

机器学习笔记3

qhy

2017 年 7 月 31 日

目录

1	径向基函数网络 Radial Basis Function RBF(样本中心 c_i 这个还需要看一看)	2
2	极限学习机 Extream Learning Machine ELM	2
2.1	简介	2
2.2	原理	2
2.2.1	人工神经网络应用的理论基础	2
2.2.2	人工神经网络学习的问题核心	2
2.2.3	ELM理论	3
2.3	例子	3
3	降维与度量学习	4
3.1	k近邻学习 k-Nearest Neighbor kNN	4
3.1.1	简介	4
3.1.2	懒惰学习与急切学习	4
3.1.3	学习效果分析	4
3.1.4	维数灾难 curse of dimensionality	5
3.2	多维缩放 Multiple Dimensional Scaling MDS	5
3.2.1	低维嵌入	5
3.2.2	MDS 推导过程	5
3.2.3	MDS 算法	8
3.3	主成分分析 Principal Component Analysis PAC	8
3.3.1	线性降维方法	8
3.3.2	PAC 简介	8
3.3.3	基于最近重构性的推导	8
3.3.4	基于最大可分性的推导	8
3.3.5	PAC 算法	8
3.4	核主成分分析 Kernelized PAC KPAC	8
3.4.1	KPCA 简介	8
3.4.2	KPCA 推导	9
4	核函数	9

1 径向基函数网络 Radial Basis Function RBF(样本中心 c_i 这个还需要看一看)

RBF网络,是一种单隐层前馈神经网络,它使用径向基函数作为隐层神经元的激活函数,而输出层则是对隐层神经元输出的线性组合。假定输入为 d 维向量 x ,输出为实值,则RBF网络可表示为:

$$\varphi(x) = \sum_{i=1}^q \omega_i \rho(x, c_i) \quad (1)$$

其中 q 为隐层神经元个数, c_i 和 w_i 分别是第 i 个隐层神经元对应的中心和权重, $\rho(x, c_i)$ 是径向基函数,这是某种沿径向对称的标量函数,通常定义为样本 x 到数据中心 c_i 之间欧氏距离的单调函数。常用的高斯径向基函数形如:

$$\rho(x, c_i) = e^{-\beta_i \|x - c_i\|^2} \quad (2)$$

[Park and Sandberg,1991]证明,具有足够多的隐层神经元的RBF网络能以任意精度逼近任意连续函数。

通常采用两步过程来训练RBF网络:

1. 确定神经元的中心 c_i ,常用的方式包括随机采样,聚类等
2. 利用BP算法等来确定参数 ω_i 和 β_i

简单地说,就是使用径向基函数作为隐层的激活函数的单隐层网络,(径向基函数初被用来进行精确的函数内插),每一个径向基函数局部地逼近目标函数的一部分。

2 极限学习机 Extream Learning Machine ELM

2.1 简介

极限学习机是求解单隐层神经网络的算法。ELM最大的特点,就是对于传统的神经网络,在保证学习精度的前提下比传统的学习算法速度更快。

该算法在网络参数的确定过程中,除了输出层的权重,其他参数在训练过程中都不进行调整,并且,其初始值也是随机的,而网络外权(即输出权)则是通过最小化均方误差得到的最小二乘解。这样的网络在参数确定的过程中就不需要任何迭代的步骤,从而大大降低了网络参数的调节时间。

2.2 原理

2.2.1 人工神经网络应用的理论基础

- 万能逼近能力定理 (Universal Approximation capability) [Leshno 1993, Park and Sandberg 1991, Chen, et al 1995]: 任何连续目标函数可以用前馈神经网络以任意小的误差近似逼近。
- 分类能力定理 (Classification capability) [Huang, et al 2000]: 任何理论上可以分开的目标都可以用人工 (前馈) 神经网络加以分开。

2.2.2 人工神经网络学习的问题核心

- 这些理论都只是在理论上回答了存在性: 给定某种应用存在能提供所需解决能力的对应的人工神经网络结构。
- 但这两个定理没有能回答和提供学习的方法 (网络结构和算法)。

2.2.3 ELM理论

面对如此多的神经网络结构,真的需要如此多的不同的对应神经网络算法?

- 不同的前馈网络结构(不仅限于Sigmoid/RBF网络)

- sigmoid networks
- RBF networks
- polynomial networks
- complex (domain) networks(复变域网络)
- fourier series(傅立叶变换)
- wavelet networks(小波网络),etc

- 多层网络(不仅限于多层网络,包括层次性网络)

生物脑(生物学习系统)中有几十到几百种不同神经元,(其数学模型未知),问题是在生物学习中真的需要调整各种生物神经元?

ELM学习理论:(机器或生物)学习可以不需要调整隐层节点,给定任何连续目标函数或可分类目标,只要前馈神经的隐层节点是非线性阶段连续的,神经网络无需调整隐层节点就能任意逼近目标连续函数或对分类目标加以分类。

2.3 例子

对于 m 个的训练样本 (\mathbf{x}_j, t_j) , $\mathbf{x}_j \in \mathbf{R}^n$, $t_j \in \mathbf{R}$,具有 L 个隐层神经元的单隐层神经网络(图1),其输出的函数表达式为:

$$o_j = \sum_{i=1}^L \beta_i G(\mathbf{a}_i, b_i, \mathbf{x}_j) \quad (3)$$

其中, $G(\mathbf{a}_i, b_i, \mathbf{x})$ 为激活函数, β_i 为输出权, $\mathbf{a}_i \in \mathbf{R}^n$ 为输入权, b_i 为偏置项

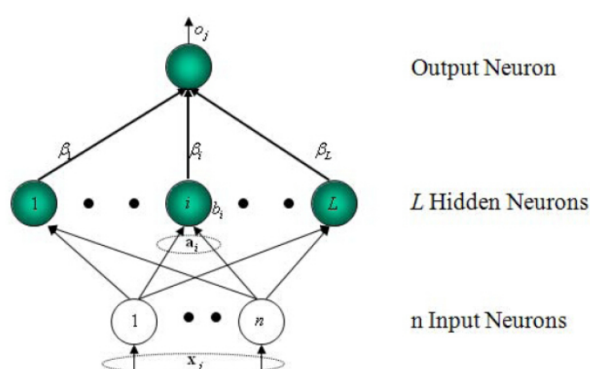


图 1: 单隐层前馈神经网络

单隐层神经网络的输出目标是使得输出的误差最小, 由万能逼近定理,误差可以达到0,可以表示为:

$$E = \sum_{i=1}^N \|o_j - t_j\| = 0 \quad (4)$$

也就是说, $\exists \beta_i, \mathbf{a}_i, b_i$ 使得

$$o_j = \sum_{i=1}^L \beta_i G(\mathbf{a}_i, b_i, x_j) = t_j \quad (5)$$

写成矩阵的形式:

$$H\beta = T \quad (6)$$

H 为隐层节点的输出, β 为输出权重, T 为目标。

$$H = \begin{bmatrix} G(\mathbf{a}_1, b_1, x_1) & \cdots & G(\mathbf{a}_L, b_L, x_1) \\ \vdots & \ddots & \vdots \\ G(\mathbf{a}_1, b_1, x_m) & \cdots & G(\mathbf{a}_L, b_L, x_m) \end{bmatrix}_{m \times L} \quad \beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_L \end{bmatrix}_{L \times 1} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_m^T \end{bmatrix}_{m \times 1} \quad (7)$$

经推导可以得到 $\hat{\beta} = \arg \min_{\beta} \|H\beta - T\| = H^+T$ 这个结论怎么得到的?还需要去推导

3 降维与度量学习

3.1 k近邻学习 k-Nearest Neighbor kNN

3.1.1 简介

k近邻(*k-Nearest Neighbor*, 简称*kNN*)学习是一种常用的监督学习方法,其工作机制非常简单:给定测试样本,基于某种距离度量找出训练集中与其最靠近的k个训练样本,然后基于这k个“邻居”的信息进行预测。

在分类任务中,可使用投票法,即选择这k个样本中出现最多的类别标记作为预测结果。

在回归任务中,可使用平均法,即将这k个样本的实值输出标记的平均值作为预测结果。

还可以基于距离远近进行加权平均或加权投票,距离越近的样本权重越大。

3.1.2 懒惰学习与急切学习

与其他的学习方法相比, k近邻学习有一个明显的不同之处, 它没有显示的训练过程。这称为懒惰学习 *lazy learning*即在训练阶段, 它仅仅是把样本保存起来, 训练时间开销为0,待收到测试样本之后再进行处理。

相应的,那些在训练阶段就对样本进行学习处理的方法,称为急切学习 *eager learning*

3.1.3 学习效果分析

给定测试样本 \mathbf{x} ,若其最近邻样本为 \mathbf{z} ,则最近邻分类器出错的概率就是 \mathbf{x} 与 \mathbf{z} 类别标记不同的概率,即

$$P(err) = 1 - \sum_{c \in y} P(c|\mathbf{x})P(c|\mathbf{z}) \quad (8)$$

为啥是用后验概率呀?

假设样本独立同分布,且对任意 \mathbf{x} 和任意小的 δ ,在 \mathbf{x} 附近 δ 距离范围内总能找到一个训练样本;换言之,对任意测试样本,总能在任意近的范围找到式(8)中的训练样本 \mathbf{z} 。

令 $c^* = \arg \max_{c \in y} P(c|x)$ 表示贝叶斯最优分类器的结果,有

$$P(err) = 1 - \sum_{c \in y} P(c|\mathbf{x})P(c|\mathbf{z}) \quad (9)$$

$$\simeq 1 - \sum_{c \in y} P^2(c|\mathbf{x}) \quad (10)$$

$$\leq 1 - P^2(c^*|\mathbf{x}) \quad (11)$$

$$= (1 + P(c^*|\mathbf{x}))(1 - P(c^*|\mathbf{x})) \quad (12)$$

$$\leq 2 \times (1 - P(c^*|\mathbf{x})) \quad (13)$$

贝叶斯最优分类器需要去看

也就是说,最近邻分类器它的繁华错误不超过贝叶斯最优分类器的错误率的两倍。

3.1.4 维数灾难 curse of dimensionality

上一节的讨论是基于一个重要的假设:任意测试样本 \mathbf{x} 附近任意小的 δ 距离范围内总能找到一个训练样本。也就是说,训练样本的采样密度要足够大,或称为**密采样**

然而,这个假设在现实任务中通常很难满足,现实应用中的属性维数经常成千上万,要满足密采样条件所需的样本数目是无法达到的天文数字。

此外,许多学习算法都涉及距离计算,而高维度空间会给距离计算带来很大的麻烦。

事实上,在高维情况下,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 curse of dimensionality**

另一方面,在高维的情况,为达到好的训练效果,那么就需要更多的样本以适应更多的属性;

解决维度灾难的一个重要途径是**降维 dimension reduction**,亦称为**维数简约**,即通过某种数学变换,将原始高维属性空间转换为一个低维子空间。

3.2 多维缩放 Multiple Dimensional Scaling MDS

3.2.1 低维嵌入

为什么能进行降维? 在很多时候,人们观测或者收集到的数据样本虽是高维的,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**。图2是一个直观的例子。

根据上面的说法,最简单的降维思路,就是人工把不需要的属性去掉,只取认为有用的属性进行学习

但是,大部分情况我们并不知道哪些属性,后面的算法中,根据什么东西来选择哪些属性,这样选择的意义在哪里,为什么这么选择能得到需要的低维样本,需要进一步的思考

3.2.2 MDS 推导过程

MDS是一种经典的降维方法,它的目的是,尽可能保证原始空间中样本的距离在低维空间中得以保持。也就是说,样本“投影”到低维空间之后,样本之间的距离尽可能地保持一样。

下面是推导过程: 假设 m 个样本在原始空间的距离矩阵为 $D \in \mathbb{R}^{m \times m}$,第 i 行第 j 列的元素 $dist_{ij}$ 为样本 \mathbf{x}_i 到 \mathbf{x}_j 的距离。我们的目标是获得样本在 d' 维空间的表示 $Z \in \mathbb{R}^{d' \times m}$, $d' \leq d$ 且任意两个样本在 d' 维空间中的欧式距离等于原始空间中的距离,即 $\|z_i - z_j\| = dist_{ij}$

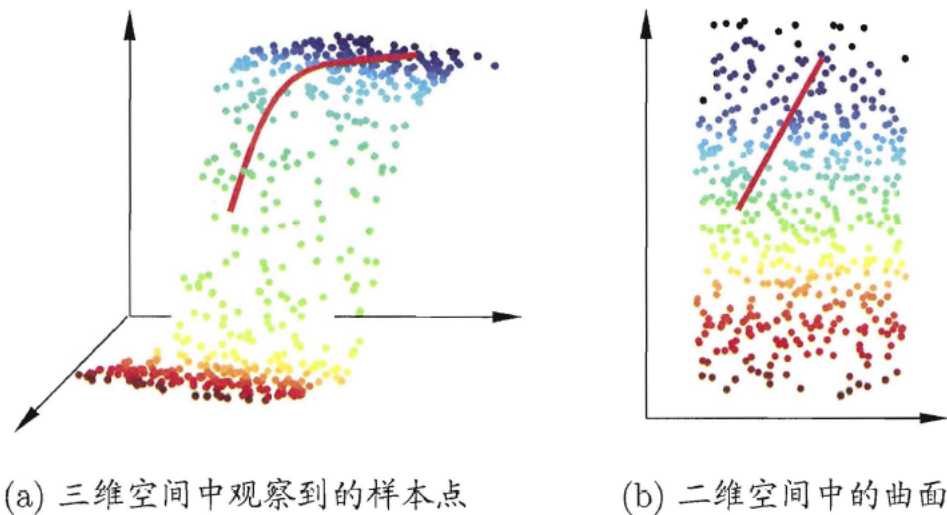


图 2: 低维嵌入示意图

是否一定存在这样的 d' 维空间?比如四面体怎么嵌入到一个二维平面?貌似无法保持边与边的关系,后面实验看看是什么情况?

令 $\mathbf{B} = \mathbf{Z}^T \mathbf{Z} \in \mathbb{R}^{m \times m}$,其中 \mathbf{B} 为降维后样本的内积矩阵, $b_{ij} = \mathbf{z}_i^T \mathbf{z}_j$,有

$$dist_{ij}^2 = \|\mathbf{z}_i\|^2 + \|\mathbf{z}_j\|^2 - 2\mathbf{z}_i^T \mathbf{z}_j \quad (14)$$

$$= b_{ii} + b_{jj} - 2b_{ij} \quad (15)$$

为方便讨论,令降维后的样本被中心化,即 $\sum_{i=1}^m \mathbf{z}_i = 0$,显然矩阵 \mathbf{B} 的行和列之后也均为零。即:

$$\sum_{i=1}^m b_{ij} = \sum_{i=1}^m \mathbf{z}_i^T \mathbf{z}_j = \left(\sum_{i=1}^m \mathbf{z}_i^T \right) \mathbf{z}_j = 0 \quad (16)$$

$$\sum_{j=1}^m b_{ij} = \sum_{j=1}^m \mathbf{z}_i^T \mathbf{z}_j = \mathbf{z}_i \left(\sum_{j=1}^m \mathbf{z}_j^T \right) = 0 \quad (17)$$

易知:

$$\sum_{i=1}^m dist_{ij}^2 = tr(\mathbf{B}) + mb_{jj} \quad (18)$$

$$\sum_{j=1}^m dist_{ij}^2 = tr(\mathbf{B}) + mb_{ii} \quad (19)$$

$$\sum_{i=1}^m \sum_{j=1}^m dist_{ij}^2 = 2mtr(\mathbf{B}) \quad (20)$$

其中, $tr(\mathbf{B}) = \sum_{i=1}^m b_{ii} = \sum_{i=1}^m \|\mathbf{z}_i\|^2$

由式(15) (18) (19) (20)得:

$$b_{ij} = -\frac{1}{2} (dist_{ij}^2 - b_{ii} - b_{jj}) \quad (21)$$

$$= -\frac{1}{2} \left(dist_{ij}^2 - \frac{\sum_{k=1}^m dist_{ik}^2 - tr(B)}{m} - \frac{\sum_{k=1}^m dist_{kj}^2 - tr(B)}{m} \right) \quad (22)$$

$$= -\frac{1}{2} \left(dist_{ij}^2 - \frac{\sum_{k=1}^m dist_{ik}^2}{m} - \frac{\sum_{k=1}^m dist_{kj}^2}{m} + \frac{2tr(B)}{m} \right) \quad (23)$$

$$= -\frac{1}{2} (dist_{ij}^2 - dist_{i.}^2 - dist_{.j}^2 + dist_{..}^2) \quad (24)$$

其中

$$dist_{i.}^2 = \frac{1}{m} \sum_{k=1}^m dist_{ik}^2 \quad (25)$$

$$dist_{.j}^2 = \frac{1}{m} \sum_{k=1}^m dist_{kj}^2 \quad (26)$$

$$\begin{aligned} dist_{..}^2 &= \frac{1}{m^2} \sum_{s=1}^m \sum_{t=1}^m dist_{st}^2 \\ &= \frac{1}{m^2} \sum_{s=1}^m \sum_{t=1}^m (b_{ss} + b_{tt} - 2b_{st}) \\ &= \frac{1}{m^2} \sum_{s=1}^m \sum_{t=1}^m (b_{ss} + b_{tt}) \\ &= \frac{1}{m^2} \left(\sum_{t=1}^m \sum_{s=1}^m b_{ss} + \sum_{s=1}^m \sum_{t=1}^m b_{tt} \right) \\ &= \frac{1}{m^2} \sum_{t=1}^m 2mb_{tt} \\ &= \frac{2}{m} \sum_{i=1}^m \|\mathbf{z}_i\|^2 \\ &= \frac{2}{m} tr(\mathbf{B}) \end{aligned} \quad (27)$$

由此, \mathbf{B} 矩阵就能通过样本的距离矩阵确定对 \mathbf{B} 矩阵进行特征值分解: $\mathbf{B} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ 其中 $\mathbf{\Lambda} = diag(\lambda_1, \lambda_2, \dots, \lambda_d)$ 为特征值构成的对角矩阵, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$ $\mathbf{\Lambda}$ 为特征向量矩阵。

$$\mathbf{B} = \mathbf{Z}^T \mathbf{Z}$$

假定其中有 d^* 个非零特征值,它们构成对角矩阵 $\mathbf{\Lambda}_* = diag(\lambda_1, \lambda_2, \dots, \lambda_{d^*})$,令 \mathbf{V}_* 表示相应的特征向量矩阵,则 \mathbf{Z} 可以表示为

$$\mathbf{Z} = \mathbf{\Lambda}_*^{1/2} \mathbf{V}_*^T \in \mathbb{R}^{d^* \times m}$$

特征值为0的情况,是不是说明样本中,有某两个维度表示的是同一个属性?

在现实应用中,为了有效降维,往往仅需降维后距离与原始空间距离尽可能相近,而不必严格相等。

此时,可以取 $d' \ll d$ 个最大特征值构成对角矩阵, $\tilde{\mathbf{\Lambda}} = diag(\lambda_1, \lambda_2, \dots, \lambda_{d'})$,令 $\tilde{\mathbf{V}}$ 表示相应的特征向量矩阵,则 \mathbf{Z} 可以表示为

$$\mathbf{Z} = \tilde{\mathbf{\Lambda}}^{1/2} \tilde{\mathbf{V}}^T \in \mathbb{R}^{d' \times m}$$

结果看起来像是直接开方,具体运算细节还需要学习。 特征值越大,是不是说明这个特征向量对矩阵的所占的比重越大?需要查查特征值和特征向量的意义加深理解。

3.2.3 MDS 算法

输入:距离矩阵 $\mathbf{D} \in \mathbb{R}^{m \times m}$,其元素 $dist_{ij}$ 为样本 \mathbf{x}_i 到 \mathbf{x}_j 的距离: 等等,不需要对样本先进行中心化处理吗?

过程:

- 1: 根据式(25),(26),(27) 计算 $dist_{i.}^2, dist_{.j}^2, dist_{..}^2$
- 2: 根据式(24)计算矩阵 \mathbf{B}
- 3: 对矩阵 \mathbf{B} 做特征值分解
- 4: 取 $\tilde{\Lambda}$ 为 d' 个最大特征值所构成的对角矩阵, \tilde{V} 为相应的特征向量矩阵

输出:矩阵 $\tilde{\Lambda}^{1/2} \tilde{V}^T \in \mathbb{R}^{d' \times m}$,每一列是一个样本的低维坐标。西瓜书是 $\tilde{V}^{1/2} \in \mathbb{R}^{m \times d'}$,把两个量交换了位置,不知

3.3 主成分分析 Principal Component Analysis PAC

3.3.1 线性降维方法

一般来说,欲获得低维子空间,最简单的是对原始空间进行线性变换。

给定 d 维空间中的样本 $X = x_1, x_1, \dots, x_m \in \mathbb{R}^{d \times m}$,变换之后得到 $d' \leq d$ 维的空间中的样本:

$$\mathbf{Z} = \mathbf{W}^T \mathbf{X} \quad (28)$$

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。

这里 \mathbf{Z} 的每一维度的坐标,都是原始空间上的所有坐标的一个线性组合,这个值可以看成是新低维空间中,当前前以变换矩阵中某一列为基的值。

矩阵乘法的物理意义:将右边矩阵中的每一列列向量变换到左边矩阵中每一行行向量为基所表示的空间中去

从另一个角度来理解的话,降维是对样本在原始空间上进行一定的转换,再根据某个评价标准(比如特征值的大小),取转换后的样本的一部分维度作为低维空间的组成。

基于线性变换来进行降维的方法,称为**线性降维方法**

3.3.2 PAC 简介

PAC是有种线性降维方法,它要求降维后的样本具有最大的可分性。

3.3.3 基于最近重构性的推导

3.3.4 基于最大可分性的推导

3.3.5 PAC 算法

3.4 核主成分分析 Kernelized PAC KPAC

3.4.1 KPCA 简介

KPCA的想法是,先把数据升维,再进行降维操作。

在降维的时候,我们希望在保持原有特性的情况下,把样本映射到低维空间来达到减小时间开销的目的,但是如果样本的特性不明显,或者是降维之后特性消失了,那么降维之后的学习效果就不会很好。

基于高维空间往往更线性可分的想法,可以考虑把样本先升维到高维空间,使得样本的特性更加明显,再进行降维。

核主成分分析就是把样本映射到了高维空间,再进行降维操作。

3.4.2 KPCA 推导

4 核函数

核函数:高维空间和低维空间的一个桥梁,从效果上看,就是把低维空间的样本映射到了高维空间,高维空间的计算,则是通过核函数来进行,避免了维数灾难的问题。