离和ICA

Independent Component Analysis，ICA

存在独立源分量

$$\boldsymbol{s}(n) = [s_1(n), s_2(n), \cdots, s_N(n)]^T$$

经过一个混合系统，产生可测量到的向量

$$\boldsymbol{x}(n) = [x_1(n), x_2(n), \cdots, x_M(n)]^T$$

混合系统表示为

$$\boldsymbol{x}(n) = \boldsymbol{f}\left(\boldsymbol{s}(n), \boldsymbol{s}(n-1), \cdots, \boldsymbol{s}(n-L), n\right)$$

最简单的混合系统 $\boldsymbol{x}(n) = \mathbf{A}\boldsymbol{s}(n)$

<u>混合系统未知，由测量向量估计源向量，欠定问题</u>

独立分量分析：ICA

假设各源分量是统计独立的，即

$$p_{\mathbf{s}}(\mathbf{s}) = p_{s_1}(s_1)p_{s_2}(s_2)\cdots p_{s_N}(s_N)$$

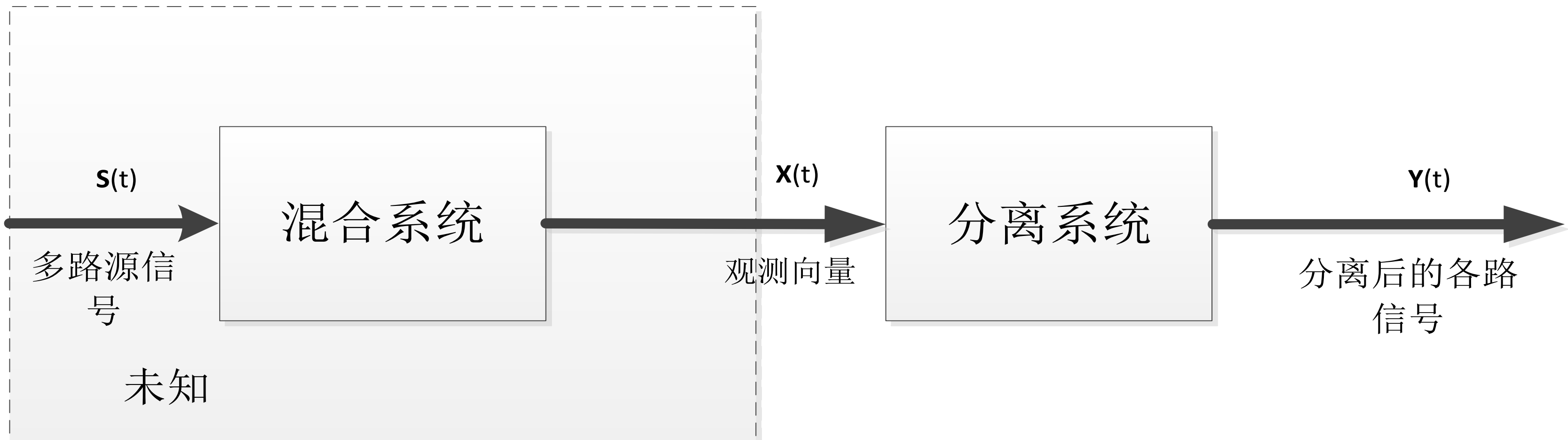
ICA定义为求解如下优化问题

$$\max_{\mathbf{W}}\left\{J_{indep}\left(\boldsymbol{y} = \mathbf{W}\boldsymbol{x}(n)\right)\right\}$$

$J_{indep}(\boldsymbol{y})$ 是描述 $\boldsymbol{y}$ 的独立性的度量函数，$\boldsymbol{y}$ 是 $\boldsymbol{s}$ 的估计

给出样本集，$\{\boldsymbol{x}(n),\ n=1, 2\cdots, \mathrm{T}\}$，首先学习 $\mathbf{W}$

在线ICA系统框图



ICA的常用算法

- 不动点算法-Fast-ICA
- 自然梯度算法
- 最大似然ICA算法
- 信息最大化ICA算法
- 非线性PCA算法
- 稀疏ICA算法
- 等等

4. 不动点算法-Fast-ICA算法简介

讨论抽取一个独立分量 $y = \mathbf{w}^T \mathbf{x}$ 且 $\|\mathbf{w}\| = 1$

目标函数是输出非高斯性最大化，目标函数为

$$J(\mathbf{w}) = E\left[G(\mathbf{w}^T \mathbf{x})\right] + \lambda\left(\mathbf{w}^T \mathbf{w} - 1\right)$$

$G(\cdot)$ 是一个选定的非线性函数

利用牛顿迭代算法，得到

$$\mathbf{w}^+ = E\left[g'(\mathbf{w}^T \mathbf{x})\right]\mathbf{w} - E\left[\mathbf{x} g(\mathbf{w}^T \mathbf{x})\right]$$

$$\mathbf{w} = \frac{\mathbf{w}^+}{\|\mathbf{w}^+\|}$$

Fast-ICA的几个推荐非线性函数

非线性函数 $G(y)$	1阶导数 $g(y)$	2阶导数 $g'(y)$
$\frac{1}{a}\log\cosh(ay)$ $1 \le a \le 2$	$\tanh(ay)$	$a\left(1-\tanh^2(ay)\right)$
$-\exp(-y^2/2)$	$y\exp(-y^2/2)$	$(1-y^2)\exp(-y^2/2)$
$y^4/4$	y^3	$3y^2$

Fast-ICA算法描述

初始步：观测数据向量首先白化，$\mathbf{x}$ 是白化向量，确定，$K \le M$ ，$k=1$

第1步，选择范数为1的随机初始权向量 $\mathbf{w}_k$

第2步，迭代计算

$$\mathbf{w}_k^{+} = E\left[g'(\mathbf{w}_k^T \mathbf{x})\right] \mathbf{w}_k - E\left[\mathbf{x} g(\mathbf{w}_k^T \mathbf{x})\right]$$

$$\mathbf{w}_k^{++} = \mathbf{w}_k^{+} - \sum_{j=1}^{k-1} \left(\mathbf{w}_k^{+T} \mathbf{w}_j\right) \mathbf{w}_j \qquad \mathbf{w}_k = \frac{\mathbf{w}_k^{++}}{\left\| \mathbf{w}_k^{++} \right\|}$$

第3步，若 $\mathbf{w}_k$ 尚未收敛，返回第2步

第4步，若 $k < K$ 　$k \leftarrow k+1$ ，返回第1步

隐变量分析问题

- 混合模型的参数估计应用了离散隐变量方法
- PCA和ICA等方法实际是一种连续隐变量方法。
- 隐变量问题，是机器学习、现代统计学和信号处理中公共关注的一个问题。

本章附录：向量样本的白化

由样本估计相关矩阵 $\mathbf{R}_\mathbf{x}$，并进行特征分解，得

$$\Lambda = diag\{\lambda_1, \cdots, \lambda_M\}$$

$$\mathbf{Q} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_M]$$

定义变换矩阵

$$\mathbf{T} = \Lambda^{-1/2}\mathbf{Q}^\mathrm{T}$$

则 $\boldsymbol{z}(n) = \mathbf{T}\boldsymbol{x}(n) = \Lambda^{-1/2}\mathbf{Q}^\mathrm{T}\boldsymbol{x}(n)$ 是白化的，即

$$E[\boldsymbol{z}(n)\boldsymbol{z}^\mathrm{T}(n)] = E[\Lambda^{-1/2}\mathbf{Q}^\mathrm{T}\boldsymbol{x}(n)\boldsymbol{x}^\mathrm{T}(n)\mathbf{Q}\Lambda^{-1/2}] = \Lambda^{-1/2}\mathbf{Q}^\mathrm{T}\mathbf{R}_\mathbf{x}\mathbf{Q}\Lambda^{-1/2}$$

$$= \Lambda^{-1/2}\mathbf{Q}^\mathrm{T}\mathbf{Q}\Lambda\mathbf{Q}^\mathrm{T}\mathbf{Q}\Lambda^{-1/2} = \mathbf{I}$$

在许多机器学习应用中，预白化是有效的预处理