

基于K-means优化的SOM神经网络算法的视频推荐系统

付丽梅

(大连东软信息学院软件工程系, 辽宁 大连 116023)

✉ fulimei@neusoft.edu.cn



摘要: 为解决视频推荐系统中推荐精度不够精准的问题, 提出一种K-means优化的自组织映射(Self-organizing Map, SOM)神经网络视频推荐方法。首先, 爬取视频网站的数据并对其进行处理; 其次, 将处理后的数据输入K-means算法优化的SOM神经网络中, 得到聚类结果; 最后通过计算归类视频的弹幕数量、点击量、评分等推荐出优秀的视频。文中系统的预期结果为主界面选择分类并输入关键词之后, 通过算法计算, 为用户推荐感兴趣的视频, 并按评分高低列出视频的超链接。实验结果表明, 优化的SOM算法在视频推荐的精度上提升了5%—8%。

关键词: 视频推荐; K-means; SOM算法; 优化

中图分类号: TP391.41 **文献标识码:** A

Video Recommendation System based on SOM Neural Network Algorithm Optimized by K-means

FU Limei

(Department of Software Engineering, Dalian Neusoft University of Information, Dalian 116023, China)

✉ fulimei@neusoft.edu.cn

Abstract: Aiming at the insufficient recommendation accuracy in video recommendation systems, this paper proposes a Self-organizing Map (SOM) neural network video recommendation algorithm optimized by K-means. Firstly, video website data is crawled and processed; secondly, the processed data is input into the SOM neural network optimized by K-means algorithm to obtain the clustering result; finally, relatively excellent videos are recommended by calculating the number of bullets, clicks and scores of classified videos. The expected result of the proposed system is that after selecting categories and inputting keywords on the main interface, the videos that interest the users will be recommended by the algorithm after calculation, and the hyperlinks of the videos will be listed according to ratings. Experimental results show that the optimized SOM algorithm improves the accuracy of video recommendation by 5%-8%.

Keywords: video recommendation; K-means; SOM algorithm; optimization

1 引言(Introduction)

智能手机的高速发展使各种小视频成为人们生活中的重要娱乐手段, 随之而来的视频推荐已成为当前视频应用的一个关键问题。当前视频推荐的精度和速度都还有一定的提升空间, 可通过对SOM算法的研究改进视频推荐系统的精度和速度。目前SOM神经网络的研究方向主要有以下几种^[1-3]: (1)基于动态确定结构和网络数目的改进。为了摆脱传统SOM模型需要提前给定结构和网络单元数目的限制, 科研人员提

出在训练过程中动态确定网络形状和数目的解决方案。(2)基于匹配神经元策略的改进。该研究方向中比较著名的研究算法有SOFM-CV、SOFM-C、DSOM等, 其改进方向为修改神经元的竞争方式或竞争过程, 或者在竞争过程中添加其他参数和限制条件等, 以防止竞争单元层中出现“始终不能获胜”的结果。(3)最后一种就是用其他算法来组合SOM算法, 其中比较有代表性的是提出把SOM和粗糙集进行组合的RSOM算法; 科研人员提出使用自适应共振理论模型和

SOM组合对文档进行聚类；另有科研人员使用了多层感知器(Multilayer Perceptron, MLP)和SOM结合来进行语音识别。本文使用K-means算法对SOM神经网络算法进行改进,并应用到视频搜索推荐系统中,该推荐系统分为爬虫、算法实现、可视化三个模块。

2 算法概述(Algorithm overview)

2.1 SOM算法

SOM是一种自组织特征映射聚类算法,它包括输入层和竞争层(也叫输出层)两部分^[4]。训练过程采用竞争的方式,竞争层在接收到输入层的样本数据后,计算样本与神经元自身的权向量,与样本最近的神经元为最佳匹配单元,接着更新最佳匹配单元的权值,同时和最佳匹配单元临近的点也根据它们距最佳匹配单元的距离适当更新参数,这个过程迭代反复直至收敛。

2.2 K-means算法

K-means算法是一种无监督的聚类算法^[5],其思想如下:将给定的样本集按照样本之间的距离大小划分为 K 个簇,划分时要让簇内部点的排列尽量紧密,簇与簇的间距相对要尽量大。K-means算法的 K 值一般是根据对数据的先验经验来选择,若先验知识不足,也可通过交叉比对选择。 K 个质心的选择可以采用随机方式,质心的初始位置会极大地影响最终的聚类结果及运行时间,质心之间的距离不应太近,否则会影响聚类效果。

2.3 K-means改进的SOM算法

SOM算法作为一种无监督的学习算法,它的训练过程不是很稳定,如果有一个如同K-means算法的稳定训练过程,就可以大大提升工作效率。SOM算法的训练过程需要刷新初始值较小的神经元,和K-means算法选定质心的方式有着相似之处,只不过一个是在输入层之前选定,一个是在训练过程之中选定。本文综合两种算法对视频推荐算法进行优化。

3 基于改进SOM算法的视频推荐系统实现(Implementation of video recommendation system based on improved SOM algorithm)

3.1 数据采集

数据采集使用爬虫技术,爬取www.bilibili.com网页中较火的视频类数据进行处理后作为实验数据^[6]。爬虫的编写采用Python 3.7自带的urllib模块,当用户打开网页时,浏览器根据输入的地址找到相应的服务器发送请求,服务器收到请求后,解析请求中携带的信息并进行响应,最终返回请求结果。urllib模块模拟用户的浏览过程,发送请求后获取服务器返回的数据。获取网页HTML文件后,使用BeautifulSoup库解析HTML文件。BeautifulSoup为开发者提供一些Python风格的函数来分析提取HTML文档中的信息,通过解析HTML筛选出有用的信息导入数据库以备算法使用。实验需要的信息包括哔哩哔哩视频弹幕网的视频播放量、点击量、评论及弹幕数量等数据^[7],至于视频的编号(AV)则是通过对URL的

拆分处理而得到的。拆分的过程就是利用split函数以“/”为分隔,将字符串切割成列表的形式。

3.2 SOM算法实现

SOM算法的学习过程分为三个部分。首先是竞争,也就是针对输入的数据集,计算其与数据单元权向量的欧几里得距离,距离最小的为获胜神经元,也可通过其他判别函数得出,记为最佳匹配单元。然后是合作,最佳匹配单元决定了兴奋的神经元的拓扑邻域占据的空间位置,是相邻的神经元合作的基础。最后要调整权值,兴奋的神经元通过对自身权向量的调整,来增加数据集的判别函数值(使用向量间的欧几里得距离),然后让神经元对接下来的相似输入有一个强化的响应。

SOM算法的实现过程可描述为如下几方面。

(1)初始化。使用权值较小的随机值进行初始化,并对输入向量和权值做归一化处理:

$$x' = x / \|x\|$$

$$\omega'_i = \omega_i / \|\omega_i\|, 1 \leq i \leq m$$

其中, $\|\omega_i\|$ 和 $\|x\|$ 分别为权值向量和输入样本向量的欧几里得范数。

(2)将样本输入网络。计算样本与权值向量的欧几里得距离,距离最小的神经元赢得竞争,记为最佳匹配单元。

(3)更新权值。更新获胜的神经元拓扑邻域内的神经元,重新归一化学习后的权值,基本公式如下:

$$\omega(t+1) = \omega(t) + \eta(t, n)(x - \omega(t))$$

其中, $\eta(t, n) = \eta(t)e^{-n}$, η 为学习率,它是关于训练时间 t 及与获胜神经元拓扑距离 n 的函数。

(4)更新学习速率 η 及拓扑邻域 N , N 随时间的增大而变小。

(5)判断是否收敛。如果达到预设的迭代次数或学习率 $\eta \leq \eta_{\min}$,则算法结束。

该算法的优点是无须监督,无须提前告知分类数便能自动对输入模式进行聚类,容错性强,对异常值和噪声不敏感。其缺点是在训练时有些神经元始终不能胜出,导致分类结果不准确,SOM网络收敛时间较长,运算效率较低。

3.3 K-means算法实现

假设输入样本集为 $D = \{x_1, x_2, \dots, x_m\}$,聚类的簇数量为 K ,最大迭代次数为 N ,输出簇为 $C = \{C_1, C_2, \dots, C_k\}$ 。K-means算法的基本流程如下所述。

(1)随机选择数据集 D 中的 K 个样本作为初始质心,其集合为 $\{\mu_1, \mu_2, \dots, \mu_k\}$,划分并初始化输出簇 C ,初始化结果为

$$C_t = \emptyset \quad t = 1, 2, \dots, k$$

(2)对于每个簇 C 中的样本 x_i ,计算每一个质心向量和该样本的距离 d ,并对比 d 的大小,将该样本并入相对应的簇中,更新输出簇 C 。

(3)重新计算每个簇的新质心,新质心为各个簇内所有样本距离的平均值,若 K 个质心向量未发生变化,则进行步骤(4);否则,重复步骤(1)—步骤(3),直至最大迭代次数或

聚类完成。

(4)重新划分输出簇C。

3.4 使用K-means改进SOM算法

SOM算法具有一些缺陷,例如算法运行后期有可能会出现钟摆效应(即网络在几个最佳极值点之间反复跳动),以及不稳定的训练输出等,可结合K-means算法的训练过程来使SOM算法的训练过程稳定化,并且借用K-means算法的思想来强化SOM算法的效率。

改进的思想大致如下:首先,SOM算法需要随机选定神经元,随机性会导致后面的训练过程不稳定;而K-means是首先选定质心,在训练过程的稳定性上占有优势。其次,SOM算法以神经元为中心,使得数据向神经元移动;而K-means算法是通过移动质心的方式来达成对数据类的分簇。综合这两种思想,首先观察数据集的分布状态判定质心,调用K-means算法的训练过程将质心移动到分簇的附近,然后将选定得到的质心作为现成的最佳神经元定位拓扑邻域,使用SOM算法开始聚合。改良后的算法的基本流程如图1所示。

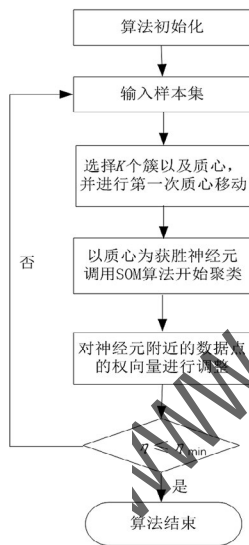


图1 使用K-means改进SOM算法流程

Fig.1 Using K-means to improve SOM algorithm process

改良后的算法虽然以SOM算法为主要运行部分,但是使用K-means算法指定的质心取代了SOM算法需要算法运行选择的最佳神经元,并且在经过质心选定之后使得质心/最佳神经元更靠近需要聚类的数据簇。该算法的优点是算法简单,收敛速度快,准确率较高;缺点是初始聚类中心的设定对聚类结果影响较大,聚类种数需要预先给定,而在很多情况下K的估计是非常困难的。

3.5 视频推荐结果可视化

本系统的可视化模块采用了Django框架进行开发,可可视化模块分为关键字搜索和结果显示两个页面。关键字搜索页面在提供自定义搜索的同时,也提供标签式的定向搜索,

即根据用户选择的标签推送出最吻合的视频,满足不同类型用户的需要。结果显示页面除了提供视频的超链接之外,还提供该视频的评分项,计算方法是点击数、播放数、评论数、弹幕数等属性进行归一化处理后乘以100得出基本评分,方便用户选择视频^[8]。本系统采用MySQL数据库,需要安装PyMySQL模块。系统的数据流向如下:启动爬虫,对网页进行数据爬取,经过一个临时数据存储模块处理后进入数据库,算法在数据库中提取数据,通过分析后输出推荐视频到结果页显示。

4 算法实验结果与对比(Experimental results and comparison of the algorithm)

4.1 算法参数设置实验

本文对SOM算法单体效率的研究采用修改参数调整效率的方式,大部分参数可以通过学习获得,能够手动修改且对运行结果影响较大的是迭代次数及批尺寸(batch_size)。

(1)迭代次数实验

在这一步中,本文使用了500个样本数据,在不影响batch_size的情况下进行改动迭代次数的实验,实验使用单一的SOM算法。实验结果表明,在不修改batch_size的情况下,单纯地修改迭代次数对算法的运行效率几乎没有影响。

(2)batch_size实验

batch_size是机器学习算法中一个重要的参数,本实验以已经完成的组合算法为框架,输入200个样本,通过调整batch_size不断实验可以看出,如果batch_size减小,算法在200次迭代内不能收敛。如果batch_size增大,处理数据的速度会变快,而达到相同精度所需的迭代数量增多。当batch_size增大到一定程度时,会达到运行时间的最优化,但最终的收敛精度会陷入不同的极值。因此,batch_size需要在0—200取舍,本实验的batch_size定为100。

4.2 K-means改进SOM算法实验

第一组测试的是运行所需要的时间。实验限制条件为batch_size100,迭代次数100,只对输入样本的个数进行更改。实验结果为算法的运行时间,如表1所示。

表1 算法运行时间对比

Tab.1 Comparison of algorithm running time

算法	不同样本个数的运行时间/s								
	50	100	200	500	800	1,000	2,000	2,500	3,000
SOM	0.0163	0.1096	0.2247	0.5168	0.8619	1.0509	2.1008	2.6433	3.1814
改良算法	0.0183	0.1167	0.2446	0.5417	1.0064	1.0962	2.1928	2.8225	3.4723

由表1可以得出,在限制了迭代次数及batch_size的情况下,改良后的算法在运行时间上要略差于原本的SOM算法,其运行时间比原算法长4%—12%。

第二组实验测试的是同等运行条件下的运行精度,参考值为输出关键字的精确性。实验结果为用户需求的关键词在推荐结果中所含数量占算法推荐数总量的百分比,如表2所示。

(下转第7页)