Contents lists available at ScienceDirect

Engineering

journal homepage: www.elsevier.com/locate/eng

Research
Smart Process Manufacturing—Article

一种局部二次嵌入学习算法及其在软测量中的应用

包垚垚，朱远明 *，钱锋 *

Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China

ARTICLE INFO

Article history:
Received 30 December 2019
Revised 25 January 2021
Accepted 29 April 2022
Available online 28 September 2022

关键词
局部二次嵌入
度量学习
回归机
软测量

摘要

鉴于元学习在众多领域取得的巨大成就，本文针对数据回归问题提出了融合度量学习和神经网络（NN）的局部二次嵌入学习（LQEL）算法。首先，通过优化输入输出空间里样本间度量的全局一致性来改进马氏度量（Mahalanobis metric）学习算法；同时，通过引入松弛约束进一步证明了改进的度量学习问题等价于一个凸规划问题。然后，基于局部二次插值假设原理，引入了两个轻量级的神经网络，其一用于学习局部二次模型中的系数矩阵，另一个则用于对从不同局部近邻获得的预测结果进行权重分配。最后，将两个子模型嵌入统一的回归框架中，并通过随机梯度下降（SGD）算法学习模型参数。所提出的算法优势在于可充分利用目标标签中隐含的信息找到更可靠的参考样本。并且，使用LQEL算法对变量进行差分建模，避免了因传感器漂移或不可测量变量导致的模型退化问题。多个基准数据集和两个实际工业应用数据集的计算结果表明，所提出的方法优于几种典型的回归方法。

1. 引言

在水泥生产过程中，产品质量如生料细度、熟料中游离氧化钙含量等都是需要监控的关键指标。然而，这些指标的在线测量仪器往往成本高昂且需要频繁的定期维护。在实际工业应用中，化验室通常每隔两小时或更长时间对这些指标进行离线化验分析，导致控制系统得不到及时的反馈信息。而这类问题往往可通过软测量技术来解决[1–2]。

软测量本质上是一种回归模型，可利用在线可用的其他辅助变量实时评估其质量指标。换言之，给定D维输入变量$X=\{x^{(1)},x^{(2)},x^{(3)},...,x^{(s)}\}$及其相应的输出变量$Y=\{y^{(1)},y^{(2)},y^{(3)},...,y^{(s)}\}$，回归模型的目标是使用训练数据中隐含的知识构造最优映射函数，以在测试集上实现显著的预测精度。目前，软测量技术已在炼油[3]、冶金[4]和能源管理[5–6]等多元化行业中取得成功应用。

软测量模型源自多元统计回归模型，包括线性回归（LR）、主成分回归（PCR）、偏最小二乘（PLS），以及一些具有正则化策略的变种方法，如最小绝对收缩选择算子（LASSO）和岭回归[7]，用来平衡模型的经验误差和复杂性。核函数策略已得到广泛研究，并与上述算法相结合用于解决非线性回归问题[8–9]。此后，众多机器学习方法涌现以实现海量数据的知识挖掘，诸如k最近邻回归（k-NNR）[10]、分类与回归树（CART）[11–12]以及支持向量回归（SVR）[13–14]。为了提高单一树模型的性能，

* Corresponding authors.
E-mail addresses: yuanmingzhu@ecust.edu.cn (Y. Zhu), fqian@ecust.edu.cn (F. Qian).

英文原文：Engineering 2022, 18(11): 186–196
引用本文：Yaoyao Bao, Yuanming Zhu, Feng Qian. A Local Quadratic Embedding Learning Algorithm and Applications for Soft Sensing. *Engineering*, https://doi.org/10.1016/j.eng.2022.04.025

在随机森林（RF）算法中使用了装袋策略[15–16]。类似地，可以通过组合一系列弱学习器[17–18]，如梯度提升（GBM）和极限梯度提升（XGBoost）来提高增强算法的预测精度。此外，深度学习在图像和语音识别方面的突破使神经网络[19–20]成为机器学习领域最流行的方法之一，尤其在数据样本充足的情况下效果更加明显。此类方法之所以流行，原因在于经过特殊设计的神经网络结构具有强大的特征提取能力[21]。

在这些算法中，*k*-NNR是最简单和最流行的回归方法之一。由于*k*-NNR不需要显式的模型结构或任何数据分布的先验知识，因此被广泛用于机器学习问题。但是，使用*k*个最近邻（*k*-NN）样本的平均输出作为预测结果的策略也是该方法最大的缺点。最初，*k*-NNR算法采用欧几里得距离度量来测量样本相似性。然而，输入特征的变化范围可以很大，变量之间的冗余和相关性也可能会产生使人误解的信息，从而导致不切实际的距离度量。为了解决这一问题，已有文献提出了一种更一般化的马氏距离[22]，马氏距离等效于两个线性投影之间的加权欧几里得距离。然而，在实际应用中，输入特征往往对输出变量有不同的贡献。关键在于开发一个可靠的特征提取模型，并将经典度量（如欧几里得距离和余弦相似度）应用到特征映射上。局部线性嵌入（LLE）使用局部线性加权方法在低维空间中重构样本，并通过最小化重构误差来实现降维[23]。然而，在高维空间中由经典欧几里得度量构造的邻近关系不能满足所有分类任务的需求。因此，研究人员尝试将输入特征变换至缩放空间[24–25]，并通过空间中的局部重建获得权重系数来预测标签。但是，这类方法非常依赖变换模型的精巧设计。例如，在模糊变换中，基函数和模糊区间的划分可能对预测结果有很大影响，因为输出标签中包含的有价值的信息没有被充分利用。为了解决这个问题，Weinberger和Saul [26]引入了马氏距离度量学习的概念，利用马氏距离中的逆协方差矩阵表示任意正半定矩阵。类似于线性判别式分析（LDA）[27]的思想，通过最大化平均类内距离与平均类间距离的比率来学习马氏距离度量。Xing等[28]通过将平均类间距离作为优化目标，将平均类内距离作为约束，构造了度量学习的凸优化问题。目前，该方法已应用于处理半监督数据聚类问题。

上述方法主要针对分类问题，而对于回归问题，Nguyen等[27]通过在每个实例附近的一组受约束的三元组上最大化输入和输出距离的一致性，建立了一个凸优化问题。然而，针对变换矩阵$\boldsymbol{A}$，研究人员没有采用度量学习进行求解，只是根据给定的变换矩阵$\boldsymbol{A}$通过优化得到权重矩阵$\boldsymbol{W}$。而且，平衡参数C的选择对算法的性能也有显著影响。线性度量学习（LML）在特征表示方面的能力亦有限，尤其是对于图像和文本数据等高维样本。深度度量学习（DML）则使用深度神经网络（DNN）模型而非线性变换来提取特征，进而可实现度量学习[29–31]。LML和DML之间最大的区别在于损失函数的形式。例如，Song等[30]最小化来自同一类的样本之间的距离，并最大化了来自不同类间样本的距离。一般而言，这些方法涉及三元组的构造，三元组由训练数据样本、同类样本和异类样本组成，意味着这些方法不能被直接应用于回归问题。

此外，使用*k*-NN的平均值作为预测输出通常会带来保守的结果。以加利福尼亚大学欧文分校（UCI）机器学习资料库上的葡萄酒质量评估数据集为例。*k*-NNR算法不能很好地区分特别高级或低级的葡萄酒。那么品酒师是如何进行鉴别的呢？他会先识别历史数据中与当前样本最相似的样例并作为参考，然后根据输入特征的变化修改标签。本文基于此思想提出了一种局部二次嵌入学习（LQEL）算法。虽然二次嵌入函数的系数矩阵往往难以获得，但幸运的是，该矩阵取决于展开点的位置，即当前样本点。因此，可以通过将当前样本输入神经网络进行估计。同时，必须事先确定适当的网络规模，否则模型容易被过度拟合。为此，使用多个神经网络进行集成的方法可提高神经网络模型的泛化能力[20,32]。文献表明，通过批量归一化（BN）标准化网络中隐藏层的输出可以防止训练过程中的分布发生变化[33]，并加速网络的收敛速度，而dropout策略可以提高神经网络的泛化能力[34]。此外，在样本数据上叠加一定强度的高斯噪声可以增加训练样本的数量，从而提高模型的鲁棒性[35]。通常，这些方法可在两方面改进神经网络的泛化能力：其一，增加了训练样本的数量；其二，为网络结构加入了约束，降低其复杂性，从而提升网络的预测能力。本文将采用后一种技术路线。

在本文中，首先利用度量学习通过最大化输入和输出空间之间距离的一致性来确定某个样本的邻域，这充分利用了目标标签中包含的信息。然后，用一个训练好的神经网络生成局部二次系数矩阵，以实现基于近邻参考点的预测；并通过差分补偿方法防止传感器漂移或未测量变量引起的模型退化。进而，用另一个神经网络根据不同近邻数据的置信度为每个预测结果提供预测权重，实现了预测误差和测量噪声之间的平衡，从而最小化预测误差。这两个神经网络的参数可以利用随机梯度下降（SGD）算法的端到端训练来优化。经过对几个回归数据集（包括水泥生产过程和加氢裂化过程的两个实际工业数据集）的实证研究表明，在大部分情况下，所提出的方法优于流行的回归方法。

本文的其余章节安排如下：第2节引入了度量学习模

型，并证明了该优化问题等价于凸优化问题；第3节介绍了LQEL模型框架；第4节列举了几项实证研究，其中包括实际工业验证结果；第5节总结了本文的结论和贡献。

2. 度量学习

度量学习中的度量就是希望能在输入空间中找到一个映射函数$d: X\times X\to R_0^+$，用于衡量输入空间中两个样本之间的距离。其中，根据空间中度量的定义，对于空间中任意三个样本点$\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)},\boldsymbol{x}^{(k)}\in X$，必须满足以下三个条件。

（1）非负性：$d(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)})\geq 0$，其中，等号当且仅当$\boldsymbol{x}^{(i)}\equiv\boldsymbol{x}^{(j)}$时取得；

（2）对称性：$d(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)})=d(\boldsymbol{x}^{(j)},\boldsymbol{x}^{(i)})$；

（3）三角不等式：$d(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)})+d(\boldsymbol{x}^{(j)},\boldsymbol{x}^{(k)})\geq d(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(k)})$。

给定一个D维输入变量$X=\{\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\boldsymbol{x}^{(3)},...,\boldsymbol{x}^{(s)}\}$和相应的输出标签$Y=\{y^{(1)},y^{(2)},y^{(3)},...,y^{(s)}\}$，度量学习需要从这些训练数据中找到一个隐含的度量函数。在这个度量函数定义的度量空间中，有相似标签的样本可以被聚集在一起，而不相似的样本则会被分离开。在度量学习领域，马氏度量学习（MML）[26]由于简洁明了的结构而被广泛研究，其模型结构如下所示：

$$d^2(\boldsymbol{u}-\boldsymbol{v})=(\boldsymbol{u}-\boldsymbol{v})^{\mathrm{T}}\boldsymbol{M}(\boldsymbol{u}-\boldsymbol{v}) \tag{1}$$

式中，$\boldsymbol{M}$为正定矩阵；$\boldsymbol{u}$和$\boldsymbol{v}$表示两个不同的样本。MML的目标就是根据学习的期望找到最优的矩阵$\boldsymbol{M}$。

本研究希望利用输出标签中隐含的信息来指导度量学习的方向。基本原理是相似的输入样本导致相似的目标标签。从统计角度来看，输入和输出空间之间距离的一致性可以用皮尔逊相关系数来描述。因此，优化问题形成如下：

$$\underset{M}{\operatorname{argmin}}\ J(\boldsymbol{M})=\frac{-\sum_{i>j}d_{ij}^{(y)}\cdot\left(\boldsymbol{\delta}_{ij}^{(x)}\right)^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\delta}_{ij}^{(x)}}{\sqrt{\sum_{i>j}\left[\left(\boldsymbol{\delta}_{ij}^{(x)}\right)^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\delta}_{ij}^{(x)}\right]^2-\frac{2}{N(N-1)}\left[\sum_{i>j}\left(\boldsymbol{\delta}_{ij}^{(x)}\right)^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\delta}_{ij}^{(x)}\right]^2}} \tag{2}$$

s.t. $\boldsymbol{M}\geq 0$

式中，$\boldsymbol{M}$为正定矩阵；$d_{ij}^{(y)}$表示i样本和j样本在输出空间的度量的平方；$\boldsymbol{\delta}_{ij}^{(x)}$表示两个样本在输入空间中的偏差，也就是$\boldsymbol{\delta}_{ij}^{(x)}\equiv\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(j)}$；$N$表示样本的总个数。由于目标函数的分子、分母关于矩阵$\boldsymbol{M}$是齐次的，因此，可以将上述式（2）问题转换成式（3）的等价优化问题。

$$\underset{M}{\operatorname{argmin}}\ J'(\boldsymbol{M})=-\sum_{i>j}d_{ij}^{(y)}\cdot(\boldsymbol{\delta}_{ij}^{(x)})^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\delta}_{ij}^{(x)}$$

s.t. $\boldsymbol{M}\geq 0$ （3）

$$\sum_{i>j}\left[(\boldsymbol{\delta}_{ij}^{(x)})^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\delta}_{ij}^{(x)}\right]^2-\frac{2}{N(N-1)}\left[\sum_{i>j}(\boldsymbol{\delta}_{ij}^{(x)})^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\delta}_{ij}^{(x)}\right]^2=1$$

本研究证明了上述问题具有唯一的全局最优解，并且可以通过松弛约束来获得该解。约束松弛后的重构问题如式（4）所示：

$$\underset{M}{\operatorname{argmin}}\ J'(\boldsymbol{M})=-\sum_{i>j}d_{ij}^{(y)}\cdot(\boldsymbol{\delta}_{ij}^{(x)})^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\delta}_{ij}^{(x)}$$

s.t. $\boldsymbol{M}\geq 0$

$$\sum_{i>j}\left[(\boldsymbol{\delta}_{ij}^{(x)})^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\delta}_{ij}^{(x)}\right]^2-\frac{2}{N(N-1)}\left[\sum_{i>j}(\boldsymbol{\delta}_{ij}^{(x)})^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\delta}_{ij}^{(x)}\right]^2\leq 1 \tag{4}$$

记$g(\boldsymbol{M})=\sum_{i>j}\left[(\boldsymbol{\delta}_{ij}^{(x)})^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\delta}_{ij}^{(x)}\right]^2-\frac{2}{N(N-1)}\left[\sum_{i>j}(\boldsymbol{\delta}_{ij}^{(x)})^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\delta}_{ij}^{(x)}\right]^2$，那么$g(\boldsymbol{M})$对$\boldsymbol{M}$的一阶偏导数和二阶偏导数的结果分别如下所示：

$$\frac{\partial g}{\partial\mathrm{vec}(\boldsymbol{M})}=2\sum_{i>j}(\boldsymbol{\delta}_{ij}^{(x)})^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\delta}_{ij}^{(x)}\cdot\left[\boldsymbol{\delta}_{ij}^{(x)}\otimes\boldsymbol{\delta}_{ij}^{(x)}\right]-\frac{4}{N(N-1)}\left[\sum_{i>j}(\boldsymbol{\delta}_{ij}^{(x)})^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\delta}_{ij}^{(x)}\right]\left[\sum_{i>j}\boldsymbol{\delta}_{ij}^{(x)}\otimes\boldsymbol{\delta}_{ij}^{(x)}\right] \tag{5}$$

$$\frac{\partial^2 g}{\partial\mathrm{vec}(\boldsymbol{M})^2}=2\sum_{i>j}\left[\boldsymbol{\delta}_{ij}^{(x)}\otimes\boldsymbol{\delta}_{ij}^{(x)}\right]\cdot\left[\boldsymbol{\delta}_{ij}^{(x)}\otimes\boldsymbol{\delta}_{ij}^{(x)}\right]^{\mathrm{T}}-\frac{4}{N(N-1)}\left[\sum_{i>j}\boldsymbol{\delta}_{ij}^{(x)}\otimes\boldsymbol{\delta}_{ij}^{(x)}\right]\left[\sum_{i>j}\boldsymbol{\delta}_{ij}^{(x)}\otimes\boldsymbol{\delta}_{ij}^{(x)}\right]^{\mathrm{T}} \tag{6}$$

式中，$\otimes$表示克罗内克积（Kronecker product），vec($\boldsymbol{M}$)表示矩阵$\boldsymbol{M}$的列展开。显然，式（7）中的不等式表明函数$g(\boldsymbol{M})$是凸函数。

$$\boldsymbol{u}^{\mathrm{T}}\frac{\partial^2 g}{\mathrm{vec}(\boldsymbol{M})^2}\boldsymbol{u}=2\sum_{i>j}m_{ij}^2-\frac{4}{N(N-1)}\left[\sum_{i>j}m_{ij}\right]^2\geqslant 0 \tag{7}$$

式中，$m_{ij}\equiv\boldsymbol{u}^{\mathrm{T}}\cdot\left[\boldsymbol{\delta}_{ij}^{(x)}\otimes\boldsymbol{\delta}_{ij}^{(x)}\right]$。这意味着式（4）中的约束导致可行域为凸集。同时，目标函数$J(\boldsymbol{M})$对于$\boldsymbol{M}$的二阶偏导数计算如下：

$$\frac{\partial^2 J}{\mathrm{vec}(\boldsymbol{M})^2}=0 \tag{8}$$

综上分析可知，式（4）为凸优化问题，即具有唯一的全局最优解[36]。记该最优解为$\boldsymbol{M}^*$，那么得出结论：$g(\boldsymbol{M}^*)=1$。否则，如果$0<g(\boldsymbol{M}^*)<1$，令$\boldsymbol{M}'=\boldsymbol{M}^*/g(\boldsymbol{M}^*)$，并将$\boldsymbol{M}^*$代入式（4）中，不难验证$\boldsymbol{M}'$在可行域内。并且，$J(\boldsymbol{M}')=\frac{J(\boldsymbol{M}^*)}{g(\boldsymbol{M}^*)}<J(\boldsymbol{M}^*)$，与$\boldsymbol{M}^*$为最优解矛盾。因此，式（3）所得的唯一最优解满足等式约束，可通过求解式（4）中的凸优化问题获得。

3. 局部二次嵌入学习

大多数k-NNR都以样本在输入空间的k-NN样本输出的加权均值作为预测结果。但是，由于数据本身存在噪声，在近邻数较少时预测结果不稳定；而在近邻数较多时预测的结果往往较保守。给定待预测的样本$\boldsymbol{x}$，定义其k-NN样本为$\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\ldots,\boldsymbol{x}^{(K)}$，以及相应的输出为$y^{(1)},y^{(2)},\ldots,y^{(K)}$，则采用非常直观的加权平均的方式$\hat{y}=\sum_{i=1}^{K}w_i y^{(i)}\ (w_i\geq 0,\sum_{i=1}^{K}w_i=1)$，得到的结果必然满足$\min_i\{y^{(i)}\}\leq\hat{y}\leq\max_i\{y^{(i)}\}$。显然，这是一个非常保守的结果。为了解决这个问题，本文建立了关于两个空间中的一个局部线性映射增量模型，以及一个独立的模型，以区分不同近邻预测的可靠性。也就是说，基于不同相邻预测为预测结果分配不同的权重。

LQEL算法的框架如图1所示。为了获得与样本$\boldsymbol{x}$相对应的输出标签，首先使用第2节中的度量学习结论（图1左侧的椭圆）确定k-NN。定义一个函数$\mathcal{F}:\delta\boldsymbol{x}\rightarrow\delta y$，那么对于每一个近邻样本$\boldsymbol{x}_j\left(j=1,2,3,\ldots,K\right)$，算法都可以根据其相应的输出标签及$\boldsymbol{x}_j$与样本$\boldsymbol{x}$之间的差分得到一个预测输出：$\hat{y}_j=y_j+\mathcal{F}(\boldsymbol{x}-\boldsymbol{x}_j)\ (j=1,2,3,\ldots,K)$。最后，将上述预测结果与适合的权重线性组合以获得最终输出：

$$\hat{y}=\sum_{j=1}^{K}w_j\left[y_j+\mathcal{F}\left(\boldsymbol{x}-\boldsymbol{x}_j\right)\right]\quad(j=1,2,3,\ldots,K)\qquad(9)$$

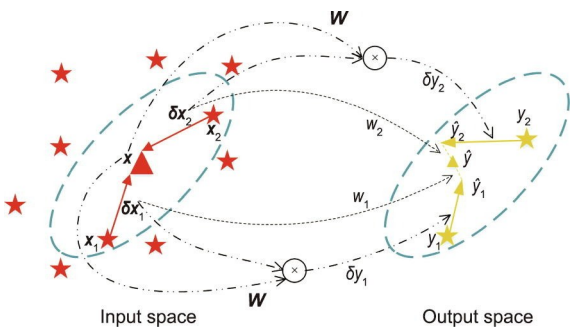

图1. 局部二次嵌入学习的框架图。

定义从输入到输出的真实映射函数为$g_0\in C^2:X\rightarrow R$，并且定义该函数在$\boldsymbol{x}_0$点的$\delta$邻域内做二阶泰勒展开得到的函数为$\tilde{g}_0(\boldsymbol{x}|\boldsymbol{x}_0)=0.5\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}_{\boldsymbol{x}_0}\boldsymbol{x}+\boldsymbol{B}_{\boldsymbol{x}_0}\boldsymbol{x}+C_{\boldsymbol{x}_0}$，其中：

$$\boldsymbol{A}_{\boldsymbol{x}_0}=\nabla^2 g_0(\boldsymbol{x}_0)$$

$$\boldsymbol{B}_{\boldsymbol{x}_0}=\nabla g_0(\boldsymbol{x}_0)^{\mathrm{T}}-(\boldsymbol{x}_0)^{\mathrm{T}}\nabla^2 g_0(\boldsymbol{x}_0)$$

$$C_{\boldsymbol{x}_0}=g_0(\boldsymbol{x}_0)-\nabla g_0(\boldsymbol{x}_0)^{\mathrm{T}}\boldsymbol{x}_0+0.5(\boldsymbol{x}_0)^{\mathrm{T}}\nabla^2 g_0(\boldsymbol{x}_0)\boldsymbol{x}_0$$

那么，对于$\forall\boldsymbol{x}_1,\boldsymbol{x}_2\in U_\delta(\boldsymbol{x}_0)$，其相应输出空间上的差分计算结果如下所示：

$$\begin{aligned}y_1-y_2&\approx\tilde{g}_0\left(\boldsymbol{x}_1|\boldsymbol{x}_0\right)-\tilde{g}_0\left(\boldsymbol{x}_2|\boldsymbol{x}_0\right)\\&=\left(0.5(\boldsymbol{x}_1)^{\mathrm{T}}\boldsymbol{A}_{\boldsymbol{x}_0}\boldsymbol{x}_1+\boldsymbol{B}_{\boldsymbol{x}_0}\boldsymbol{x}_1+C_{\boldsymbol{x}_0}\right)-\\&\quad\left(0.5(\boldsymbol{x}_2)^{\mathrm{T}}\boldsymbol{A}_{\boldsymbol{x}_0}\boldsymbol{x}_2+\boldsymbol{B}_{\boldsymbol{x}_0}\boldsymbol{x}_2+C_{\boldsymbol{x}_0}\right)\\&=0.5(\boldsymbol{x}_1+\boldsymbol{x}_2)^{\mathrm{T}}\boldsymbol{A}_{\boldsymbol{x}_0}(\boldsymbol{x}_1-\boldsymbol{x}_2)+\boldsymbol{B}_{\boldsymbol{x}_0}(\boldsymbol{x}_1-\boldsymbol{x}_2)\\&\approx\left((\boldsymbol{x}_0)^{\mathrm{T}}\boldsymbol{A}_{\boldsymbol{x}_0}+\boldsymbol{B}_{\boldsymbol{x}_0}\right)(\boldsymbol{x}_1-\boldsymbol{x}_2)\\&\equiv\boldsymbol{W}\left(\boldsymbol{x}_1-\boldsymbol{x}_2\right)\end{aligned}\qquad(10)$$

式中，$U_\delta(\boldsymbol{x}_0)$表示在第2节中定义的度量空间下的$\delta$邻域，$\boldsymbol{W}\equiv(\boldsymbol{x}_0)^{\mathrm{T}}\boldsymbol{A}_{\boldsymbol{x}_0}+\boldsymbol{B}_{\boldsymbol{x}_0}$是线性映射函数的权重系数矩阵。

式（10）的结果表明，可以设计线性模型来预测$\boldsymbol{x}_0$的δ邻域。在不同参考点上展开的矩阵$\boldsymbol{W}$可以由独立的神经网络估计。例如，使用神经网络$\mathcal{N}:X\rightarrow X$将矩阵近似为$\mathcal{N}(\boldsymbol{x}_0)=\boldsymbol{W}$。考虑参数矩阵$\nabla^2 g(\boldsymbol{x}_0)$和$\nabla g(\boldsymbol{x}_0)$在大多数实际情况下往往比$g(\boldsymbol{x}_0)$更稳定，这里需要的神经网络应该比直接估计输出标签的神经网络简单得多。特别地，当g_0是二次函数时，矩阵$\boldsymbol{A}_{\boldsymbol{x}_0}$和$\boldsymbol{B}_{\boldsymbol{x}_0}$不随参考点变化。在这种情况下，简单的线性神经网络可以很好地工作。总的来说，这些过程可以有效地降低模型的复杂性并提高泛化能力。

该策略使每个样本可以得到不同近邻的k个预测结果，然而不同的近邻样本给出的预测结果的可靠性是不一样的。一个直观的想法是，非近邻点给出的预测往往具有更高的不确定性，意味着应该为每个预测结果分配不同的权重。而由测量噪声引起的预测不确定性可以通过平均法来抑制。受此启发，本文设计一个估计器，根据实例的相对位置生成不同的权重，从而最小化均方误差（MSE）的期望。

记叠加在近邻样本标签$y^{(i)}$上的噪声为$v^{(i)}$，并假设该噪声服从正态分布$v^{(i)}\sim N(0,\sigma^2)$，那么该近邻样本给出的估计结果的误差如下所示：

$$\begin{aligned}\hat{y}_i-y_0&=y^{(i)}+((\boldsymbol{x}_0)^{\mathrm{T}}\boldsymbol{A}_{\boldsymbol{x}_0}+\boldsymbol{B}_{\boldsymbol{x}_0})(\boldsymbol{x}_0-\boldsymbol{x}^{(i)})-g(\boldsymbol{x}_0)\\&\approx 0.5(\boldsymbol{x}^{(i)})^{\mathrm{T}}\boldsymbol{A}_{\boldsymbol{x}_0}\boldsymbol{x}^{(i)}+\boldsymbol{B}_{\boldsymbol{x}_0}\boldsymbol{x}^{(i)}+C_{\boldsymbol{x}_0}+\\&\quad v^{(i)}+((\boldsymbol{x}_0)^{\mathrm{T}}\boldsymbol{A}_{\boldsymbol{x}_0}+\boldsymbol{B}_{\boldsymbol{x}_0})(\boldsymbol{x}_0-\boldsymbol{x}^{(i)})-g(\boldsymbol{x}_0)\\&=0.5(\boldsymbol{x}^{(i)}-\boldsymbol{x}_0)^{\mathrm{T}}\boldsymbol{A}_{\boldsymbol{x}_0}(\boldsymbol{x}^{(i)}-\boldsymbol{x}_0)+v^{(i)}\\&\equiv e^{(i)}+v^{(i)}\end{aligned}\qquad(11)$$

式中，$\hat{y}_i$表示第i个近邻样本给出的估计结果；$e^{(i)}$表示由于局部近似而引入的误差项；$v^{(i)}$表示不确定性。那么，问题就转换成为获取一组最优的权重，使得目标函数$H(\boldsymbol{w})$最小：

$$\begin{aligned}\min H(\boldsymbol{w})&=E\left[\left(\sum_{i=1}^{k}w_i\hat{y}_i-y_0\right)^2\right]\\&=E\left\{\left[\sum_{i=1}^{k}w_i(e^{(i)}+v^{(i)})\right]^2\right\}\\&=\left(\sum_{i=1}^{k}w_i e^{(i)}\right)^2+\sigma^2\sum_{i=1}^{k}{w_i}^2\end{aligned}$$

$$\text{s.t.}\quad \sum_{i=1}^{k} w_i = 1$$

$$w_i \geq 0 \quad (i = 1, 2, 3, ..., K) \tag{12}$$

上述优化问题可以通过拉格朗日乘子法和KKT（Karush-Kuhn-Tucker）条件求解得到。式（12）所涉及的变量有$e^{(i)}$和σ^2，其中σ^2与样本点本身无关，$\boldsymbol{x}_0$的最优权重结果由$\boldsymbol{\delta}_{\boldsymbol{x}_0}^{(i)} = \boldsymbol{x}^{(i)} - \boldsymbol{x}_0$确定。这里引入了一个神经网络来生成不同的权重，而不是直接求解优化问题，其输入是样本与其近邻之间的差分，即$\boldsymbol{\delta}_{\boldsymbol{x}_0}^{(i)}(i = 1, 2, 3, \ldots, K)$。

综上所述，本节提出一个如图2所示的预测框架。对于给定样本$\boldsymbol{x}^{(i)}$，首先根据第2节得到的度量学习结果来确定k-NN。其中，样本为$\boldsymbol{x}_j^{(i)}(j = 1, 2, 3, ..., k)$，相应的目标标签为$y_j{}^{(i)}(j = 1, 2, 3, ..., k)$。

另外，样本$\boldsymbol{x}^{(i)}$的系数矩阵$\boldsymbol{W} \in R^D$用神经网络I来计算，进而用于估计$\hat{\delta}_{y^{(i)},j} = \boldsymbol{W}^{\mathrm{T}} \boldsymbol{\delta}_{\boldsymbol{x}^{(i)},j} = \boldsymbol{W}^{\mathrm{T}} \left(\boldsymbol{x}^{(i)} - \boldsymbol{x}_j^{(i)} \right)$；$\boldsymbol{x}^{(i)}$的第$j$个预测结果为$\hat{y}_j{}^{(i)} = y_j^{(i)} + \hat{\delta}_{y^{(i)},j}$。最后，每个估计的权重$w_j^{(i)}$由神经网络II提供，并以$\boldsymbol{\delta}_{\boldsymbol{x}^{(i)},j}$作为输入。最终预测值则通过$\hat{y}_j{}^{(i)}(j = 1, 2, 3, \ldots, K)$的线性组合来实现。

在本文中，采用MSE作为损失函数，两个神经网络的权重和偏差参数利用SGD算法进行优化。

4. 算法验证

为了评估所提出算法的效果，使用公开的标准回归数据集和两个实际的工业数据集进行验证。同时，为了更好地进行比较，本文简要介绍了一系列经典的方法。最后，通过图表展示相应的实验结果。

4.1. 测试数据集的简介

4.1.1. 基准数据集

数据集[37–39]的详细信息如表1所示。例如，第一行所示的红酒数据集包含1599个样本，每个样本包含12个特征属性和一个要预测的目标标签。实验的目的是建立一个数学模型，通过颜色、成分等评价红酒的质量。在这种情况下，红酒的质量从高到低等分为9个等级，数据集中只包括三级和八级之间的样品。

表1 UCI数据集的详细描述

Name	Prediction label	Attribute number	Sample number
Wine quality	Wine quality	12	1 599
Forest fire	Burned area	12	517
CASP	RMSD-size of residue	10	45 730
Air quality	CO concentration	14	9 356
Air quality	NO_x concentration	14	9 356
Air quality	NO_2 concentration	14	9 356
Boston house price	House price	13	506

CASP: critical assessment of protein structure prediction; RMSD: root mean square deviation.

4.1.2. 生料细度数据集

第一个实际工业应用的目的是在线预测生料制备过程中的粉末细度。该工艺流程的细节如图3所示。在生料制备过程中，由三种或四种矿物组成的原料被输送到磨盘的中心，物料受到离心力的作用而沿外径扩散。物料在离开磨盘之前，通过研磨辊和磨盘之间的挤压被破碎成小颗粒。当高速热风从底部进入磨机时，较细的颗粒被吹入磨腔，而较大的颗粒落在底部，由斗式提升机运回磨机入口。由引风机驱动的高速气流将这些较细的颗粒送入高效的选粉机，不合格的颗粒沿着锥形筒落回磨盘上并被重新研磨。通过旋风筒和电收尘器回收的粉尘最终被输送至均化库进行储存。

该过程最重要的指标是产品的细度，该指标将影响后续煅烧过程的能耗甚至产品质量。由于实验室化验分析能力有限，每两小时采集一次样品并进行分析，无法进行有效的实时控制，生料细度波动较大。因此，本节的目的在于利用

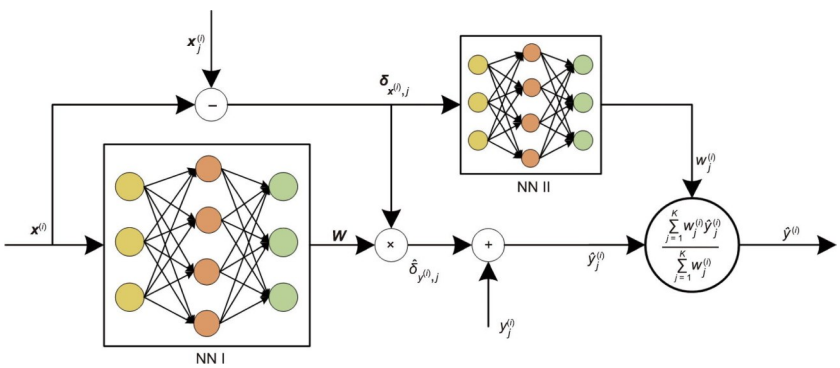

图2. 局部二次嵌入模型学习预测的结构框图。

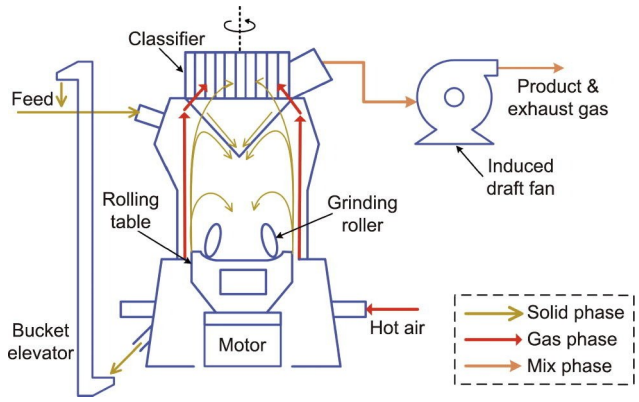

图3. 生料制备过程的工艺流程图。

可在线测量的相关过程变量实现对产品细度的在线估计。

所有可能影响或反映细度的变量都被视为辅助变量，包括引风机电流、选粉机电流、主电机电流、用于外循环的斗式提升机电流、输送成品的斗式提升机电流、压差、入口温度、出口温度、进料量等。通常，80 μm筛余量和200 μm筛余量被认为是生料细度的重要指标，且前者更灵敏。因此，数据集由14个辅助变量和一个输出标签构成，共959个样本（约4个月）。

4.1.3. 加氢裂化工艺数据集

典型加氢裂化工艺的简化流程图如图4所示。将原料与外部供应的氢气混合，并将氢气加热至规定温度，进入两个级联反应器。第一反应器装有加氢处理催化剂，以除去大部分硫和氮以及一些重金属化合物。第二反应器中装有加氢裂化催化剂以完成裂化反应。在这些反应器中，直接加入低温氢以吸收放热反应释放的热量，从而保持稳定的温度。反应产物通过高压分离器再循环未反应的氢气，然后通过低压分离器分离一些轻气体。最后，通过分馏塔实现不同组分的分离。收集了6种产品：轻馏分（LE）、轻石脑油（LN）、重石脑油（HN）、煤油（KE）、柴油（DI）和底油（BO）。

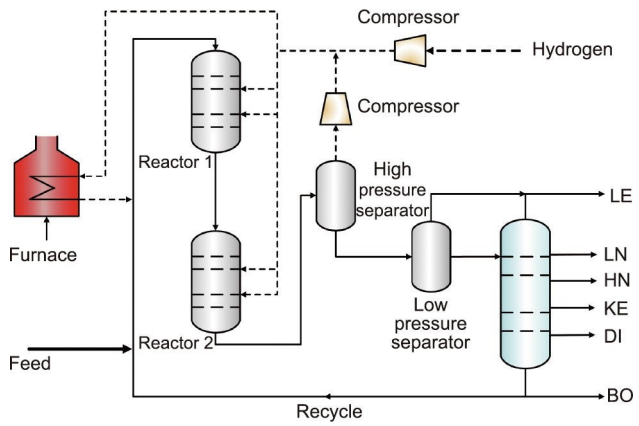

图4. 加氢裂化工艺流程图。

由于产品价格的波动和市场供求的变化，必须对不同产品的收益率进行相应调整，以实现总利润的最大化。因此，及时准确预测每种产品的产量以指导操作优化至关重要。本文以柴油产量为例建立了预测模型。在这个问题中，采样周期为4 h，数据集总共覆盖15个月。最后收集具有55个相关输入变量的2052个样品，包括进料质量流速、产出氢气的体积流量等。

4.2. 用户指定参数

本文的对比实验包括7个经典回归算法：

（1）基于MML的k-NNR首次采用参考文献[27]中提出的MML方法。该模型首先定义了基于三元组的约束，然后将优化问题转化为凸二次规划问题。该算法需要确定近邻的数量K^k。

（2）支持向量回归模型通过引入正则化参数C实现了结构风险和经验风险之间的平衡，并通过引入核函数方法实现了数据的非线性映射。本文使用线性、多项式和高斯核函数进行了性能对比测试。结果表明高斯核函数更适合这类回归问题。因此，正则化参数C和核参数γ需要被优化。

（3）随机森林模型是最成熟的bagging算法之一，该模型通过集成多个弱分类器的预测结果来提升算法的稳定性，另外，随机的行列采样也有利于进一步提升算法的泛化性能。在该算法中需要通过交叉验证的方式给出最大树深度$d_{\max}^{\mathrm{rf}}$和弱估计器的数量$N_{\mathrm{e}}^{\mathrm{rf}}$。

（4）Xgboost算法通过不断降低拟合残差得到一系列的弱估计器。其中，拟合过程中优化的方向采用残差二阶泰勒展开的方式进行估计。模型在多个实践应用场景中都表现出稳定可靠的性能。LightGBM [40]是XGBoost的改进分支之一，采用面向叶子节点分裂方法，并应用直方图的方法进行预处理以加速计算。已经证明，LightGBM可以在确保预测性能的同时大大提高计算性能。因此，本文将该方法作为对比算法之一，其中需要通过交叉验证给出的最佳参数包括：最大叶子节点数$N_{\mathrm{l}}^{\mathrm{lgb}}$、学习速率$\mathrm{lr}^{\mathrm{lgb}}$和$l_2$正则化系数$l_2^{\mathrm{lgb}}$。

（5）神经网络模型是解决回归问题的有效学习工具。采用包括BN和dropout在内的策略，这些策略在各个领域都是最先进的[35]。具体而言，在训练过程中选择的批次大小为30，dropout的比例为0.3，通过五重交叉验证选择隐藏神经元的数量$N_{\mathrm{h}}^{\mathrm{n}}$。

（6）深度因式分解机（deep factorization machine, DeepFM）[41]在广告点击率（click-through rate, CTR）预测问题[42]和股市价格的预测问题[43]上都有非常成功的

应用。DeepFM旨在通过因式分解机（factorization machine, FM）和深度神经网络实现低阶和高阶特征之间的交互与融合。其中，FM得到的嵌入向量被用于深度神经网络初始嵌入状态。对于其中的深度神经网络模型，本文设置的隐层数量为2。因此在该算法中，待确定的参数包括嵌入的维度$d_{\mathrm{e}}^{\mathrm{df}}$和隐层神经元的个数$N_{\mathrm{h}}^{\mathrm{df}}$。

（7）基于DML的*k*-NNR算法旨在通过深度学习模型找到样本的关键特征，并以此找到样本的相似近邻。由于本文针对的是回归问题，无法像当前DML那样构建三元组[29–31]。因此，根据相似输入对应相似输出的基本原则，本文采用如式（13）所示损失函数进行深度度量模型的学习。

$$L=\sum_{i\neq j}\left(d_{i,j}^{y}-\left\|f(x^{(i)})-f(x^{(j)})\right\|_2\right)^2 \tag{13}$$

式中，$d_{i,j}^{y}=\left|y^{(i)}-y^{(j)}\right|$；$f(\cdot)$表示度量学习的嵌入算子。在此基础上，相似的样本对应于相似的输出标签，进而采用*k*-NNR算法就能实现可靠的预测。上述算法中待确定的参数包括嵌入的维度$d_{\mathrm{e}}^{\mathrm{dml}}$和近邻的数量$k^{\mathrm{dml}}$。

对于所提出的LQEL算法，待估计的参数包括嵌入的维度$d_{\mathrm{e}}^{\mathrm{lqel}}$、两个神经网络模型中隐层神经元的个数$N_{\mathrm{hs}}^{\mathrm{lqel}}$、$N_{\mathrm{hw}}^{\mathrm{lqel}}$和近邻样本的数量$k^{\mathrm{lqel}}$。本文采用了五重交叉验证的方式给出上述参数的选择结果。实验中使用的指定参数如表2所示。

结果表明，算法的近邻数在不同数据集上差别较大。首先，取决于数据集的规模，较大的数据量意味着较大的样本密度。如在蛋白质结构预测的关键评估（critical assessment of protein structure prediction, CASP）数据集中，样本数足够多，这就意味着在样本附近有足够多的近邻数据可供参考，从而有效地提高模型的预测能力。然而对于工业中产品细度数据集，可用于建模的样本有限。此外很难使用辅助变量的值来表征状态。例如，立式辊磨机（VRM）中的吐渣量通常通过斗式提升机的电流来评估，但在定期维护（大约每两天一次）时，尤其是在添加润滑油后，会出现电流值漂移。因此，有必要更加关注当前的变化。在这些情况下，空间中近邻数据可能不如CASP数据集那样具有指导意义。因此，该模型选择少量相邻样本进行预测。由表2可知，所提出的具有简单前向神经网络的LQEL模型可以很好地解决回归问题。与前向神经网络模型相比，LQEL模型中的隐藏神经元更少（不超过4个），并且必须要估计的参数规模更小。这大大降低了模型的复杂性，从而提升了模型的泛化能力。

4.3. LQEL算法性能评估结果

为验证本文所提算法的有效性，在7个数据集的9个回归问题上进行了比较。每组实验重复30次，并记录每个模型在测试集上预测结果的MSE指标和平均绝对误差（mean absolute error, MAE），并且对这些评价指标进行统计分析，评估算法的准确性和鲁棒性。

表3是每个算法在不同数据集上的MSE指标和MAE指标的均值，其中每个数据集上的最佳结果通过加粗进行标记。结果表明，对于下面列出的9个验证测试，LQEL算法在大部分数据集上取得了最佳的性能。此外，LQEL算法的性能与表现较好的LightGBM和随机森林算法相当，与其他算法相比有明显的优势。

除此之外，为评估算法的稳定性，需要对所获取的30组损失指标的分布情况进行分析。图5和图6分别是各个算法在不同数据集上所得MSE指标和MAE指标的箱线图。从图中可以看到，LQEL算法在除红酒质量、CASP和细度之外的众多数据集上都表现出非常稳定的结果。尽管在这些数据集的表现中，性能波动比其他一些算法稍大，但总体MSE和MAE明显更低，也就是说，性能更稳定的算法往往以牺牲预测精度作为代价。即便是在神经网络和LQEL算法中都采用了dropout、批量学习和BN等策

表2 本章测试案例采用的参数

Datasets	MML-based	SVR		RF		LightGBM			NN,	DeepFM		DML-based *k*-NNR		LQEL			
	k-NNR, K^k	C	γ	$d_{\max}^{\mathrm{rf}}$	$N_{\mathrm{h}}^{\mathrm{n}}$	$N_{\mathrm{e}}^{\mathrm{rf}}$	N_{1}^{lgb}	$\mathrm{lr}^{\mathrm{lgb}}$	l_2^{lgb}	$d_{\mathrm{e}}^{\mathrm{df}}$	$N_{\mathrm{h}}^{\mathrm{df}}$	$d_{\mathrm{e}}^{\mathrm{dml}}$	k^{dml}	$d_{\mathrm{e}}^{\mathrm{lqel}}$	$N_{\mathrm{hs}}^{\mathrm{lqel}}$	$N_{\mathrm{hw}}^{\mathrm{lqel}}$	k^{lqel}
Wine quality	50	1	0.01	7	15	500	64	0.1	0	3	256	16	50	10	2	3	80
Forest fire	40	100	10 000	3	22	200	32	0.001	1	3	128	8	10	5	1	4	100
CASP	80	1 000	100	4	18	600	128	0.01	0.1	8	256	8	80	14	2	4	140
CO	60	100	0.01	3	26	200	32	0.1	0	12	128	8	20	5	1	5	20
NO_2	5	100	0.01	9	22	200	32	0.1	0.5	9	128	4	10	10	1	2	50
NO_x	10	100	1	9	26	200	64	0.1	0	12	128	4	10	10	2	3	20
House price	5	10 000	10	7	22	50	32	0.1	0.5	9	128	16	5	20	4	3	50
Fineness	10	100	0.01	9	28	100	32	0.1	0	12	256	16	50	10	1	4	20
Hydrocracking	50	100	0.0001	7	110	50	64	0.1	1	12	256	4	50	9	2	4	50

表3 不同算法预测结果的指标对比结果

Dataset	Error	MML-based *k*-NNR	SVR	RF	LightGBM	NN	DeepFM	DML-based *k*-NNR	LQEL
Wine quality	MSE	0.627	0.537	0.517	**0.486**	0.528	0.582	0.530	0.487
	MAE	0.792	0.733	0.700	0.697	0.769	0.768	0.720	**0.684**
Forest fire	MSE	3570	3680	3730	3620	3890	4220	4010	**3470**
	MAE	62.4	63.4	65.7	62.9	72.2	70.0	68.5	**61.8**
CASP	MSE	23.0	23.9	**20.9**	21.6	23.8	24.7	23.6	21.5
	MAE	4.80	4.89	4.67	4.65	4.89	4.87	4.80	**4.59**
CO	MSE	0.0690	0.0610	0.0617	0.0628	0.0838	0.0766	0.0635	**0.0564**
	MAE	0.263	0.247	0.261	0.251	0.281	0.301	0.269	**0.246**
NO_2	MSE	133.0	212.0	84.3	77.6	287.0	205.0	102.0	**63.3**
	MAE	11.50	14.60	8.81	8.81	16.20	14.20	11.10	**8.26**
NO_x	MSE	502	546	440	434	527	490	458	**403**
	MAE	22.4	23.4	21.1	**20.8**	22.7	22.7	23.0	20.8
House price	MSE	7.84	9.56	6.27	5.48	8.12	8.86	5.76	**4.71**
	MAE	2.80	3.09	2.34	**2.27**	2.98	2.86	2.51	2.29
Fineness	MSE	0.292	0.220	0.262	0.247	0.283	0.205	0.208	**0.202**
	MAE	0.541	0.448	0.523	0.497	0.502	0.447	0.456	**0.431**
Hydrocracking	MSE	0.0703	0.0843	0.0749	0.0381	0.0718	0.0393	0.0567	**0.0307**
	MAE	0.265	0.290	0.286	0.195	0.264	0.200	0.234	**0.181**

Bold values in each line indicate the best performance among different algorithms.

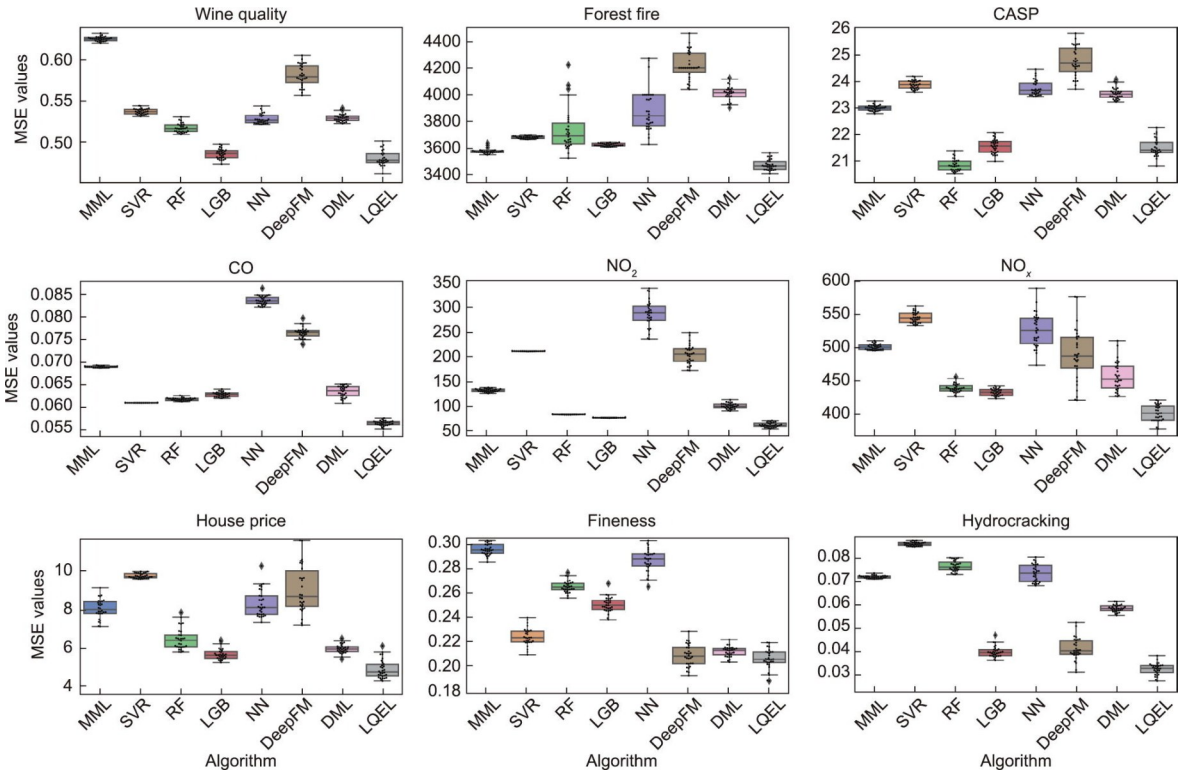

图5. 不同算法在测试集上MSE指标的箱线图。MML：MML-based *k*-NNR；LGB：LightGBM；DML：DML-based *k*-NNR。

略，后者仍优于前者。

图7和图8分别展示了两个工业数据集上不同算法的预测结果的散点图，其中横坐标是实际值，纵坐标是预测结果。确定性系数（R^2）标记在左上角，并表明LQEL算法在这两种软测量应用中优于其他算法。这可归因于两个方面：

（1）这些工业数据集中变量的绝对值不能很好地描述过程状态。本文提出的方法根据辅助变量的变化对最近邻进行校正，更加强化了数据的差异性，从而降低了上述问

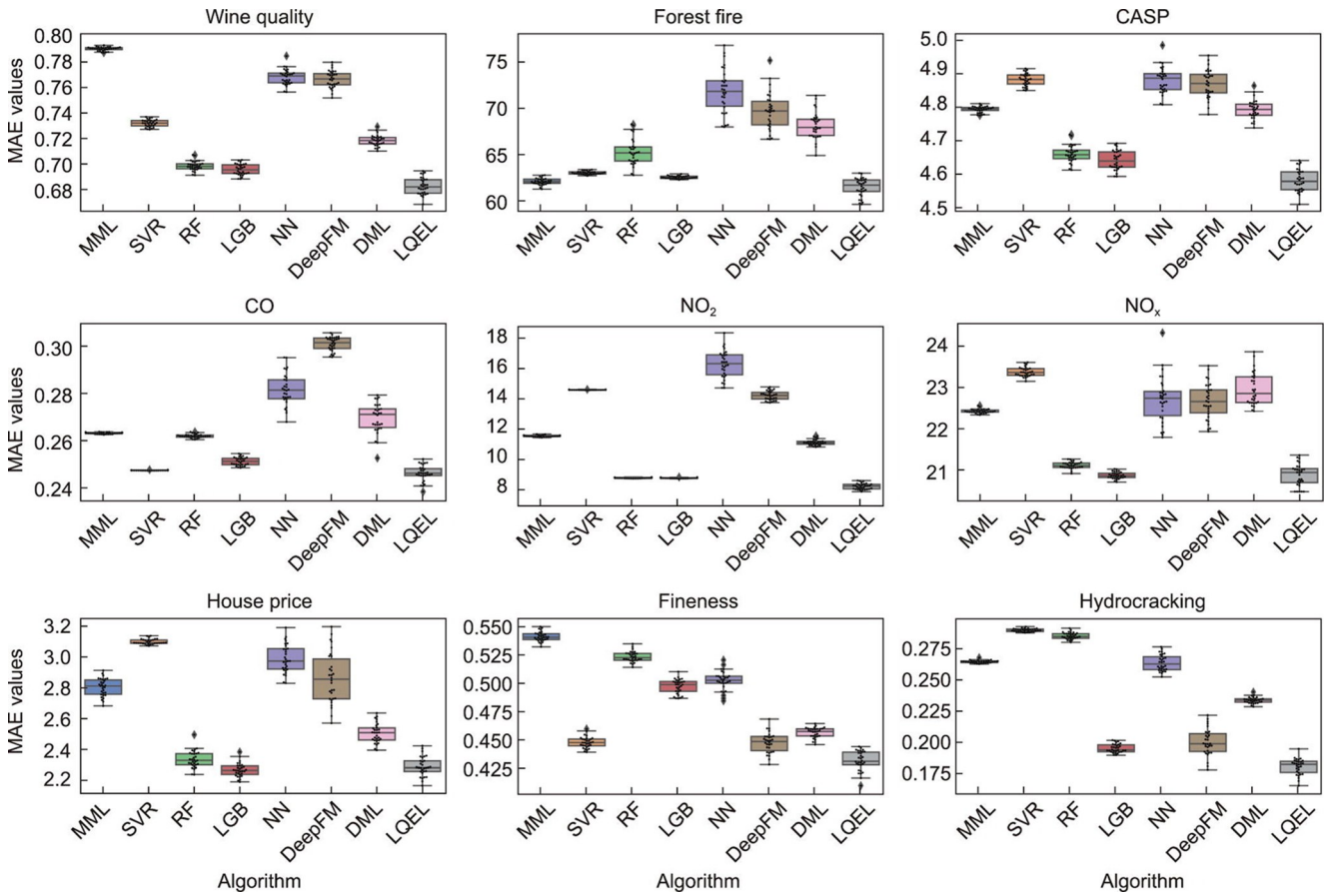

图6. 不同算法在测试集上MAE指标的箱线图。

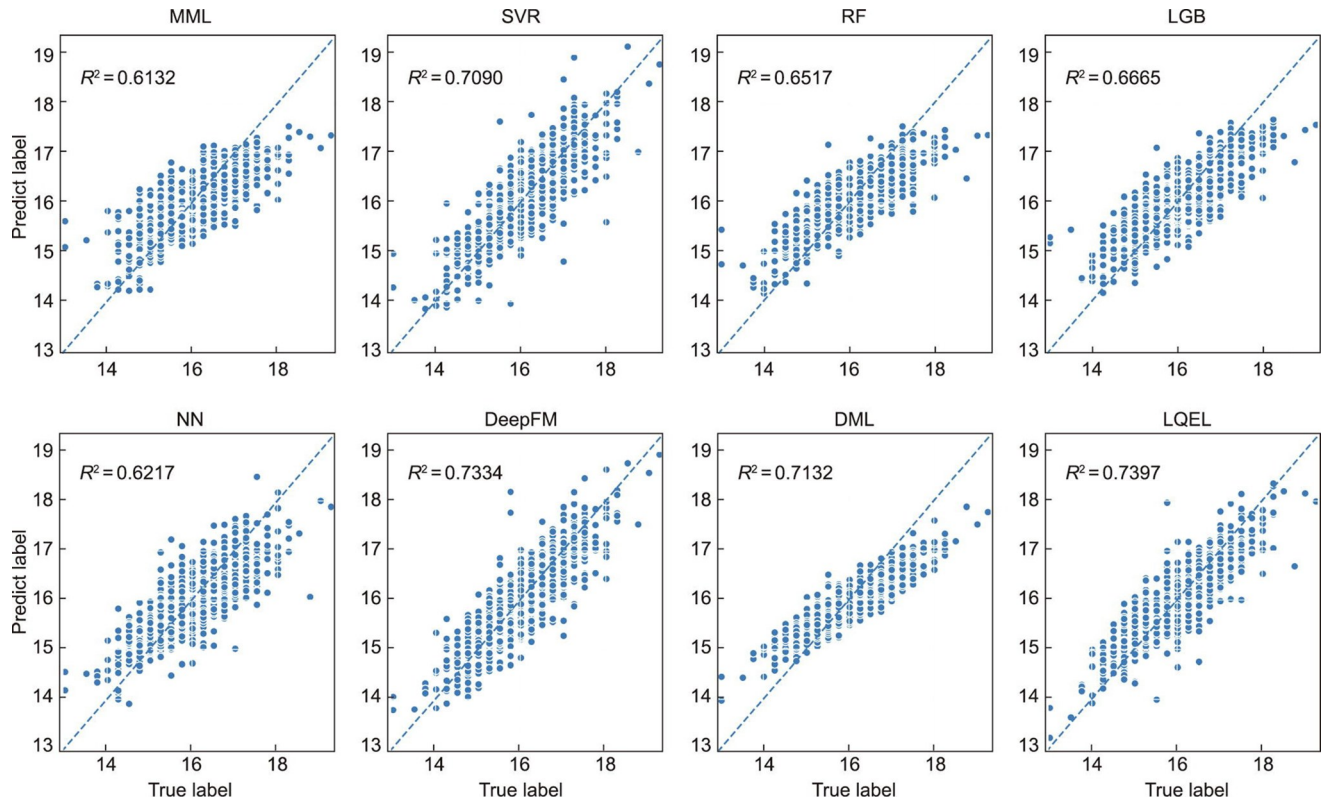

图7. 生料细度数据集上不同算法预测结果的散点图。

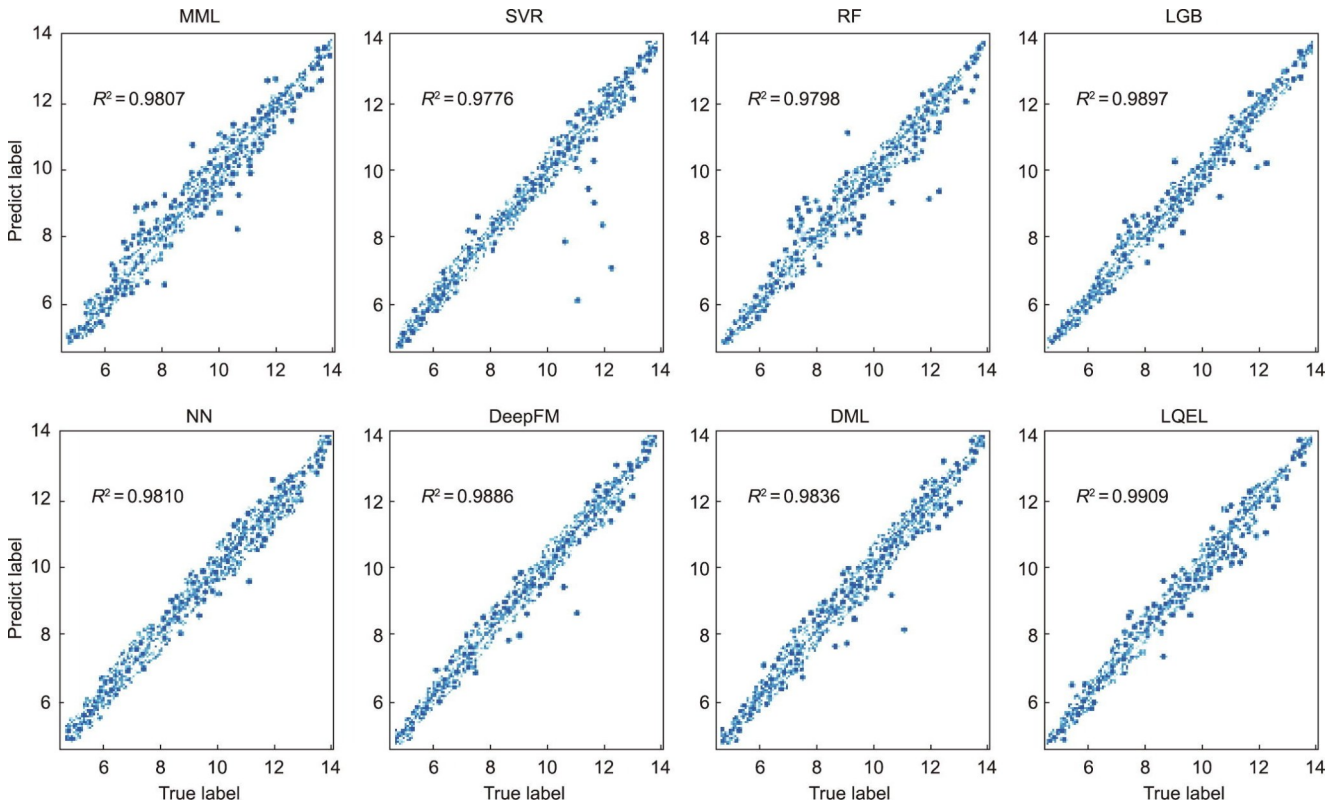

图8. 加氢裂化数据集上不同算法预测结果的散点图。

题的风险。

（2）该方法使用两个非常简单的神经网络来实现LQEL。其中一个神经网络的目标是找到局部二次函数的系数，另一个用于实现最近邻预测结果的权重分配。基于这些优点，可以有效地提高该算法的泛化能力。

5. 结论

本文提出了一种用于回归建模的LQEL算法。首先以最大化输入输出空间中距离的一致性来改进MML，通过松弛约束条件，证明了修正问题是一个凸优化问题，同时保持了与原问题相同的解。在此基础上，建立了局部二次嵌入模型，并为预测结果分配不同的权重，以最小化预测误差的期望值。在这个框架中，通过两个非常简单的神经网络来学习二次嵌入矩阵和相邻预测的权重分配。从而构建一个统一的端到端模型，防止独立的双层优化易于陷入局部最优的问题。本文所提出的LQEL模型有以下优点：

• 通过改进的度量学习实现了输入和输出空间中距离的全局一致性。

• 充分利用了输出标签中包含的信息，从而更好地确定特定样本的邻域。

• 基于局部二次嵌入假设提出了LQEL框架，并通过从全局或局部角度简化模型结构来设计两个特殊的神经网络，改进了模型的泛化能力。

• 实验结果表明，采用轻量级神经网络的LQEL可以实现更准确和更稳定的预测。

致谢

本研究得到了国家重点研发计划（2016YFB0303401）、国际（区域）合作与交流项目（61720106008）、国家杰出青年科学基金（61725301）和上海人工智能实验室的支持。

Compliance with ethics guidelines

Yaoyao Bao, Yuanming Zhu, and Feng Qian declare that they have no conflict of interest or financial conflicts to disclose.

References

[1] Bao YY, Zhu YM, Du WL, Zhong WM, Qian F. A distributed PCA-TSS based soft sensor for raw meal fineness in VRM system. Control Eng Pract 2019;90:38–49.

[2] Zhong WM, Jiang C, Peng X, Li Z, Qian F. Online quality prediction of

industrial terephthalic acid hydropurification process using modified regularized slow-feature analysis. Ind Eng Chem Res 2018;57(29):9604–14.
[3] Lu B, Chiang L. Semi-supervised online soft sensor maintenance experiences in the chemical industry. J Process Contr 2018;67:23–34.
[4] Li YG, Gui WH, Yang CH, Xie YF. Soft sensor and expert control for blending and digestion process in alumina metallurgical industry. J Process Contr 2013; 23(7):1012–21.
[5] Song YC, Zhou H, Wang PF, Yang MJ. Prediction of clathrate hydrate phase equilibria using gradient boosted regression trees and deep neural networks. J Chem Thermodyn 2019;135:86–96.
[6] Touzani S, Granderson J, Fernandes S. Gradient boosting machine for modeling the energy consumption of commercial buildings. Energ Build 2018; 158: 1533–43.
[7] Fernández-Delgado M, Sirsat MS, Cernadas E, Alawadi S, Barro S, Febrero-Bande M. An extensive experimental survey of regression methods. Neural Netw 2019;111:11–34.
[8] Yu J. Multiway Gaussian mixture model based adaptive kernel partial least squares regression method for soft sensor estimation and reliable quality prediction of nonlinear multiphase batch processes. Ind Eng Chem Res 2012; 51(40):13227–37.
[9] Yuan XF, Ge ZQ, Song ZH. Locally weighted kernel principal component regression model for soft sensing of nonlinear time-variant processes. Ind Eng Chem Res 2014;53(35):13736–49.
[10] Gou JP, Ma HX, Ou WH, Zeng SN, Rao YB, Yang HB. A generalized mean distance-based *k*-nearest neighbor classifier. Expert Syst Appl 2019;115:356–72.
[11] Juez-Gil M, Erdakov IN, Bustillo A, Pimenov DY. A regression-tree multilayer-perceptron hybrid strategy for the prediction of ore crushing-plate lifetimes. J Adv Res 2019;18:173–84.
[12] Suarez A, Lutsko JF. Globally optimal fuzzy decision trees for classification and regression. IEEE Trans Pattern Anal Mach Intell 1999;21(12):1297–311.
[13] Huang GB, Zhou HM, Ding XJ, Zhang R. Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern B Cybern 2012;42(2):513–29.
[14] Vilela LFS, Leme RC, Pinheiro CAM, Carpinteiro OAS. Forecasting financial series using clustering methods and support vector regression. Artif Intell Rev 2019;52(2):743–73..
[15] Paul A, Mukherjee DP. Reinforced quasi-random forest. Pattern Recognit 2019; 94:13–24.
[16] Rodriguez-Galiano VF, Ghimire B, Rogan J, Chica-Olmo M, Rigol-Sanchez JP. An assessment of the effectiveness of a random forest classifier for land-cover classification. ISPRS J Photogramm Remote Sens 2012;67:93–104.
[17] Freund Y. Boosting a weak learning algorithm by majority. Inf Comput 1995; 121(2):256–85.
[18] Chen TQ, Guestrin C. XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2016 Aug 13−17; San Francisco, CA, USA; 2016. p. 785–94.
[19] Dreiseitl S, Ohno-Machado L. Logistic regression and artificial neural network classification models: a methodology review. J Biomed Inform 2002;35(5−6): 352–9.
[20] Zhou ZH, Wu JX, Tang W. Ensembling neural networks: many could be better than all. Artif Intell 2002;137(1–2):239–63.
[21] Liu WB, Wang ZD, Liu XH, Zeng NY, Liu YR, Alsaadi FE. A survey of deep neural network architectures and their applications. Neurocomputing 2017;234: 11–26.
[22] Martos G, Muñoz A, González J. On the generalization of the Mahalanobis distance. In: Ruiz-Shulcloper J, Sanniti di Baja G, editors. Progress in pattern recognition, image analysis, computer vision, and applications. Berlin: Springer; 2013. p. 125–32.
[23] Atkeson CG, Moore AW, Schaal S. Locally weighted learning. Artif Intell Rev 1997;11:11–73.
[24] Zhang JJ. Identification of moving loads using a local linear embedding algorithm. J Vib Control 2019;25(11):1780–90.
[25] Loia V, Tomasiello S, Vaccaro A, Gao JW. Using local learning with fuzzy transform: application to short term forecasting problems. Fuzzy Optim Decis Making 2020;19(1):13–32.
[26] Weinberger KQ, Saul LK. Distance metric learning for large margin nearest neighbor classification. J Mach Learn Res 2009;10:207–44.
[27] Nguyen B, Morell C, De Baets B. Large-scale distance metric learning for *k*-nearest neighbors regression. Neurocomputing 2016;214:805–14.
[28] Xing EP, Ng AY, Jordan MI, Russell S. Distance metric learning, with application to clustering with side-information. In: Becker S, Thrun S, Obermayer K, editors. Advances in neural information processing systems 15: proceedings of the 2002 conference. Cambridge: A Bradford Book; 2003. p. 521–8.
[29] Duan YQ, Lu JW, Zheng WZ, Zhou J. Deep adversarial metric learning. IEEE Trans Image Process 2020;29:2037–51.
[30] Song HO, Xiang Y, Jegelka S, Savarese S. Deep metric learning via lifted structured feature embedding. In: Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition; 2016 Jun 26−Jul 1; Las Vegas, NV, USA; 2016. p. 4004–12.
[31] Cui Y, Zhou F, Lin YQ, Belongie S. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. In: Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition; 2016 Jun 26–Jul 1; Las Vegas, NV, USA; 2016. p. 1153–62.
[32] ALzubi JA, Bharathikannan B, Tanwar S, Manikandan R, Khanna A, Thaventhiran C. Boosted neural network ensemble classification for lung cancer disease diagnosis. Appl Soft Comput 2019;80:579–91.
[33] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. 2015. arXiv:1502.03167.
[34] Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors. 2012. arXiv:1207.0580.
[35] Kay S. Can detectability be improved by adding noise? IEEE Signal Process Lett 2000;7(1):8–10.
[36] Boyd V, Vandenberghe L, Faybusovich L. Convex optimization. IEEE Trans Automat Contr 2006;51(11):1859.
[37] Cortez P, Morais A. A data mining approach to predict forest fires using meteorological data. In: Neves JM, Santos MF, Machado JM, editors. New trends in artificial intelligence: proceedings of the 13th Portuguese Conference on Artificial Intelligence; 2007 Dec 3–7; Guimarães, Portugal; 2007. p. 512–23. French.
[38] Cortez P, Cerdeira A, Almeida F, Matos T, Reis J. Modeling wine preferences by data mining from physicochemical properties. Decis Support Syst 2009; 47(4):547–53.
[39] De Vito S, Massera E, Piga M, Martinotto L, Di Francia G. On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. Sens Actuators B Chem 2008;129(2):750–7.
[40] Ke GL, Meng Q, Finley T, Wang TF, Chen W, Ma WD, et al. LightGBM: a highly efficient gradient boosting decision tree. In: Guyon I, Von Luxburg U, Bengio S, Wallach H, Fergus R, Vishwanathan S, et al., editors. Proceedings of the 31st Annual Conference on Neural Information Processing Systems; 2017 Dec 4–9; Long Beach, CA, USA; 2017.
[41] Guo HF, Tang RM, Ye YM, Li ZG, He XQ. DeepFM: a factorization-machine based neural network for CTR prediction. In: Sierra C, editor. Proceedings of the 26th International Joint Conference on Artificial Intelligence; 2017 Aug 19–25; Melbourne, VIC, Australia; 2017. p. 1725–31.
[42] Zhang L, Shen WC, Huang JH, Li SJ, Pan G. Field-aware neural factorization machine for click-through rate prediction. IEEE Access 2019;7:75032–40.
[43] Huang JY, Zhang X, Fang BX. CoStock: a DeepFM model for stock market prediction with attentional embeddings. In: Proceedings of 2019 IEEE International Conference on Big Data; 2019 Dec 9−12; AngelesLos, CA, USA. New York City: IEEE; 2019. p. 5522–31.