文章编号：0253-374X(2021)01-0153-10

DOI：10.11908/j.issn.0253-374x.20346

基于余切相似度和BP神经网络的相似度快速计算

乔　非，关柳恩，王巧玲

（同济大学 电子与信息工程学院，上海 201804）

摘要：相似性度量在大数据相关应用中具有重要的意义，然而传统余弦相似度遍历计算方法的准确性和时效性较差，具有较大局限性，无法为海量高维数据的质量评估提供有效依据。针对上述问题，利用余切三角函数和数据维度差值构造2种余切相似度公式，提高相似度计算的准确性；借助后向传播(BP)神经网络建立一个能够逼近数据集相似度映射关系的网络模型，降低相似度计算的时间复杂度。实验表明，改进的相似度快速计算方法具有良好的准确性和时效性，而且应用在大规模数据集时的性能提升更显著。

关键词：相似度计算；神经网络；大数据分析；数据质量评估

中图分类号：TP311.1　　**文献标志码**：A

A Fast Similarity Calculation Method Based on Cotangent Similarity and BP Neural Network

QIAO Fei, *GUAN Liuen*, *WANGE Qiaoling*

(College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China)

Abstract: Similarity measurement is of great significance in big data related applications. However, the traditional cosine similarity traversal calculation method has a poor accuracy and timeliness, which cannot provide an effective basis for the quality assessment of massive high-dimensional data. To improve the accuracy of similarity calculation, two types of cotangent similarity formulas with cotangent trigonometric function and data dimensional differences was constructed. Besides, a back-propagation (BP) neural network model approximating the similarity mapping relationship of datasets was established to reduce the time complexity. The experimental results demonstrate that the improved fast similarity calculation method has a good accuracy and timeliness. Moreover, it has a more significant performance improvement when applied to large-scale datasets.

Key words: similarity calculation; neural network; big data analysis; data quality assessment

随着云计算等信息技术的发展，大数据日益渗透于金融、医疗、工业等各个行业领域之中，成为重要的生产因素。因此，对海量数据的挖掘和应用具有十分重要的现实意义，然而在实际生产过程中，大数据往往伴随着数据质量问题。相似度度量作为数据质量评估的重要方面，能够挖掘数据集中各数据间的相似程度，为数据分析提供准确和有效的依据。传统相似性度量方法分为基于距离度量方法和基于相似系数度量方法，基于距离度量方法主要有欧氏距离、曼哈顿距离、切比雪夫距离和马氏距离等，而基于相似系数度量方法主要有余弦相似度、皮尔森相关系数、杰卡德相似系数等。其中，欧氏距离和余弦相似度的应用尤为普遍。相比于欧氏距离利用数据点的距离作为度量的依据，余弦相似度更关注向量之间的夹角，分析数据在方向上的差异。作为经典相似性度量方法，余弦相似度常用于文本处理[1]、特征选择[2]、视觉任务[3]、实例检索[4]等领域。只关注方向的特性使得余弦相似度对噪声的敏感度较低。文献[3]提出一种局部加权余弦相似度来衡量目标模板和候选模板的相似度，有效抑制脉冲噪声导致的负面影响。文献[5]在直流线路短路故障中通过余弦相似度来检测故障端子极性现象，无需考虑幅值问题，从而有良好的抗噪性能。

尽管余弦相似度在多个应用中表现出良好的效果，但由于对绝对数值不敏感，余弦相似度无法识别方向相同但模长相异的数据向量的差异，在需要度量模长的场景中准确性较差。文献[6]指出余弦相似度只关注方向差异的特点会给某些识别任务带来

收稿日期：2020-08-27
基金项目：国家自然科学基金(71690230/71690234,61973237,61873191)
第一作者：乔　非(1967—)，女，教授，博士生导师，工学博士，主要研究方向为智能生产系统.
E-mail：fqiao@tongji.edu.cn

论文拓展介绍

比较大的影响,当数据分布比较密集时,向量夹角往往趋向于零,相似度普遍较高的情况下分类器无法区分实际上不相似的模式。针对上述存在的问题,文献[7]提出一种改进余弦相似度,通过引入数据点距离的L_p范数以及正则项,使得相似度函数能兼顾数据点距离以及向量夹角两方面。文献[8]则利用数据序列的模长比值来构造相似系数进行修正。还有文献[9]使用调整余弦相似度进行度量,通过减去数据均值化原点矩为中心矩,消除各个维度的量纲差异。当前,模糊集、中智集领域中关于相似性度量的研究工作比较深入,在余弦函数和余切函数的基础上也提出了多种相似度计算公式[10-12]。但上述研究基本没有对所提方法如何解决余弦相似度的不足进行清晰的文字阐述和相关的数学推算。

除此之外,以往相似度计算需要逐一遍历数据来计算数据集之间的相似度,适用于小规模数据集,但其运算时间会随着数据集规模扩增呈指数型增长,不适合应用于大规模数据集。为了摆脱高计算成本的困扰,文献[13]认为实现高速计算的方法之一是预先将计算所需的一些统计信息存储在索引中。文献[14]通过预先选择的代表性查询图像和相似度表来评估图像的相似程度。上述研究中,减少计算时间的本质是通过预计算来减少后继的冗余计算。值得注意的是,目前相似性度量的研究中关于快速计算这一方面的文献还比较欠缺。

综上所述,本文将针对余弦相似度准确性较低这一不足以及遍历计算方法时效性差的局限性展开相关的探索和研究。提出2种余切相似度并分析其在相似性度量中的优势,阐述基于BP神经网络的相似度快速计算方法和流程,针对改进方法的实验结果进行准确性和时效性的讨论分析,并进行总结与展望。

1 余切相似度计算公式

首先通过研究余弦相似度的计算公式揭示其不足,然后提出2种余切相似度计算公式,通过数学推算和实例说明分析余切相似度在衡量数据相似度方面的优势。

1.1 余弦相似度

余弦相似度的原理是通过计算数据向量的夹角余弦来表征两者的相似程度。假设有3个数据$\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$,其中$\boldsymbol{A}=(a_1,a_2,...,a_m)$,$\boldsymbol{B}=(b_1,b_2,...,b_m)$,$\boldsymbol{C}=(c_1,c_2,...,c_m)$,$m(m\in\mathbf{N}^*)$表示数据维数。为了方便表述,令$\mathcal{M}=\{i\in\mathbf{N}^*|1\leqslant i\leqslant m,m\in\mathbf{N}^*\}$表示维度索引取值范围。$\boldsymbol{A}$与$\boldsymbol{B}$的余弦相似度计算公式如式(1):

$$\cos(\boldsymbol{A},\boldsymbol{B})=\frac{\boldsymbol{A}\cdot\boldsymbol{B}}{\|\boldsymbol{A}\|_2\cdot\|\boldsymbol{B}\|_2}=\frac{\sum_{i=1}^{m}a_ib_i}{\sqrt{\sum_{i=1}^{m}(a_i)^2}\sqrt{\sum_{i=1}^{m}(b_i)^2}} \tag{1}$$

其中,$\|\cdot\|_2$表示数据向量$\cdot$的二范数。数据向量夹角越大,余弦值越小,表示两者相似程度越低;夹角越小,余弦值越大,表示两者相似程度越高。式(1)表明余弦相似度实质上等于向量$\boldsymbol{A}$、$\boldsymbol{B}$单位化后的乘积,通过将数据点映射到单位超球面上,消除了数据模长的影响,只关注数据向量的方向。这也是余弦相似度对绝对数值(即模长差异)不敏感的原因。以$m=2$为例说明余弦相似度现存的问题,二维数据$\boldsymbol{A}$、$\boldsymbol{B}$的示意图如图1所示,其中θ表示二维向量$\boldsymbol{A}$、$\boldsymbol{B}$的夹角。

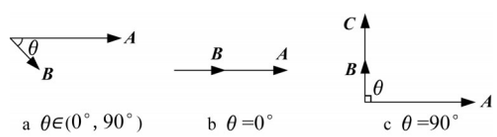

图1 二维向量关系示意图

Fig. 1 Schematic diagram of relationship between two-dimensional vectors

当相似度度量任务需要考虑数据模长差异的时候,式(1)仍然存在2个缺点:

(1) 假设$\boldsymbol{A}=k\cdot\boldsymbol{B}(k\in\mathbf{N}^*,k\neq1)$,由$\cos(\boldsymbol{A},\boldsymbol{B})=1$可以得到数据向量$\boldsymbol{A}$和$\boldsymbol{B}$方向相同即完全相似的结论。但实际如图1b所示,两者虽方向相同但模长相异,意味着余弦相似度对此无法作出准确判断。

(2) 假设$\sum_{i=1}^{m}a_ib_i=0$以及$\sum_{i=1}^{m}a_ic_i=0$,由$\cos(\boldsymbol{A},\boldsymbol{B})=0$和$\cos(\boldsymbol{A},\boldsymbol{C})=0$知,两数据向量只要正交就完全不相似,如图1c所示。因此,余弦相似度并不能进一步比较$\boldsymbol{A}$与$\boldsymbol{B}$、$\boldsymbol{A}$与$\boldsymbol{C}$哪一对数据的相异程度更大。

上述问题增大了余弦相似度计算误差,无法为数据的相似度度量提供一个准确的理论依据。为了能够更加准确地评估数据集的相似程度,改进的相似度计算公式$\mathrm{simi}(\boldsymbol{A},\boldsymbol{B})$必须满足以下原则:①

simi($\boldsymbol{A},\boldsymbol{B}$)需体现出数据各个维度上的差异。② simi($\boldsymbol{A},\boldsymbol{B}$)∈[0,1],并且当simi($\boldsymbol{A},\boldsymbol{B}$)→1,其表征数据的相似度越高;当simi($\boldsymbol{A},\boldsymbol{B}$)→0,其表征数据的相似度越低。③当且仅当$\boldsymbol{A}=\boldsymbol{B}$时,才有simi($\boldsymbol{A},\boldsymbol{B}$)=1。

1.2　余切相似度

数据的相似程度取决于各个维度之间的数值差异,针对余弦相似度的缺点以及新计算公式需遵循的原则,提出2种余切相似度计算公式,该公式侧重于计算数据各维度之间的距离。

1.2.1　两种余切相似度定义

假设数据$\boldsymbol{A}$、$\boldsymbol{B}$进行归一化后得到$\boldsymbol{A}'=(a_1',a_2',\cdots,a_m')$以及$\boldsymbol{B}'=(b_1',b_2',\cdots,b_m')$,其中$a_i',b_i'\in[0,1]$,$i\in\mathcal{M}$,有$0\leqslant|a_i'-b_i'|\leqslant1$。提出的第1种余切相似度具体公式如式(2):

$$\cot_1(\boldsymbol{A},\boldsymbol{B})=\cot\left(\frac{\pi}{4}+\frac{\pi}{4}\times\max_{i\in\mathcal{M}}|a_i'-b_i'|\right)\tag{2}$$

第2种余切相似度具体公式如式(3),其中k表示$|a_i'-b_i'|\neq0$的个数。当$k=0$,表明$\boldsymbol{A}$和$\boldsymbol{B}$完全相似;当$k\neq0$,说明$\boldsymbol{A}$和$\boldsymbol{B}$存在差异,则计算各维度差值的平均值,然后计算数据的相似度。

$$\cot_2(\boldsymbol{A},\boldsymbol{B})=\begin{cases}\cot\left(\dfrac{\pi}{4}+\dfrac{\pi}{4}\times\dfrac{1}{k}\displaystyle\sum_{i=1}^{m}|a_i'-b_i'|\right), & k\neq0\\ 1, & k=0\end{cases}\tag{3}$$

由式(2)、式(3)可知,第1种余切相似度计算公式根据数据维度差值的最大值来比较两者的相似程度,而第2种余切相似度计算公式则是基于数据维度差值的均值来比较两者的相似程度。2种余切相似度公式从2个角度来评判数据的相似程度,为数据集的相似度计算提供了全面的参考依据。

1.2.2　余切相似度分析

基于数学公式对上述2种余切相似度的有效性进行分析。

已知余切函数在区间$(\frac{\pi}{4},\frac{\pi}{2})$单调递减,取值范围为(0,1),符合公式原则。令$|a_s'-b_s'|=\max\limits_{i\in\mathcal{M}}|a_i'-b_i'|$,$s\in\mathcal{M}$。对于第1种余切相似度而言,由于$|a_i'-b_i'|\in[0,1]$,仅当$|a_s'-b_s'|=0$,有$\cot_1(\boldsymbol{A},\boldsymbol{B})=1$,表征数据$\boldsymbol{A}$和$\boldsymbol{B}$完全相似;仅当$|a_s'-b_s'|=1$,有$\cot_1(\boldsymbol{A},\boldsymbol{B})=0$,表征数据$\boldsymbol{A}$和$\boldsymbol{B}$完全不相似;当$0<|a_s'-b_s'|<1$时,$\cot_1(\boldsymbol{A},\boldsymbol{B})\in(0,1)$。

对于第2种余切相似度而言,当且仅当$\forall i\in\mathcal{M}$都有$|a_i'-b_i'|=0$,即$k=0$时,有$\cot_2(\boldsymbol{A},\boldsymbol{B})=1$,表征数据$A$和$B$完全相似;仅当$\frac{1}{k}\sum\limits_{i=1}^{m}|a_i'-b_i'|=1$,有$\cot_2(\boldsymbol{A},\boldsymbol{B})=0$,表征数据$\boldsymbol{A}$和$\boldsymbol{B}$完全不相似;当$0<\frac{1}{k}\sum\limits_{i=1}^{m}|a_i'-b_i'|<1$,有$\cot_2(\boldsymbol{A},\boldsymbol{B})\in(0,1)$。

对2种余切相似度公式进行对比分析:

(1)第1种余切相似度以维度差异峰值作为判别相似度的标准,第2种余切相似度则以维度差异均值作为标准。前者希望相似样本各个维度的数值都能够尽量接近,后者则倾向于相似样本各个维度的数值整体上比较相近,不需要每个维度都十分贴近。

(2)不失一般性地,假设数据$\boldsymbol{A}$与$\boldsymbol{B}$只有前$k(k\in\mathcal{M})$个维度不相同,因为$|a_s'-b_s'|=\frac{1}{k}\times k|a_s'-b_s'|\geqslant\frac{1}{k}\sum\limits_{i=1}^{k}|a_i'-b_i'|$,所以总有$\cot_1(\boldsymbol{A},\boldsymbol{B})\leqslant\cot_2(\boldsymbol{A},\boldsymbol{B})$成立。3.4节实验数据能够佐证这个特点。

综上,对比于余弦相似度,2种余切相似度能够应对数据向量方向相同模长相异的特殊情况,而具体应用哪条公式需要根据实际需求进行选择。

1.3　相似度计算实例说明

为了更清晰地呈现相似度计算过程,列举5个10维数据(已归一化)加以说明,如表1所示。

表1　5个10维数据实例

Tab. 1　Five instances of 10-dimensional data

数据名称	维度1	维度2	维度3	维度4	维度5	维度6	维度7	维度8	维度9	维度10
A	0.372 8	0.478 1	0.361 7	0.684 4	0.520 2	0.293 5	0.320 9	0.838 7	0.551 1	0.253 1
B	0.321 2	0.464 4	0.350 8	0.680 6	0.509 1	0.281 5	0.314 1	0.837 6	0.543 8	0.236 1
C	0.404 2	0.462 7	0.366 2	0.726 8	0.504 3	0.256 6	0.323 8	0.845 6	0.503 0	0.215 2
D	0.409 1	0.524 5	0.284 2	0.673 0	0.623 6	0.282 5	0.197 2	0.846 0	0.621 7	0.209 5
E	0.519 0	0.501 5	0.299 2	0.689 2	0.613 1	0.289 8	0.223 5	0.802 1	0.574 7	0.214 6

根据式(1)、式(2)、式(3)计算出5个数据的相似度矩阵,如表2~表4。其中,第i行第j列数值表示第i个数据和第j个数据的相似度。由于每个数据与自身完全相等,矩阵对角线的相似度均为1。表1中,数据B在维度1、5、7的取值分别是0.321 2、0.5091、0.314 1,数据E在维度1、5、7的取值分别为0.519 0、0.613 1、0.223 5。根据表2~表4可知:$\cos(B,E)=0.988\,0$,$\cot_1(B,E)=0.729\,2$,$\cot_2(B,E)=0.912\,0$。由于B和E夹角只有8.9°,余弦相似度认为数据B和数据E的相似程度非常高;而余切相似度认为两者的相似程度不那么高,尤其是第1种余切相似度认为数据差异比较大。实际上,由于数据已进行归一化,数据B和数据E在维度1、5、7的差值都接近甚至超过0.1,意味着两者的相似程度是比较低的。该计算实例反映了余弦相似度的缺点在于其只关注向量方向差异,当向量夹角比较小时,使用余弦相似度将严重弱化数据间的差异,无法准确判别其相似程度,意味着如果应用到分类等任务中无法对数据进行正确区分。而2种余切相似度能够有效分辨数据各维度的差异,不仅能有效弥补上述不足,当数据向量夹角比较大时,依然有比较好的相似度衡量能力。

表2 余弦相似度矩阵

Tab. 2 Similarity matrix by cosine similarity formula

数据	*A*	*B*	*C*	*D*	*E*
A	1.000 0	0.999 4	0.998 3	0.992 0	0.991 0
B	0.999 4	1.000 0	0.997 7	0.991 2	0.988 0
C	0.998 3	0.997 7	1.000 0	0.989 0	0.995 2
D	0.992 0	0.991 2	0.989 0	1.000 0	0.996 6
E	0.991 0	0.988 0	0.990 4	0.996 6	1.000 0

表3 第1种余切相似度矩阵

Tab. 3 Similarity matrix by first cotangent similarity formula

数据	*A*	*B*	*C*	*D*	*E*
A	1.000 0	0.922 1	0.927 2	0.822 4	0.793 2
B	0.922 1	1.000 0	0.877 4	0.831 4	0.729 2
C	0.927 2	0.877 4	1.000 0	0.818 6	0.834 2
D	0.822 4	0.831 4	0.818 6	1.000 0	0.840 7
E	0.793 2	0.729 2	0.834 2	0.840 7	1.000 0

2 基于BP神经网络的相似度快速计算

传统相似度计算采用遍历方法,通过计算所有数据之间的相似度得到整个数据集的平均相似度,但面对高维数据时,存在运算时间长、内存消耗大的问题。为了提高相似度计算性能,引入BP神经网络,旨在建立一个能够拟合数据集相似度映射关系的网络模型,减少相似度计算时间。

表4 第2种余切相似度矩阵

Tab. 4 Similarity matrix by second cotangent similarity formula

数据	*A*	*B*	*C*	*D*	*E*
A	1.000 0	0.979 0	0.962 6	0.919 9	0.920 1
B	0.979 0	1.000 0	0.960 7	0.914 6	0.912 0
C	0.962 6	0.960 7	1.000 0	0.910 1	0.907 6
D	0.919 9	0.914 6	0.910 1	1.000 0	0.953 3
E	0.920 1	0.912 0	0.907 6	0.953 3	1.000 0

2.1 BP神经网络

BP神经网络是一种按照误差逆传播算法训练的前馈神经网络,具有比较优秀的非线性逼近、自学习、自适应和泛化能力,其应用十分广泛。根据万能逼近定理[15]可知,一个前馈神经网络如果具有线性输出层以及至少一层具有“挤压”性质激活函数的隐藏层,只要有足够多隐藏神经元,可以以任意精度逼近有限维空间内的任意连续函数。由于必须考虑计算时间与空间的开销,实际上使用的多层神经网络通常是放弃苛刻的精确表示,而是在近似表示的基础上寻找合适的参数对数据集与标签集之间的非线性映射关系进行逼近。这为使用BP神经网络进行相似度快速计算提供理论基础。

2.2 相似度快速计算实现

相似度计算公式选定之后,数据的数值与数据之间的相似度可以构成确定的多输入多输出的非线性映射关系,因此可以采用BP神经网络拟合这种映射关系。为了减少遍历计算的时间,基于部分样本的相似度对模型进行训练,用精度换速度,在误差允许范围内表征完备数据集的非线性相似度映射关系,从而计算出完备数据集的近似相似度。

基于余切相似度和BP神经网络的相似度快速计算方法主要分为4个部分:训练样本提取、训练样本相似度计算、网络模型训练以及完备数据集仿真,具体流程如图2。假设数据集P共有n个m维数据,$P=\{\boldsymbol{p}_1,\boldsymbol{p}_2,\ldots,\boldsymbol{p}_n\}$。为了简化表述,令$\mathcal{N}=\{j\in\mathbf{N}^*|1\leqslant j\leqslant n,n\in\mathbf{N}^*\}$表示数据索引取值范围。相似度快速计算的算法伪代码如图3所示,具体步骤如下。

(1)对数据集P进行归一化处理得到$P'=\{\boldsymbol{p}'_1,\boldsymbol{p}'_2,\ldots,\boldsymbol{p}'_n\}$,其中$\boldsymbol{p}'_j(j\in\mathcal{N})$表示经过归一化后的数据集$P'$的第$j$个数据。

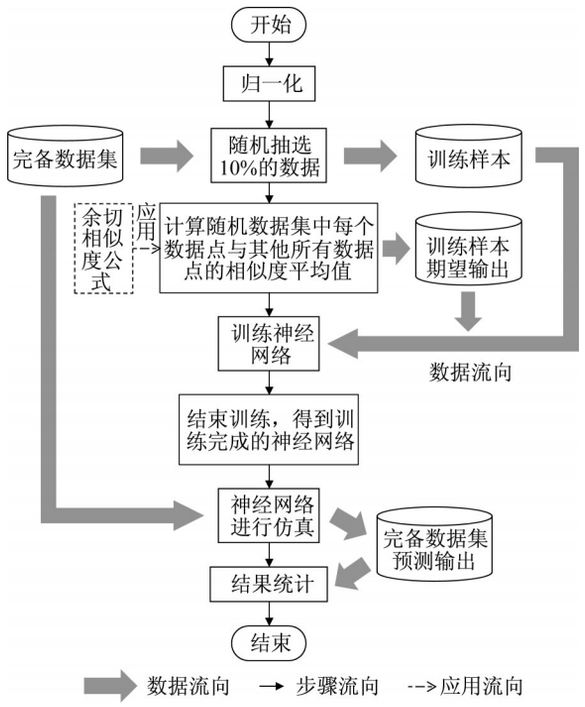

图2　基于余切相似度和BP神经网络的相似度快速计算流程

Fig. 2　Flowchart of fast similarity calculation based on cotangent similarity and BP neural network

(2)随机从归一化数据集P'中抽取10%的数据作为训练样本。该训练集表示为$P'_x=\{\boldsymbol{p}'_{x_1},\boldsymbol{p}'_{x_2},...,\boldsymbol{p}'_{x_k}\},x_1,x_2,...,x_k\in\mathcal{N}$。应根据数据集大小以及维数合理地调整百分比。

(3)使用余切相似度计算训练集P'_x中每一个数据与其他数据的相似度的平均值,以此作为期望输出,设为$Y_x=\left(y_{x_1},y_{x_2},...,y_{x_k}\right)$,其中$y_{x_t}=\frac{1}{n-1}\sum_{j\neq x_t,j=1}^{n}\mathrm{simi}(\boldsymbol{p}'_{x_t},\boldsymbol{p}'_j)$,表示数据$\boldsymbol{p}'_{x_t}$的平均相似度,$x_t\in\{x_1,x_2,...,x_k\}$。

(4)向初始化后的神经网络输入训练集P'_x和期望输出Y_x,训练网络模型,直至误差精度达到要求。

(5)将完备数据集输入训练好的网络模型进行仿真,求得所有近似平均相似度,设为$Y=\left(y_1,y_2,...,y_n\right)$,其中$y_j$表示数据$\boldsymbol{p}'_j$的近似平均相似度。

(6)通过式(4)求所有数据的近似平均相似度的均值,得到完备数据集P的相似度。

$$\mathrm{simi}(P)=\frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i,j=1}^{n}\mathrm{simi}(\boldsymbol{p}'_i,\boldsymbol{p}'_j)=\frac{1}{n}\sum_{i=1}^{n}y_i \tag{4}$$

Input: dataset $P=\{\boldsymbol{p}_1,\boldsymbol{p}_2,...,\boldsymbol{p}_n\}(n\in\mathbf{N}^*)$
1. **Normalize** dataset $P'=\{\boldsymbol{p}'_1,\boldsymbol{p}'_2,...,\boldsymbol{p}'_n\}$
2. Randomly **draw** 10% from P' to get:$P'_x=\{\boldsymbol{p}'_{x_1},\boldsymbol{p}'_{x_2},...,\boldsymbol{p}'_{x_k}\}$, $x_1,x_2,...,x_k\in\mathcal{N}$
3. Calculate **average similarity** of P'_x
　for $t=1,2,...,k$ do
　　for $j=1,2,...,n$ do
　　$\mathrm{simi}(\boldsymbol{p}'_{x_t},\boldsymbol{p}'_j)\leftarrow\cot_1(\boldsymbol{p}'_{x_t},\boldsymbol{p}'_j)\ or\ \cot_2(\boldsymbol{p}'_{x_t},\boldsymbol{p}'_j)$
　　end for
　$y_{x_t}\leftarrow\frac{\sum_{j\neq x_t,j=1}^{n}\mathrm{simi}(\boldsymbol{p}'_{x_t},\boldsymbol{p}'_j)}{n-1}$
　end for
4. Train **BP neural network model**
　while $iter<max\_epochs$ and $Err_{iter}>\delta$ do
　　for $l=1,2,...$ do
　　$O_l\leftarrow\mathrm{sigmoid}(\sum\omega_l O_{l-1}+b_l)$
　　end for
　　$Y_x'\leftarrow\mathrm{relu}(\sum_i\omega_i O_l+b_l)$
　　$Err_{iter}\leftarrow\frac{1}{2}\sum_{t=1}^{k}(y_{x_t}-y'_{x_t})^2$
　　for $l=1,2,...$ do
　　$\Delta\omega_l\leftarrow\frac{\partial Err_{iter}}{\partial\omega_l}$, $\omega_l\leftarrow\omega_l+\Delta\omega_l$
　　$\Delta b_l\leftarrow\frac{\partial Err_{iter}}{\partial b_l}$, $b_l\leftarrow b_l+\Delta b_l$
　　end for
　　$iter\leftarrow iter+1$
　end while
5. Calculate **approximate average similarity** of P
　$\mathrm{simi}(P)\leftarrow\frac{1}{n}\sum_{i=1}^{n}y_i$
Output: $\mathrm{simi}(P)$

图3　基于余切相似度和BP神经网络的相似度快速计算伪代码

Fig. 3　Pseudocode of fast similarity calculation based on cotangent similarity and BP neural network

2.3　快速计算复杂度分析

提出的相似度快速计算方法建立在对训练样本集进行精确的相似度遍历计算的基础上。全数据集遍历计算是一个缓慢且低效的操作:如果存在n个m维数据,计算两两数据之间的相似度共需要$0.5n(n-1)$次相似度计算,每次计算遍历m个维度的算法复杂度为$O(m)$,因此遍历计算时间复杂度为$O\left(0.5\times n(n-1)m\right)\sim O\left(mn^2\right)$,运算时间随着数据量的增加呈指数型增长。但是实际上,在相似度公式确定以及数据量足够的前提条件下,只需要部分数据的相似度即可拟合相似度映射关系,其余的遍历计算可谓冗余计算。换句话说,只要神经网络的参数设置合理,在误差允许的范围内,对部分数据的遍历计算可以取代对全部数据的遍历计算。

现在探讨神经网络方法的时间复杂度,分为精确计算部分和训练测试网络部分。在精确计算过程中,设b是训练集比例,$0<b<1$。bn条训练数据的平均相似度需要进行$0.5bn(n-1+n-bn)=0.5bn[(2-b)n-1]$次相似度计算,总时间复杂度

为$O\left(0.5bmn\left[(2-b)n-1\right]\right)\sim O(mn^2)$,但由于数据量远少于全遍历计算,所以精确计算过程的实际耗时接近遍历计算的b倍($0<b<1$)。在测试n个m维数据时,假设单隐层全连接神经网络的输入神经元数量为m,隐层神经元数量为t,输出神经元数量为1(输入1条m维数据,输出1个相似度),神经网络方法的时间复杂度[16]为$O\left(n(1^2\times 1\times mt+1^2\times 1\times t\times 1)\right)\sim O(mnt)$,在海量高维数据中往往$t\ll mn$。因此,当数据量大或者数据维度高时,神经网络方法在效率上优于遍历计算方法。

3 实验与讨论

首先对比2种余切相似度和余弦相似度公式的计算结果以验证余切相似度的准确性。其次,对比基于BP神经网络和基于遍历计算这2种计算方法检验前者的时效性。其中,相似度算法准确性是指该算法是否能够准确表征数据的相似程度,以相似度计算误差表征;时效性是计算给定数据集平均相似度的效率,用总运算耗时作为评价指标。实验基于MATLAB平台完成。

3.1 数据集介绍

3.1.1 UCI数据集

实验首先采用标准数据库UCI Machine Learning Repository[17-18]中的Iris、Modeling、Eledeal这3个数据集验证2种余切相似度的准确性,数据集具体信息如表5。由于Iris和Modeling数据集比较小,若只抽取10%的数据作为训练数据,很可能因为训练样本不具备表征整个数据集数据分布的能力,致使网络模型欠拟合而无法提供高精度的相似度计算结果。而且,数据维度的增加使得数据集的分布特征更加复杂。因此,这里对Iris和Modeling数据分别抽取75和150个数据作为训练样本。除了数据集规模和数据维度,在实际应用中,还需要根据实际情况和生产需求作调整。

表5 UCI数据集基本信息

Tab. 5 Basic information of UCI datasets

数据集名称	特征维数	数据量	训练集个数	占比/%
Iris	4	150	75	50
Modeling	5	258	150	58
Eledeal	3	10 000	1 000	10

3.1.2 CWRU数据集

为了探讨改进方法对于高维大数据的准确性和时效性,采用美国凯斯西储大学的轴承故障数据集[19-20](Case Western Reserve University,CWRU数据集),将其按照不同数据量和数据维度进行切分成相应子数据集,在数据分布相似的前提下,说明和分析当数据量和数据维度增加时所得相似度误差和计算时间的变化。设CWRU-N-M表示包含N个M维数据的子数据集,例如CWRU-512-100表示512个100维数据集。实验中数据量取值512、1 024、5 120、10 240,特征维数取值100、500、1 000、2 000,共有16个子数据集。根据10%抽取训练样本原则,设定各数据量对应的训练样本数分别为100、102、512、1 024,其中为了保证模型精度,512数据量训练集比例约20%。

3.2 数据集归一化

在计算相似度之前先进行归一化处理,避免由数据度量标准不统一引起的误差。令数据集P的第j个数据$\boldsymbol{p}_j=(p_{j,1},p_{j,2},\cdots,p_{j,m})$,其中$p_{j,i}(i\in\mathcal{M},j\in\mathcal{N})$表示数据$\boldsymbol{p}_j$的第$i$个维度的数值。令归一化数据集$P'$的第$j$个数据$\boldsymbol{p}'_j=(p'_{j,1},p'_{j,2},\cdots,p'_{j,m})$,其中$p'_{j,i}(i\in\mathcal{M},j\in\mathcal{N})$表示归一化数据$\boldsymbol{p}'_j$的第$i$个维度的数值。考虑到各个维度的值域范围有可能差异过大,为了保留数据的原始特征,采用min-max标准化分别对每个维度进行处理,保持数据与最小值的距离比例,计算如式(5)所示:

$$p'_{j,i}=\frac{p_{j,i}-\min\limits_{k\in\mathcal{N}}\{p_{k,i}\}}{\max\limits_{k\in\mathcal{N}}\{p_{k,i}\}-\min\limits_{k\in\mathcal{N}}\{p_{k,i}\}},i\in\mathcal{M}\tag{5}$$

3.3 实验设置与结果

根据2.1节所述,可知具有单隐层、线性输出层以及足够多隐藏神经元的前馈神经网络能够无限逼近任意连续函数。因此在对训练时间成本和模型精度做出权衡后,实验采用单隐层全连接神经网络,模型训练的部分超参数设置如下:隐层神经元个数为10,学习率为0.01,迭代停止精度为0.000 1,最大迭代次数分别为500次(UCI数据集)和1 000 000次(CWRU数据集)。准确性实验验证结果如表6和图4,时效性实验验证结果如表7。为了简化表述,下文图表使用cos、$\cot_1$、$\cot_2$分别表示基于余弦相似度、2种余切相似度的实验结果。

3.4 准确性分析

算法准确性的验证对于不同规模的数据集都有重要意义,而且确保算法的准确性是分析其时效性的前提。基于表6和图4对UCI数据集和CWRU数据集2次实验的准确性进行分析验证。

表6　基于神经网络和遍历计算的相似度计算结果(UCI数据集)

Tab. 6　Similarity calculation results based on BP network and traversal method(UCI datasets)

数据集名称	计算方法	cos	$\cot_1$	$\cot_2$
Iris	遍历计算	0.895 8	0.484 4	0.544 1
	神经网络	0.892 0	0.472 5	0.533 3
	误差	0.424 2%	2.456 6%	1.984 9%
Modeling	遍历计算	0.892 8	0.406 5	0.472 9
	神经网络	0.886 4	0.403 9	0.469 0
	误差	0.716 8%	0.639 6%	0.824 7%
Eledeal	遍历计算	0.953 6	0.437 7	0.632 6
	神经网络	0.952 3	0.435 3	0.631 9
	误差	0.136 3%	0.548 3%	0.110 7%

3.4.1　相似度公式差异分析

以表6中Iris数据集为例,$\cos\left(P_{\text{Iris}}\right)=0.8958$,$\cot_1\left(P_{\text{Iris}}\right)=0.4844$,$\cot_2\left(P_{\text{Iris}}\right)=0.5441$,显然余弦相似度判定数据走向趋势相近,过高估计了该数据集的平均相似度,因而准确性比较差;而余切相似度从维度上判别Iris数据集的差异比较大,其平均相似度比较低。不难发现,表6中3个数据集都有$\cot_1(P)<\cot_2(P)$(P表示数据集名称),这是由于第1种余切相似度侧重于比较数据维度差异的峰值,第2种余切相似度倾向于比较数据维度差异的均值。在相同数据集的情况下,前者的计算值总是不超过后者的计算值。

图4是以遍历计算的相似度为基准2种计算方法之间的相似度误差。第1种余切相似度具有最大的计算误差,而余弦相似度和第2种余切相似度的计算误差比较小,这是由于第1种余切相似度使用max函数,拟合难度相对较大,但可通过调整模型超参数进一步提升。

3.4.2　计算方法差异分析

表6中Iris数据集基于遍历计算和基于神经网络的余弦相似度误差为0.424 2%,第1种余切相似度误差为2.456 6%,第2种余切相似度误差为1.984 9%。总体来说计算误差都比较小,均在容许范围内。这很好地说明了网络模型能够较好拟合训练样本的相似度映射关系,并且具备一定的泛化能力,对完备数据集的输入具有良好的鲁棒性。图4中2种计算方法在CWRU数据集的计算误差都小于4%,对于大部分应用场景均在误差允许范围内,说明在高维数据集中,神经网络方法依然保有足够的准确度,为后文的时效性分析提供前提条件。综上所述,基于余切相似度和BP神经网络的相似度计算方法应用在数据集相似度评估中具有令人满意的准确性。

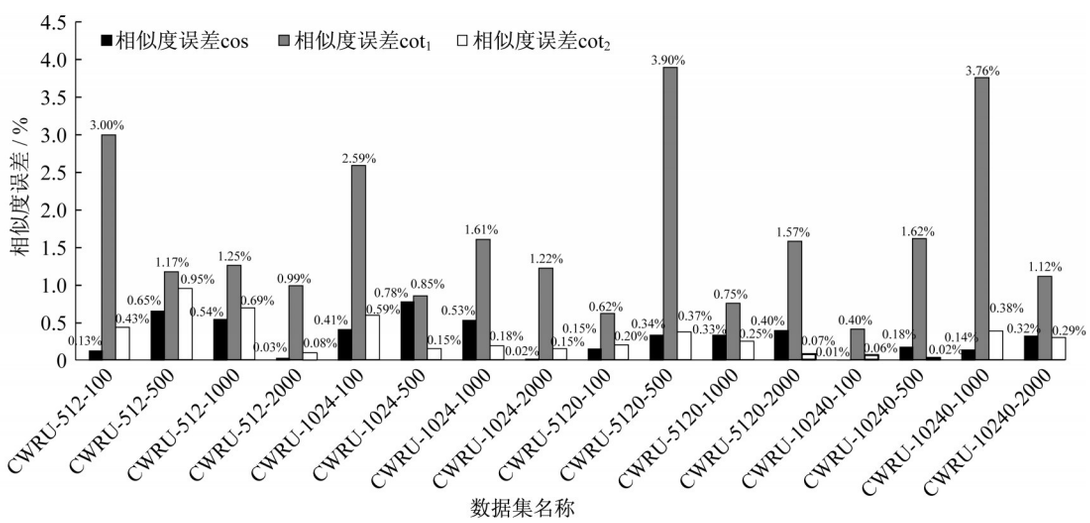

图4　基于神经网络和遍历计算的相似度计算误差(CWRU子数据集)

Fig. 4　Similarity calculation error based on neural network and traversal calculation(CWRU subdatasets)

3.5　时效性分析

一般来说,计算量少时不同算法之间的效率差异比较小,运算复杂度大的算法或程序才更有统计耗时的必要。由于CWRU数据集的“大规模”、“高维”特性相比于UCI数据集更加明显,因此着重分析改进方法在CWRU数据集的时效性能。为了更清晰地对比分析基于不同计算公式以及基于不同计算方法的相似度计算效率,将表7数据依据不同相似度公式以及依据不同的计算方法分别绘制相关趋势图像,如图5、6所示,注意纵坐标均经过对数缩放

表7 基于神经网络和遍历计算的相似度计算时间(CWRU子数据集)

Tab. 7 Running time of similarity calculation based on neural network and traversal calculation(CWRU sub-datasets) s

子数据集名称	计算方法	cos	cot_1	cot_2	子数据集名称	计算方法	cos	cot_1	cot_2
CWRU－512－100	遍历计算	0.184 8	0.077 4	0.189 8	CWRU－5120－100	遍历计算	16.710 4	8.590 0	17.814 4
	神经网络	0.517 7	0.973 6	0.247 6		神经网络	3.010 3	2.573 1	2.929 6
CWRU－512－500	遍历计算	0.751 5	0.310 6	0.635 9	CWRU－5120－500	遍历计算	96.739 5	49.093 7	81.472 7
	神经网络	0.389 0	1.708 0	0.480 2		神经网络	10.622 9	6.578 4	8.740 2
CWRU－512－1000	遍历计算	4.769 7	2.012 6	2.478 2	CWRU－5120－1000	遍历计算	931.890 7	482.002 1	524.509 1
	神经网络	0.542 5	0.853 1	0.600 8		神经网络	58.497 9	30.530 5	34.887 5
CWRU－512－2000	遍历计算	19.410 1	7.951 5	7.711 0	CWRU－5120－2000	遍历计算	2 694.014 7	1 068.391 8	1 078.569 9
	神经网络	2.088 6	1.914 6	0.999 9		神经网络	203.104 6	86.689 7	94.687 6
CWRU－1024－100	遍历计算	0.623 3	0.298 2	0.700 4	CWRU－10240－100	遍历计算	89.189 4	45.859 1	79.518 2
	神经网络	0.269 4	0.900 8	0.338 2		神经网络	11.861 6	9.667 6	11.071 0
CWRU－1024－500	遍历计算	3.293 6	1.653 3	5.636 5	CWRU－10240－500	遍历计算	372.570 1	190.164 6	325.160 5
	神经网络	0.608 9	0.625 6	0.707 5		神经网络	52.618 8	29.803 5	37.369 1
CWRU－1024－1000	遍历计算	30.834 9	13.663 8	14.997 3	CWRU－10240－1000	遍历计算	3 958.858 8	1 487.528 7	1 700.750 4
	神经网络	1.871 2	1.495 7	1.283 5		神经网络	873.362 9	166.165 7	187.560 9
CWRU－1024－2000	遍历计算	93.154 7	38.217 4	36.930 4	CWRU－10240－2000	遍历计算	10 458.184 1	4 696.142 3	5 278.172 9
	神经网络	7.416 8	3.640 6	3.709 1		神经网络	1 022.378 6	399.780 6	450.484 4

操作。

3.5.1 相似度公式差异分析

参照表7和图5,CWRU-512-100子数据集在遍历计算的基础上3种相似度的耗时分别为0.184 8s、0.077 4s、0.189 8s,由于数据量和特征维数相对其他子数据集而言较低,运算效率差异比较小。但随着数据维度和数据量的增长,差异逐渐显现:

(1)如果数据量固定为512,当特征维数从100增长到2 000时,大体上呈现出余弦相似度的计算时间最长、第2种余切相似度次之、第1种余切相似度最短的规律。

(2)如果特征维数固定为100,当数据量从512增长到10 240时,同样呈现出第(1)点所述规律。

而在神经网络方法的基础上,当数据量和特征维数比较少时,不同相似度在运算效率上没有绝对的优劣。根据表7,CWRU-512-100这3种相似度耗时分别为0.517 7s、0.973 6s、0.247 6s,CWRU-512-500则是0.389 0s、1.708 0s、0.480 2s,此时3种相似度公式还没有必然的大小关系。由于训练样本少,因此遍历计算部分少,此时主要由神经网络的训练决定运算时间。当数据量和特征维数增大时,遍历计算部分逐渐主导运算时间的长短。表7中,当数据量达到1 024及以上,神经网络方法的运算时间同样呈现出余弦相似度耗时最长、第2种余切相似度次之、第1种余切相似度最短的规律。

上述现象是由于余弦相似度计算了2次L_2范数,时间复杂度最大;第2种余切相似度因比第1种

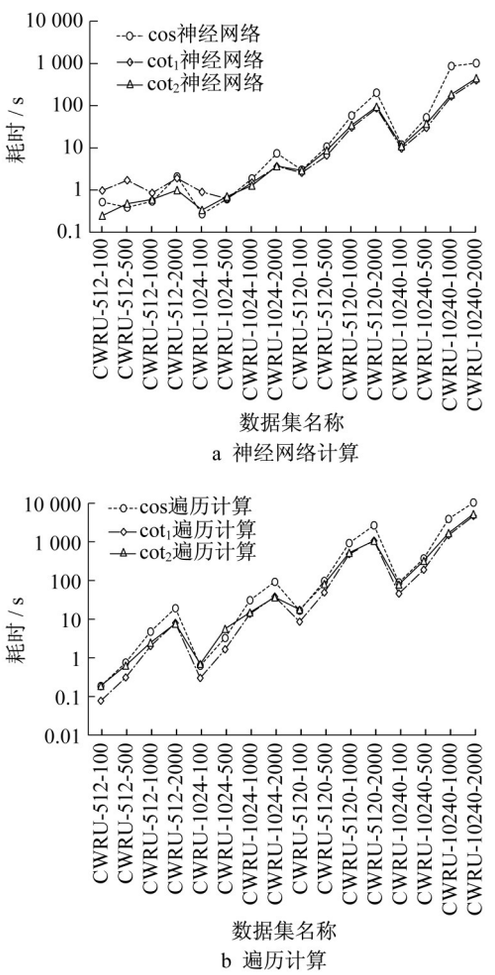

图5 基于不同计算公式的相似度计算时间对比

Fig. 5 Comparison of running time of similarity calculation based on different calculation formulas

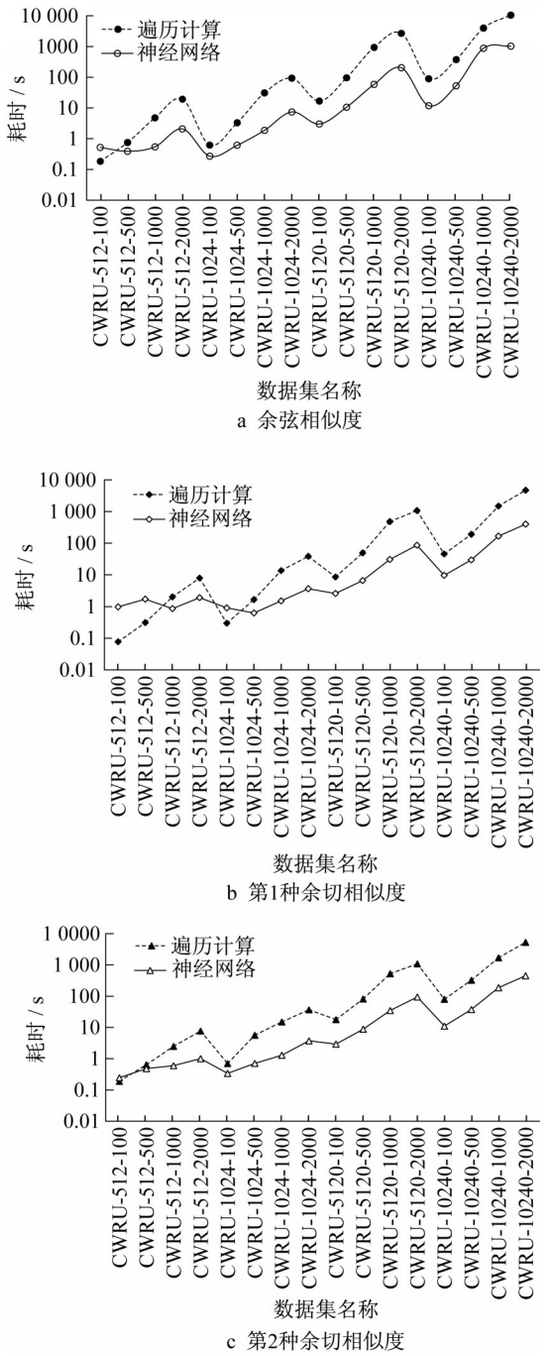

图6　基于不同计算方法的相似度计算时间对比

Fig. 6　Comparison of running time of similarity calculation based on different calculation methods

余切相似度多1个统计非零维数步骤，运算速度相对较慢。总体而言，无论是基于遍历计算还是基于神经网络方法，2种余切相似度较余弦相似度都有一定的时效优势，尤其是针对海量高维数据。

3.5.2　计算方法差异分析

参照表7和图6，当数据量和维数比较小的情况下，遍历计算的速度比神经网络方法要快但相差不大。以第1种余切相似度为例，表7中CWRU-512-100基于遍历计算的耗时0.077 4s比神经网络方法0.973 7s略少。随着数据集规模的扩大，遍历计算的耗时开始超过神经网络方法，其增长速度也远远高于后者，如CWRU-5120-100的遍历计算耗时8.590 0s，是神经网络耗时2.573 1s的3倍多，相对于CWRU-512-100，前者增长了100多倍，后者仅增长了2倍多。由于工业大数据的数据量一般都十分庞大，可以预想，当数据量和维数继续增加到一定程度时，遍历计算将因为巨大的空间和时间代价而无法继续直接计算，而此时神经网络方法的计算成本依然能被接受。因此，神经网络方法有效提高了相似度计算效率，在评估大规模数据集相似度时比遍历计算方法更加适用。

根据上述分析，相比于遍历计算，神经网络计算方法在时效性方面有了很大的提升，更适用在大数据相似度评估工作中。至此，基于BP神经网络的相似度快速计算方法的时效性得以验证。

4　结语

从大数据相似度评估工作切入，针对余弦相似度准确性较差、遍历计算方法时间复杂度大的问题，提出了2种余切相似度公式以及基于BP神经网络的相似度快速计算方法，并基于Iris等经典数据集进行验证。实验证明，改进相似度计算方法在面对小规模低维数据集和海量高维数据集时都能保持良好的准确性和时效性。作为对传统余弦相似度计算方法的一种改进和补充，本文提出的余切相似度快速计算方法既能改善传统余弦相似度只关注数据向量夹角而忽略模长的局限性，又在大规模高维数据集相似度计算方面表现出较好的适应性。

提出的改进相似度计算方法主要应用于结构化数据，因此未来的研究工作应该对非结构化数据和半结构化数据提出更加有针对性的相似度度量方法。其次，改进算法针对以二维数组描述的数据集，但不适用于以树、链表等复杂数据结构表达的数据集，今后的工作会考虑向复杂数据结构的相似度度量领域开展。最后，可以考虑引入卷积神经网络等深度学习方法进行计算，进一步提高算法的运算效率。

作者贡献申明：

乔　非：研究工作的思路与全程指导。
关柳恩：研究工作的完善与总结。
王巧玲：初步的研究工作。

参考文献：

[1] KUPPILI V, BISWAS M, EDLA D R, *et al*. A mechanics-based similarity measure for text classification in machine learning paradigm[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2020, 4(2): 180. DOI: 10.1109/TETCI.2018.2863728.

[2] DUBEY V K, SAXENA A K. A cosine-similarity mutual-information approach for feature selection on high dimensional datasets [J]. Journal of Information Technology Research, 2017, 10(1): 15. DOI: 10.4018/JITR.2017010102.

[3] LIN Liang, WANG Guangrun, ZUO Wangmeng, *et al*. Cross-domain visual matching via generalized similarity measure and feature learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1089. DOI: 10.1109/TPAMI.2016.2567386.

[4] EGHBALI S, TAHVILDARI L. Fast cosine similarity search in binary space with angular multi-index hashing [J]. IEEE Transactions on Knowledge & Data Engineering, 2019, 31(2):329. DOI: 10.1109/TKDE.2018.2828095.

[5] JIA Ke, WANG Congbo, BI Tianshu, *et al*. Transient current waveform similarity-based protection for flexible dc distribution system[J]. IEEE Transactions on Industrial Electronics, 2019, 66(12): 9301. DOI: 10.1109/TIE.2019.2891457.

[6] XIA Peipei , ZHANG Li, LI Fanzhang . Learning similarity with cosine similarity ensemble [J]. Information Sciences, 2015, 307:39. DOI: 10.1016/j.ins.2015.02.024.

[7] LIU Chengjun. Discriminant analysis and similarity measure [J]. Pattern Recognition, 2014, 47(1):359. DOI: 10.1016/j.patcog.2013.06.023.

[8] 蒋欣, 王开军, 陈黎飞. 基于改进余弦相似度的粒子滤波故障预报[J]. 计算机系统应用, 2015, 024(1):98.
JIANG Xin, WANG Kaijun, CHEN Lifei. Particle filter fault prediction based on improved cosine similarity [J]. Computer Systems & Applications, 2015, 024(1): 98.

[9] YANG Junhua, YONG Li, WEI Cheng, *et al*. EKF - GPR-based fingerprint renovation for subset-based indoor localization with adjusted cosine similarity[J]. Sensors, 2018, 18(2): 318. DOI: 10.3390/s18010318.

[10] YE Jun. Improved cosine similarity measures of simplified neutrosophic sets for medical diagnoses [J]. Artificial Intelligence in Medicine, 2015, 63(3): 171. DOI: 10.1016/j.artmed.2014.12.007.

[11] WEI Guiwu. Some cosine similarity measures for picture fuzzy sets and their applications to strategic decision making [J]. INFORMATICA, 2017, 28(3): 547 - 564. DOI: 10.15388/Informatica.2017.144.

[12] YE Jun. Single-valued neutrosophic similarity measures based on cotangent function and their application in the fault diagnosis of steam turbine [J]. Soft Computing, 2017, 21(3): 817. DOI: 10.1007/s00500-015-1818-y.

[13] HAYASHI T, SATO A. Fast similarity retrieval of vector images using representative queries [C]//IEEE International Symposium on Multimedia. Anaheim: IEEE, 2013: 498-499. DOI: 10.1109/ISM.2013.95.

[14] TANIOKA H. A Fast content-based image retrieval method using deep visual features [C]//2019 International Conference on Document Analysis and Recognition Workshops (ICDARW). Sydney: IEEE, 2019: 20-23.

[15] GOODFELLOW I, BENGIO Y, COURVILLE A. Deep learning[M]. Cambridge: MIT Press, 2016.

[16] YU Wanke, ZHAO Chunhui. Broad convolutional neural network based industrial process fault diagnosis with incremental learning capability [J]. IEEE Transactions on Industrial Electronics, 2020, 67(6): 5081. DOI: 10.1109/TIE.2019.2931255.

[17] YU Zhiwen, LUO Peinan, YOU Jane, *et al*. Incremental semi-supervised clustering ensemble for high dimensional data clustering [J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(3): 701. DOI: 10.1109/TKDE.2015.2499200.

[18] JAN Z, VERMA B. Multiple strong and balanced cluster-based ensemble of deep learners [J]. Pattern Recognition, 2020, 107: 107420. DOI: 10.1016/j.patcog.2020.107420.

[19] SMITH W A, RANDALL R B. Rolling element bearing diagnostics using the Case Western Reserve University data: a benchmark study [J]. Mechanical Systems and Signal Processing, 2015, 64: 100-131. DOI: 10.1016/j.ymssp.2015.04.021.

[20] MARINS M A, RIBEIRO F M L , NETTO S L , *et al*. Improved similarity-based modeling for the classification of rotating-machine failures[J]. Journal of the Franklin Institute, 2018, 355(4): 1913. DOI: 10.1016/j.jfranklin.2017.07.038.