



DOI:10.13364/j.issn.1672-6510.20200024

基于自然梯度的概率主组件分析在线学习算法

陈亚瑞, 秦智飞

(天津科技大学人工智能学院, 天津 300457)

摘要: 对于主组件分析模型,传统方法利用特征值分解的方法求解,其计算复杂度为 $O(ND^2)$,其中 N 表示数据规模, D 表示数据的维度. 概率主组件分析是从概率的角度理解主组件分析模型,此时可以使用期望最大化算法迭代求解,其计算复杂度为 $O(NDM)$,其中 M 表示选取的维度,对于大规模高维数据可有效提高模型效率. 但是,在参数更新过程中,概率主组件模型的期望最大化算法需要遍历整个数据集才能更新一次参数,参数更新较慢,同时该算法很难扩展到大规模数据集. 本文提出基于自然梯度的概率主组件分析在线学习算法,通过结合概率主组件分析的自然梯度,实现在线增量学习模型. 进一步将降维后的数据通过全连接神经网络进行分类,并通过实验证明,该方法在降维效果及算法运行效率上有明显提高.

关键词: 概率主组件分析; 期望最大化算法; 自然梯度; 在线学习

中图分类号: TP391 **文献标志码:** A **文章编号:** 1672-6510(2021)01-0075-06

Online Learning Algorithm for Probabilistic Principal Component Analysis Based on Natural Gradient

CHEN Yarui, QIN Zhifei

(College of Artificial Intelligence, Tianjin University of Science & Technology, Tianjin 300457, China)

Abstract: For the main component analysis model, the traditional method uses the eigenvalue decomposition method to analyze the model, and the computational complexity is $O(ND^2)$, where N represents the data scale and D represents the data dimension. Probabilistic main component analysis is to understand the analysis model of the main component from the perspective of probability. In this way, the expected maximization algorithm can be used to analyze the model iteratively. The computational complexity is $O(NDM)$, where M represents the selected dimension. However, in the process of parameter updating, the expectation maximization algorithm of probabilistic main component model needs to go through the whole data set to update the parameters once, which is slow, and it is difficult to extend the algorithm to large-scale data sets. This paper presents an online learning algorithm for probabilistic main component analysis based on natural gradient. After dimension reduction, the data is classified by the fully connected neural network, and experiments indicate that the method can significantly improve the dimension reduction effect and the running time of the algorithm.

Key words: probabilistic main component analysis; expectation maximization algorithm; natural gradient; online learning

数据降维^[1]是指将高维样本数据通过线性或者是非线性变换映射到低维空间,获得高维数据在低维空间的一种表示. 近年来,数据降维在众多领域变得越来越重要,高维的数据通过数据降维可以去除高维空间中不相关或者不重要的属性,减轻维数灾难,能够对数据的分类、压缩以及可视化带来良好效果. 目

前常用的降维方法有线性判别分析^[2](linear discriminant analysis, LDA)、局部线性嵌入^[3](locally linear embedding, LLE)、拉普拉斯特征映射^[3](laplacian eigenmaps, LE)以及主组件分析^[4-5](principal component analysis, PCA). LDA 是一种属于有监督学习降维技术,基本原理是使得投影之后的每一种类别数据

收稿日期: 2020-02-27; 修回日期: 2020-05-11

基金项目: 天津市自然科学基金资助项目(17JCQNJC00400,18JCZDJC32100); 天津科技大学青年教师资助计划项目(2017LG10)

作者简介: 陈亚瑞(1982—),女,河北邢台人,副教授, yrchen@tust.edu.cn

的投影点尽可能接近,而不同类别数据的类别中心之间的距离尽可能大. LDA 在降维过程中可以使用类别的先验知识经验,但是 LDA 不适合对非高斯样本分布进行降维,并且最多只能降到类别数减 1 的维度. LLE 属于流形学^[6](manifold learning)的一种. 它将高维数据投影到低维空间中,并保持数据点之间的局部线性关系,其核心思想是每个点都可以由与它相近的多个点的线性组合近似,投影到低维空间之后要保持这种线性重构关系,并且有相同的重构系数. LLE 算法归结为稀疏矩阵^[1]特征分解,计算复杂度相对较小,实现容易,但是算法对最近邻样本数的选择敏感,不同的最近邻数对最后的降维结果有很大影响. LLE 是用局部的角度去构建数据之间的关系,它的直观思想是希望相互间有关系的点(在图中相连的点)在降维后的空间中尽可能靠近,可以反映出数据内在的流形结构,但是局部特征保留特性使得它对孤立点和噪声不敏感. PCA 的主要思想是将 n 维特征映射到 k 维上,这 k 维是全新的正交特征,也被称为主组件,是在原有 n 维特征的基础上重新构造出来的 k 维特征,消除了特征之间的多重共线性. 通过计算数据矩阵的协方差矩阵,然后得到协方差矩阵特征值的特征向量,选择特征值最大(即方差最大)的 k 个特征所对应的特征向量组成的矩阵. 这样就可以将数据矩阵转换到新的空间当中,实现数据特征的降维. 但是 PCA 在求解时需要计算数据的协方差矩阵,过程较为繁琐. 概率主组件分析^[7-9](probabilistic principal component analysis, PPCA)是从概率角度理解 PCA,将 PCA 纳入生成式框架,与传统的 PCA 相比,PPCA 属于隐变量模型^[10],可以使用期望最大化^[11](expectation maximization, EM)算法求解,避免了计算数据协方差矩阵. 概率模型与 EM 的结合能够处理数据集里的缺失值问题. PPCA 也可以用一种生成式的方式运行,生成新的样本. 但是通过传统 EM 算法求解 PPCA 模型时会存在参数更新过慢的问题.

因此,本文首先介绍 PPCA,针对 PPCA 存在的问题提出基于自然梯度的概率主组件分析在线学习算法,结合实验验证该算法的有效性.

1 PPCA 及 EM 算法

1.1 概率主组件分析 (PPCA)

对于生成模型^[12] $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$, 其中 \mathbf{x} 表

示观测向量, \mathbf{z} 表示连续隐向量, $p(\mathbf{z})$ 表示隐向量先验概率分布, 一般情况下先验概率分布为 $p(\mathbf{z}) = N(\mathbf{z}|0, \mathbf{I})$, $p(\mathbf{x}|\mathbf{z})$ 表示条件概率分布. 在生成模型结构下, 观测样本 \mathbf{x}_i 的生成过程如下: 首先从隐变量先验分布 $p(\mathbf{z})$ 中随机生成一个隐向量样本 \mathbf{z}_i , 然后根据条件概率分布 $p(\mathbf{x}|\mathbf{z})$ 生成样本 \mathbf{x}_i . 生成过程的概率表示形式为

$$\mathbf{z} \sim N(0, \mathbf{I}) \tag{1}$$

$$\mathbf{x}|\mathbf{z} \sim p(\mathbf{x}|\mathbf{z}) \tag{2}$$

生成模型 $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$ 对应的概率图模型形式如图 1 所示, 其中白色节点表示隐向量, 黑色节点表示观测向量, 节点之间的边表示变量之间的依赖关系.

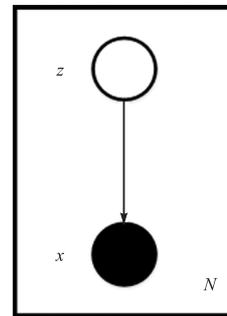


图 1 生成模型
Fig. 1 Generating models

概率主组件分析是生成模型的一种特殊形式, 其中 D 维观测变量 \mathbf{x} 由 M 维隐变量 \mathbf{z} 的一个线性变换附加一个高斯噪声^[13]定义, 即

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\varepsilon} \tag{3}$$

其中 \mathbf{W} 表示 $R^{D \times M}$ 维的一个参数矩阵, $\boldsymbol{\varepsilon}$ 表示 D 维零均值高斯分布的噪声变量, $\boldsymbol{\varepsilon} \sim N(0, \delta^2 \mathbf{I})$.

PPCA 模型的观测样本 \mathbf{x}_i 的生成过程如下: 首先从隐变量先验分布 $p(\mathbf{z})$ 中随机生成一个样本 \mathbf{z}_i , 然后根据条件概率分布 $p(\mathbf{x}|\mathbf{z})$ 生成样本 \mathbf{x}_i , 即

$$p(\mathbf{x}|\mathbf{z}) = N(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \delta^2 \mathbf{I}) \tag{4}$$

对于观测数据集 $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, 其中 N 表示观测样本的个数, $Z = \{\mathbf{z}_1, \dots, \mathbf{z}_N\}$ 为对应的隐变量数据集. PPCA 的学习任务是通过观测数据 $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ 学习模型(4)中的参数 \mathbf{W} 、 δ^2 和 $\boldsymbol{\mu}$.

1.2 EM 算法

EM 算法是求解隐变量模型最大似然问题的经典方法, 其基本思想是通过 E (Expectation) 步和 M (Maximization) 步两步不断迭代更新模型参数. 其中 E 步是通过对完备数据集的对数似然函数

$\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})$ 求关于隐变量后验概率分布 $p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta}^{\text{old}})$ 的期望, 此步骤的目的是通过隐量后验概率分布处理隐变量. M 步是对于在 E 步中的完备数据集对数似然函数的期望进行最大化处理, 更新模型参数.

对于 PPCA 模型, 完备数据集的对数似然函数为

$$\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \mathbf{W}, \delta^2) = \sum_{n=1}^N \{\ln p(\mathbf{x}_n | \mathbf{z}_n) + \ln p(\mathbf{z}_n)\} \quad (5)$$

PPCA 模型采用 EM 算法迭代计算模型参数.

E 步: 对 $\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})$ 求关于 $p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta}^{\text{old}})$ 的期望, 即

$$\begin{aligned} & E[\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \mathbf{W}, \delta^2)] \\ &= -\sum_{n=1}^N \left\{ \frac{D}{2} \ln(2\pi\delta^2) + \frac{1}{2} \text{Tr}(E[\mathbf{z}_n \mathbf{z}_n^T]) + \right. \\ & \quad \left. \frac{1}{2\delta^2} \|\mathbf{x}_n - \boldsymbol{\mu}\|^2 - \frac{1}{\delta^2} E[\mathbf{z}_n]^T \mathbf{W}^T (\mathbf{x}_n - \boldsymbol{\mu}) + \right. \\ & \quad \left. \frac{1}{2\delta^2} \text{Tr}(E[\mathbf{z}_n \mathbf{z}_n^T] \mathbf{W}^T \mathbf{W}) + \frac{M}{2} \ln(2\pi) \right\} \quad (6) \end{aligned}$$

其中计算 $E[\mathbf{z}_n]$ 和 $E[\mathbf{z}_n \mathbf{z}_n^T]$ 是该步骤的关键. 根据后验概率分布计算可得

$$E[\mathbf{z}_n] = \mathbf{M}^{-1} \mathbf{W}^T (\mathbf{x}_n - \bar{\mathbf{x}}) \quad (7)$$

$$E[\mathbf{z}_n \mathbf{z}_n^T] = \delta^2 \mathbf{M}^{-1} + E[\mathbf{z}_n] E[\mathbf{z}_n]^T \quad (8)$$

其中 $\mathbf{M} = \mathbf{W}^T \mathbf{W} + \delta^2 \mathbf{I}$.

M 步: 是最大化 $E[\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \mathbf{W}, \delta^2)]$ 更新模型参数

$$\boldsymbol{\mu}, \mathbf{W}, \delta^2 \leftarrow \arg \max_{\boldsymbol{\mu}, \mathbf{W}, \delta^2} E[\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \mathbf{W}, \delta^2)] \quad (9)$$

进一步求解可得

$$\mathbf{W}_{\text{new}} = \left[\sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}}) E[\mathbf{z}_n]^T \right] \left[\sum_{n=1}^N E[\mathbf{z}_n \mathbf{z}_n^T] \right]^{-1} \quad (10)$$

$$\begin{aligned} \delta_{\text{new}}^2 = & \frac{1}{ND} \sum_{n=1}^N \{ \|\mathbf{x}_n - \bar{\mathbf{x}}\|^2 - 2E[\mathbf{z}_n]^T \mathbf{W}_{\text{new}}^T (\mathbf{x}_n - \bar{\mathbf{x}}) + \\ & \text{Tr}(E[\mathbf{z}_n \mathbf{z}_n^T] \mathbf{W}_{\text{new}}^T \mathbf{W}_{\text{new}}) \} \quad (11) \end{aligned}$$

其中 $\bar{\mathbf{x}} = \sum_{i=1}^N \mathbf{x}_i / N$

PPCA 模型 EM 算法如下:

输入: 观测数据 $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$

repeat

for $n = 1:N$

根据 (7) 和 (8) 计算隐变量的期望 $E[\mathbf{z}_n]$, $E[\mathbf{z}_n \mathbf{z}_n^T]$.

End

根据 (10) 和 (11) 更新参数的值, 得到 \mathbf{W}_{new} 和 δ_{new}^2 .

until 参数 \mathbf{W}_{new} 和 δ_{new}^2 收敛

return 参数 \mathbf{W}_{new} 和 δ_{new}^2

1.3 PPCA 算法存在的问题

根据 PPCA 模型的 EM 算法描述, 在进行一次参数更新的时候, 需要首先计算出所有样本的 $E[\mathbf{z}_n]$ 和 $E[\mathbf{z}_n \mathbf{z}_n^T]$, 然后再更新模型参数 \mathbf{W}_{new} 和 δ_{new}^2 . 该算法最大问题是每更新一次参数需要计算所有样本的隐变量后验概率分布的期望, 会导致计算过于复杂, 参数更新太慢; 同时算法很难扩展到大规模数据集, 因为每一次算法更新时都需要遍历整个数据集是不现实的. 进一步, 当数据量不断增加时, 需要采用增量学习^[14]算法, 随着数据量增加不断提升算法的性能. 基于以上问题, 本文提出基于自然梯度的概率主组件分析在线学习^[15]算法.

2 自然梯度

梯度上升方法^[16]是求解优化问题的重要方法, 它通过沿着目标函数梯度的方向进行参数迭代更新, 求解最优化问题. 在欧氏空间中, 函数梯度方向是变化最快的方向, 也是参数更新的最优方向. 但是当优化的参数是概率分布时, 欧氏距离不能有效地度量概率分布之间的距离. 对称 KL 散度^[17] (KL divergence) 是度量两个概率分布之间相似性的一种度量方式, 数学表述为

$$D_{\text{KL}}^{\text{sym}}(\lambda, \lambda') = E_{\lambda} \left[\log \frac{q(\beta | \lambda)}{q(\beta | \lambda')} \right] + E_{\lambda'} \left[\log \frac{q(\beta | \lambda')}{q(\beta | \lambda)} \right] \quad (12)$$

欧氏梯度是欧氏空间中上升最快的方向, 而自然梯度^[18]是黎曼空间 (该空间采用对称 KL 散度度量局部距离) 内上升最快的方向. 对于目标函数 $f(\lambda)$, 它的自然梯度为

$$\tilde{\nabla}_{\lambda} f(\lambda) \triangleq \mathbf{G}(\lambda)^{-1} \nabla_{\lambda} f(\lambda) \quad (13)$$

其中 $\tilde{\nabla}_{\lambda} f(\lambda)$ 表示目标函数的自然梯度, $\nabla_{\lambda} f(\lambda)$ 表示目标函数的一般梯度, $\mathbf{G}(\lambda)$ 表示目标函数的 fisher 信息矩阵 (fisher information matrix)^[19].

3 本文算法

对于 PPCA 模型的 EM 优化算法, 其中的最大化问题 $\max E[\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \mathbf{W}, \delta^2)]$, 采用自然梯度上升方法求解该优化问题. 根据自然梯度 (式 (13)) 及平均值场^[7] (mean field) 理论可知

$$\begin{aligned} \mathbf{G}(\lambda) &= E_{\lambda} [(\nabla_{\lambda} \log p(\beta | \lambda)) (\nabla_{\lambda} \log p(\beta | \lambda))^T] \\ &= E_{\lambda} [(t(\beta) - E_{\lambda}[t(\beta)]) (t(\beta) - E_{\lambda}[t(\beta)])^T] \\ &= \nabla_{\lambda}^2 a_g(\lambda) \quad (14) \end{aligned}$$

可以得到自然梯度具有以下简单形式

$$\tilde{\nabla}_\lambda L = E_\phi[\eta_g(\mathbf{x}, \mathbf{z}, \boldsymbol{\alpha})] - \lambda \quad (15)$$

在传统的PPCA的EM算法中,每一次参数的迭代更新都需要计算所有样本的后验概率分布,会使得算法的性能变得很差.在基于自然梯度的概率主组件分析在线学习算法中,使用单个样本对参数进行局部的更新,并在参数的更新时引入自然梯度.

对于单样本 \mathbf{x}_n ,首先是计算样本所对应的隐变量后验分布

$$E[\mathbf{z}_n] = \mathbf{M}^{-1}\mathbf{W}^T(\mathbf{x}_n - \bar{\mathbf{x}}) \quad (16)$$

$$E[\mathbf{z}_n\mathbf{z}_n^T] = \delta^2\mathbf{M}^{-1} + E[\mathbf{z}_n]E[\mathbf{z}_n]^T \quad (17)$$

然后使用单个样本计算参数 $\tilde{\mathbf{W}}$ 、 $\tilde{\delta}^2$ 、 $\tilde{\boldsymbol{\mu}}$

$$\tilde{\mathbf{W}} = (\mathbf{x} - \boldsymbol{\mu})E[\mathbf{z}](E[\mathbf{z}\mathbf{z}^T])^{-1} \quad (18)$$

$$\tilde{\delta}^2 = \frac{1}{D} \{ \|\mathbf{x} - \boldsymbol{\mu}\|^2 - 2E[\mathbf{z}]^T \tilde{\mathbf{W}}(\mathbf{x} - \boldsymbol{\mu}) + \text{Tr}(E[\mathbf{z}\mathbf{z}^T] \tilde{\mathbf{W}}^T \tilde{\mathbf{W}}) \} \quad (19)$$

$$\tilde{\boldsymbol{\mu}} = \mathbf{x} \quad (20)$$

此时根据梯度下降算法,更新模型参数

$$\mathbf{W}_{\text{new}} = \mathbf{W}_{\text{old}} + \rho(\tilde{\mathbf{W}} - \mathbf{W}_{\text{old}}) \quad (21)$$

$$\delta_{\text{new}}^2 = \delta_{\text{old}}^2 + \rho(\tilde{\delta}^2 - \delta_{\text{old}}^2) \quad (22)$$

$$\boldsymbol{\mu}_{\text{new}} = \boldsymbol{\mu}_{\text{old}} + \rho(\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}_{\text{old}}) \quad (23)$$

基于自然梯度的概率主组件分析在线学习算法如下:

输入: 观测数据 $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$

对参数 \mathbf{W} 和 δ^2 进行初始化

初始化参数 $0 < \rho < 1$

repeat

for $n = 1 : N$

根据式(16) — (20) 计算隐变量的后验概率分布的期望 $E[\mathbf{z}_n]$ 及参数 $\tilde{\mathbf{W}}$ 、 $\tilde{\delta}^2$ 、 $\tilde{\boldsymbol{\mu}}$

根据式(21) — (23) 更新参数的值,得到 \mathbf{W}_{new} 、

δ_{new}^2 、 $\boldsymbol{\mu}_{\text{new}}$

end

return 参数 \mathbf{W}_{new} 、 \mathbf{W}_{new} 、 $\boldsymbol{\mu}_{\text{new}}$

4 实验分析

本节在5个数据集下设计一组对比实验,在不同训练样本个数下设计一组在线学习过程实验以及在MNIST数据集下设计一组生成样本数据实验,分析实验结果,证明本文提出算法的有效性.

4.1 实验环境

本实验选用了5个数据集,包括MNIST数据

集, Fashion-MNIST数据集和3个UCI数据集(HOP、SDD、AREM).其中:MNIST数据集为手写数字数据集,共有10类;Fashion_MNIST数据集包含有10类不同商品的图片;HOP、SDD及AREM数据集是进行过特征提取后的数据集.数据集具体信息见表1.

表1 数据集信息

Tab. 1 Data set information

数据集	数据维度	类别/类	样本/个
MNIST	784	10	60 000
Fashion_MNSIT	784	10	70 000
AREM	7	4	42 240
SDD	48	11	58 509
HOP	8	4	65 751

实验平台信息如下:GPU采用GTX1080,CPU采用i7-8700@3.30 Hz,运行内存为16 GB,操作系统Windows 7 旗舰版,运行平台工具python3.6.

4.2 实验设计

设计3组实验.实验1(在线学习过程实验):在基于自然梯度的概率主组件分析在线学习算法中,依次使用递增的训练样本个数对模型进行训练,比较分类正确率.实验2(精度对比实验):在5个数据集下分别运行PPCA的EM算法和基于自然梯度的概率主组件分析在线学习算法,并对比分析了算法运行时间及其分类正确率.实验3(生成样本实验):在基于自然梯度的概率主组件分析在线学习算法中,依次使用递增的训练样本个数对模型进行训练,在训练好的模型中使用随机生成的隐变量生成新的样本,比较生成数据的效果.

4.2.1 在线学习过程实验

在基于随机化近似的在线EM算法中,依次使用5 000、10 000、20 000、30 000、40 000、>40 000个数据样本对EM算法进行训练,在训练好的模型下对数据进行降维,并将降维后的数据按照表2所示分为训练集和测试集,通过单隐层全连接神经网络进行分类,batch-size为32,在学习率0.1,epoch为100,得到分类正确比见表3.

表2 分类数据集

Tab. 2 Classified data set

数据集	训练样本/个	测试样本/个
MNIST	55 000	5 000
Fashion_MNSIT	60 000	10 000
AREM	35 000	7 240
SDD	50 000	8 509
HOP	65 000	10 751

表 3 基于不同训练样本个数的分类正确比
Tab. 3 Classification accuracy ratio based on different training samples

数据集	5 000	10 000	20 000	30 000	40 000	>40 000
Mnist	0.487 2	0.501 0	0.574 2	0.630 0	0.670 4	0.708 6
Fashion_MNIST	0.366 7	0.405 6	0.487 7	0.560 8	0.613 8	0.648 6
AREM	0.478 4	0.508 3	0.554 6	0.542 6	0.579 8	0.605 3
SDD	0.606 6	0.644 1	0.706 8	0.734 4	0.772 4	0.786 4
Hop	0.623 4	0.654 1	0.688 1	0.694 2	0.805 1	0.876 8

4.2.2 精度对比实验

在数据集上分别运行 PPCA 的 EM 算法和在训练基于自然梯度的概率主组件分析在线学习算法时, 由于内存的限制, 每个数据集都随机选用 30 000 个样本用于训练 EM 算法, 最大迭代次数设置为 500. 在训练好的模型中, 将 MNIST、Fashion_MNIST 的数据样本降到 200 维, 将 HOP 数据集样本降到 4 维, 将 AREM 数据集降到 5 维, 将 SDD 数据集样本降到 22 维. 将降维之后的数据集按照表 2 所示比例分为训练样本和测试样本, 通过单隐层全连接神经网络进行分类, batch-size 为 32, 学习率 0.1, epoch 为 100, 得到分类正确比见表 4, 算法的运行时间见表 5.

表 4 分类精度

Tab. 4 Classification accuracy

数据集	本文算法	PPCA
MNIST	0.932 6	0.917 0
Fashion_MNIST	0.844 8	0.826 0
AREM	0.730 6	0.728 0
SDD	0.867 1	0.846 2
HOP	0.876 8	0.542 1

表 5 算法运行时间

Tab. 5 Algorithm running time

数据集	本文算法	PPCA
MNIST	115	637
Fashion_MNIST	110	622
AREM	78	456
SDD	117	484
HOP	136	300

4.2.3 生成样本实验

在基于自然梯度的概率主组件分析在线学习算法中, 依次使用 5 000、10 000、20 000、30 000、40 000、>40 000 个 MNIST 数据样本对算法进行训练, 隐变量的维度设置为 2, 从均匀分布中随机生成隐变量 z , 利用概率生成模型生成新的样本, 样本的清晰度比较如图 2 所示. 根据图 2 对比结果可知, 在基于自然梯度的概率主组件分析在线学习算法中, 随着训练样本的增多, 所生成的样本越来越清晰, 生成效果越来越好.

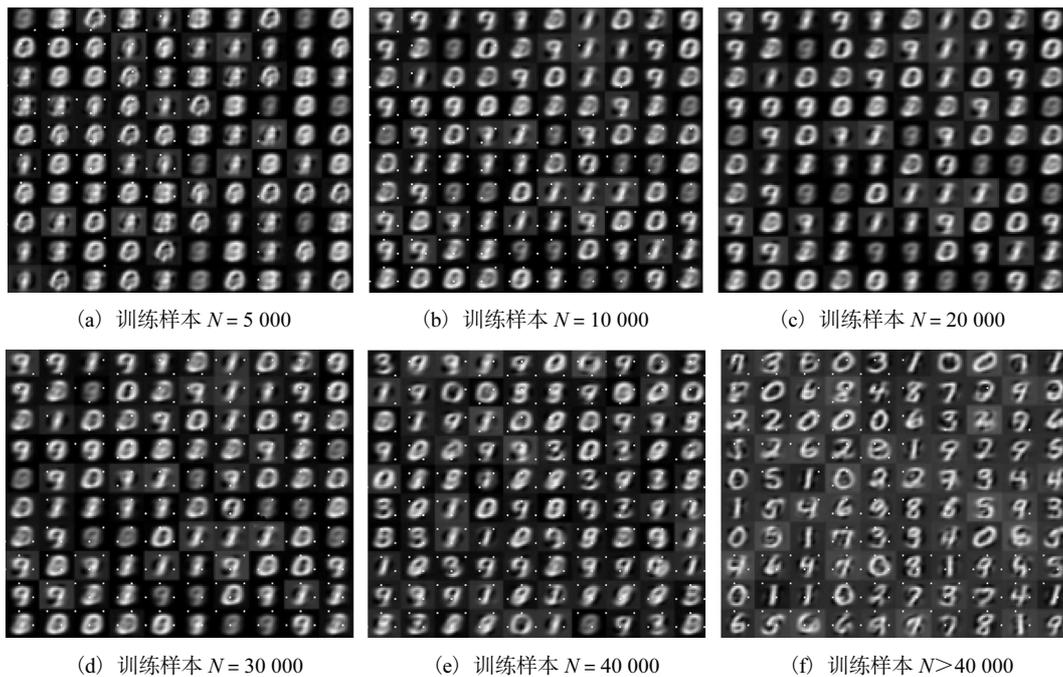


图 2 基于自然梯度的概率主组件分析在线学习算法的生成样本

Fig. 2 Generated samples of online learning algorithm based on natural gradient probability main component analysis

5 结 语

本文提出的基于自然梯度的概率主组件分析在线学习算法将自然梯度引入到 PPCA 的 EM 算法中,提高了算法的效果. 实验证明该算法相比于基于 EM 的传统 PPCA 在分类精度略有提升,但是算法的运行时间大大减少,并且算法受物理环境的影响较小,同时,随着训练样本的增加,算法在生成数据方面也有比较明显的提升.

参考文献:

- [1] 吴晓婷, 闫德勤. 数据降维方法分析与研究[J]. 计算机应用研究, 2009, 26(8): 2832-2835.
- [2] Webb A R. Linear Discriminant Analysis[M]. Hoboken: John Wiley & Sons, 2003.
- [3] Wang J. Geometric Structure of High-Dimensional Data and Dimensionality Reduction[M]. Berlin: Springer, 2012.
- [4] 张绍璞. 基于主成分分析中的几类问题的计算公式及结论[J]. 天津科技大学学报, 2010, 25(2): 80-82.
- [5] 王彦斌, 于德敏, 许增朴, 等. 基于 PCA 的扬声器响应信号特征提取及识别[J]. 天津科技大学学报, 2009, 24(3): 50-53.
- [6] Lin T, Zha H. Riemannian manifold learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30(5): 796-809.
- [7] Duff I S, Grimes R G, Lewis J G. Sparse matrix test problems[J]. ACM Transactions on Mathematical Software, 1989, 15(1): 1-14.
- [8] Bishop C M. Pattern Recognition and Machine Learning (Information Science and Statistics) [M]. Berlin: Springer, 2006.
- [9] Ahmadkhani S, Adibi P. Face recognition using supervised probabilistic principal component analysis mixture model in dimensionality reduction without loss framework[J]. IET Computer Vision, 2016, 10(3): 193-201.
- [10] 王秀美, 高新波, 张乾坤, 等. 一种基于局部保持的隐变量模型[J]. 模式识别与人工智能, 2010(3): 369-375.
- [11] Rissanen J. Minimum Description Length Principle [M]. Hoboken: John Wiley & Sons, 1985.
- [12] 郭立君, 赵杰煜, 史忠植. 生成模型与判别方法相融合的图像分类方法[J]. 电子学报, 2010, 38(5): 151-155.
- [13] 何志昆, 刘光斌, 赵曦晶, 等. 高斯过程回归方法综述[J]. 控制与决策, 2013(8): 3-11.
- [14] He H. Self-Adaptive Systems for Machine Intelligence [M]. Hoboken: John Wiley & Sons, 2011. .
- [15] Yang H, Pan Z, Tao Q, et al. Online learning for vector autoregressive moving-average time series prediction [J]. Neurocomputing, 2018, 315: 9-17.
- [16] 叶晓波, 秦海菲. Newton-Raphson 算法 Logistic 分类器性能提升应用研究[J]. 软件导刊, 2017, 16(11): 145-147.
- [17] Goldberger J, Gordon S, Greenspan H. An efficient image similarity measure based on approximations of KL-divergence between two gaussian mixtures[C]//IEEE. Proceedings Ninth IEEE International Conference on Computer Vision. New York: IEEE, 2003: 1238387.
- [18] Guo W, Wei H, Liu T, et al. An adaptive natural gradient method with adaptive step size in multilayer perceptrons[C]//IEEE. 2017 Chinese Automation Congress (CAC). New York: IEEE, 2007: 8243022.
- [19] Spall J C. Monte carlo computation of the fisher information matrix in nonstandard settings[J]. Journal of Computational & Graphical Statistics, 2005, 14(4): 889-909.

责任编辑: 郎婧