

结合概率型神经网络(PNN)和学习矢量量化(LVQ)算法的文本分类方法^①

李 敏, 余正涛

(昆明理工大学 信息工程与自动化学院, 昆明 650051)

摘 要: 针对文本自动分类问题, 提出一种基于概率型神经网络(PNN)和学习矢量量化(LVQ)相结合的文本分类算法, 该方法借助 TFIDF 方法提取文本特征及特征值, 形成文本分类特征向量, 利用概率型神经网络构建分类模型, 并利用 LVQ 学习算法对神经网络模型竞争层网络进行学习, 使相应模式向量相互靠拢, 远离其他模式, 从而实现文本分类. 实验结果表明, 提出的该方法在文本分类中表现了很好的效果, 不仅具有很好的分类准确率, 还表现出很好的学习效率.

关键词: 文本分类; 概率型神经网络; LVQ 学习算法; 特征提取;

Text Classification Combined with Probabilistic Neural Network (PNN) and Learning Vector Quantization (LVQ) Algorithm

LI Min, YU Zheng-Tao

(School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650051, China)

Abstract: Aiming at the problem of text classification, one text classification method based on the probabilistic neural network (PNN) and learning vector quantization (LVQ) is proposed. The text features and feature values are extracted by use of TFIDF method, and text categorization feature vector are formed. In addition, classification model based on probabilistic neural network can be constructed and the learning of competitive layer network is completed by using LVQ algorithms, so the corresponding pattern vector to move closer to each other, away from the other modes, thereby realizing text classification. The experimental results show that the method in the text classification performance with very good results, and not only has good classification accuracy, but also shows a good learning efficiency.

Key words: text classification; probabilistic neural network; LVQ learning method; feature extraction

1 引言

文本分类是文本信息处理的一个重要研究领域, 对提高文本检索、文本存储等应用的处理效率有着重要意义. 国内外在文本分类技术以及相关的信息检索、信息抽取等领域已经进行了比较深入的研究, 取得了不少研究成果. 常见的文本分类技术有神经网络方法、最小距离方法、朴素贝叶斯方法、KNN 方法、支持向量机方法(SVM)等. 在这些分类算法中, 神经网络的文本分类更具有优越的性能^[1]. 目前国内对于

神经网络在文本分类中的应用研究也主要是在 BP 神经网络、自组织特征映射神经网络、RBF 神经网络等几个应用比较广泛的神经网络方面, 对于概率型神经网络在文本分类中的应用研究涉及较少. 而概率神经网络是径向基网络的一种重要变形, 在模式识别问题上已经有比较多的应用, 取得较好的效果^[2].

本文利用概率型神经网络的特点, 结合 LVQ 学习算法, 探讨该方法在文本分类的研究, 并取得了非常不错的效果.

^① 基金项目: 国家自然科学基金(61175068)

收稿时间: 2012-02-06; 收到修改稿时间: 2012-03-05

2 概率型神经网络拓扑结构及学习算法

2.1 概率型神经网络的拓扑结构

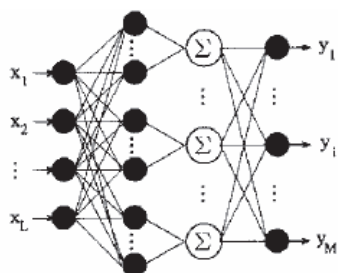


图1 概率型神经网络的基本拓扑结构

概率神经网络是神经网络的一个变形学习算法,通常由输入层、模式层、求和层、输出层四层组成^[3]。第一层为输入层,这一层对输入不做任何处理,只是简单的将输入向量 $X = [x_1, x_2, \dots, x_L]$ 传递给网络;中间两层为隐层,第一隐层为模式层,通过连接权值与输入层连接,模式层在接收输入向量 $X = [x_1, x_2, \dots, x_L]$ 后,第 i 类模式的第 j 个模式神经元所确定的输入输出关系由下式来定义:

$$f_{ji}(X) = \frac{1}{(2\pi)^{s/2} \sigma^s} \frac{1}{m} \sum_{i=1}^m \exp \left[-\frac{(X - X_{ji})(X - X_{ji})^T}{\sigma^2} \right] \quad (1)$$

其中 $X = [x_1, x_2, \dots, x_L]$ 为输入样本向量, s 为样本向量的变量个数, X_{ji} 为属于 j 类的第 i 个训练样本向量,在概率神经网络中作为权值, m 表示属于 j 类的训练样本向量个数, σ 为光滑系数。

第二隐层为求和层,有选择地对第一隐层的输出进行求和,得到:

$$g_{jN_i}(X) = \sum_{i=1}^m f_{ji}(X) \quad (2)$$

最后一层为输出层,是一个竞争层,完成输入数据分类,它的处理分为两步,首先计算每个竞争单元输入的加权求和 $S_j = \sum_i w_{ji} u_i$ (其中, w_{ji} 表示第 j 个神经元的第 i 个连接权值, u_i 表示和 w_{ji} 相关联的输入,即 $g_{jN_i}(X)$) 然后比较各个竞争单元的加权总和的大小,具有最高输入总和的那个单元被确定为胜者,输出为 1,其它的为 0,即:

$$y_j = \begin{cases} 1, S_j = \max\{S_k, k=1,2,3 \dots M\} \\ 0, \text{其他} \end{cases} \quad (3)$$

y_j 表示第 j 个神经元的输出状态,根据 $y_j=1$ 的输出

神经元的序列号判别出输入的模式类。

2.2 基于 LVQ 算法的概率型神经网络学习算法

根据前面的描述我们知道概率型神经网络的输出层是一个竞争层网络,它是一种无监督学习方法的网络,当输入到该层的两个向量相接近时,竞争层网络会把它们归结为一类^[4]。因而不利于在文本分类这种具有数据交叠模型中应用,而 LVQ 学习算法是一种针对竞争层网络的有监督训练方法,通过对网络进行监督学习,使相应的模式向量相互靠拢,远离其他的模式输入向量,从而完成对输入的正确分类。对概率型神经网络的 LVQ 学习就是针对概率型神经网络的竞争层进行 LVQ 学习。LVQ 学习算法经过改进后分两阶段完成,具体算法按照下述步骤实现^[5]:

第一阶段的 LVQ 学习(LVQ1)

Step1: 给定数据集 $X = [X_1, X_2, \dots, X_L]$, 对网络进行初始化,设竞争层学习速率 $a_k, k=1,2,\dots,K$ 、网络权矢量 $W_i(0), i=1,2,\dots,P$ (P 表示第二隐层和输出层的连接权矢量,及迭代步骤 $K=0$;

Step2: 有序的从训练集中取出一个输入量 $X^m = [x_1^m, x_2^m, \dots, x_L^m]$, 计算竞争层输入向量的加权总和 $S_j = \sum_i w_{ji} u_i$, 求胜出的编号 $\sigma(X) = \arg \max\{S_j\}$; 根据训练结果调整学习速率及网络的权值,学习速率调整公式为(4):

$$a_k = a_0(1 - k/K) \quad (4)$$

权值调整按式(5)或(6)调整:

当 $X^m \in y_j$ 时:

$$W_j(k+1) = \begin{cases} W_j(k) + a_0(u(k) - W_j(k)), S_j = \max\{S_j\} \\ W_j(k), \text{其他} \end{cases} \quad (5)$$

当 $X^m \notin y_j$ 时:

$$W_j(k+1) = \begin{cases} W_j(k) - a_0(u(k) - W_j(k)), S_j = \max\{S_j\} \\ W_j(k), \text{其他} \end{cases} \quad (6)$$

Step3: 判断 i 和 K 的大小,如果 $k > K$, 转 step2, 如果 $k < K$, 计算结果并输出计算结果: $W_j(k)$;

第二阶段的 LVQ 学习(LVQ2), 它是在 LVQ1 基础上对网络进行训练,以第一阶段学习结果作为第二阶段的初始权矢量,它是在最接近输入向量的两个竞争神经元之间进行权值的调整。其中一个对应正确类别,称为第一优胜者;另一个为错误类别,称为第二

优胜者。

Step4: 对于输入样本集合 $U_i, (i=1,2,\dots,N)$, 按下式计算该类的中心样本 X_i^{m*} :

$$\overline{X_i^m} = \frac{1}{n} \sum_{j=1}^n X_{ij}, X_i^m \in U_i \quad (5)$$

Step5: 记各中心样本 X_i^{m*} 对应的获胜竞争神经单元的为 y_j , 设 k 时刻从训练集中取输入向量 $\vec{x}(k)$ 输入网络, 设网络对输入 $\vec{x}(k)$ 的判断类别为 $y_j(k)$ 为第一优胜者, 第二优胜者为 $y_i(k)$. 在各个竞争神经单元之间它们的欧几里德距离有如下关系:

$$\|\vec{u}(k) - W_{y_j(k)}\| < \|\vec{u}(k) - W_{y_i(k)}\| < \|\vec{u}(k) - W_{y_t(k)}\|, (t \neq j \neq i) \quad (7)$$

Step6: 定义一个窗口值 ε , 当时 $\min\{\frac{d_i}{d_j}, \frac{d_j}{d_i}\} > \varepsilon$,

其中, $d_j = \|\vec{u}(k) - W_{y_j(k)}\|$, $d_i = \|\vec{u}(k) - W_{y_i(k)}\|$, 按下式对对权值进行调整:

$$\begin{cases} W_{y_j}(k+1) = W_j(k) + a_k(\vec{u}(k) - W_j(k)); \\ W_{y_i}(k+1) = W_i(k) - a_k(\vec{u}(k) - W_i(k)); \\ W_{y_t}(k+1) = W_t(k); \end{cases} \quad (8)$$

Step7: 判断 k 和 K 的大小, if $k < K$, 按式 8 调整学习速率, 转 step5, if $k > K$, 结束, 保存权值 W , 训练结束。

3 文本特征项构造及特征值提取

神经网络的输入是向量形式的, 因此, 分类过程首先需对文本进行预处理, 得到能够表示文本特征的有效特征项集, 进而提取出文本的特征向量。一个有效的特征项集, 应该具备两点: 1) 完备性: 同类文本特征项集能够完全体现该类文本的特征; 2) 可区分性: 不同类别的文本之间的特征项集之间具有相互的独立性^[6]。近年来应用最多且较好的特征词提取方法是 TFIDF 方法, 这种方法是建立在特征相互独立的假设基础上, 能够有效的降低计算时间^[7]。TF 是指词语在文本中出现的频率, 某个词语在一个文本出现次数越多, 则认为这个词语对这个文本的表示贡献越大。IDF 为倒排文本频率, 指的是词语在文本集合中的分布情况。它表示全部文本中含有某个词语的文本越少, 则该词语表示文本时越重要。IDF 降低了那些在大多数文本中都出现的词语的重要性, 增强了那些只在少数文本中

出现词语的重要性。基于 TFIDF 的文本有效特征项集可按下述方法实现:

已有 N 篇具有标记的训练文本, 特征项集构造为如下所述:

Step1: 对 N 篇文本进行处理抽取特征项(将同义词、近义词看做同一特征项), 构造特征项集: $U_1, U_2, \dots, U_N$, 然后对各个特征项集进行筛选, 去除低频词和超高频率词。即根据给定的上限和下限阈值 $\lambda_{\text{上}}$, $\lambda_{\text{下}}$, 滤除各个文本中低于 $\lambda_{\text{下}}$ 和高于 $\lambda_{\text{上}}$ 的特征项, 这样可以降低文本特征的维数而又使特征项满足上述条件。这样以后得到每篇文本的特征项集合: $U_1, U_2, \dots, U_N$;

$$U_1 = \{T_1^1, T_2^1, \dots, T_n^1\}; U_2 = \{T_1^2, T_2^2, \dots, T_n^2\}; \dots; U_N = \{T_1^N, T_2^N, \dots, T_n^N\}$$

Step2: 计算 N 个特征项集的并集: $U \cup U_2 \cup \dots \cup U_N = \{T_1, T_2, \dots, T_n\}$, 具体算法如下:

令 $C = C_1$, 对 $\forall T_i^j \in C_j (i=1,2,3,\dots,N; j=1,2,3,\dots,P)$, 若 $(T_i^j \notin C_j)$ 且 $(T_i^j \notin C_j)$ 则 $C = C \cup \{T_i^j\}$ 。

Step3: 以特征项集为 $\{T_1, T_2, \dots, T_n\}$ 为论域, 分别对 N 篇文本, 计算 $\{T_1, T_2, \dots, T_n\}$ 特征项在该文本中出现的频数, 根据每个特征项在某一文档中出现的频数构造该篇文档的特征向量。

Step4: 按下式构造 N 篇文本的特征向量

$$\{f_T(T_{N1}), f_T(T_{N2}), \dots, f_T(T_{Nn})\}, (i=1,2,\dots,N) \\ f_T(T_{Ni}) = m \lg\left(\frac{N}{N_i} + 0.5\right), (i=1,2,\dots,N; j=1,2,\dots,n) \quad (9)$$

其中, m 表示文本特征项 T_i 在文档 N 中出现的频数, N 表示全部训练文本的文本数, N_i 表示含有特征项 T_i 的文本数目。

Step5: 对以上特征向量进行归一化, 可得 N 篇文本的特征向量 $\tilde{T} = \{T_{N1}, T_{N2}, \dots, T_{Nn}\}$ 。

4 实验结果及实验结果分析

实验数据采用了从中国期刊网上下载的 1800 篇文本, 共有九个类, 涵盖: 外语、计算机、军事、法律、教育、体育、娱乐、政治、科技。将材料分为训练集和测试集两部分, 其中训练文本 900 篇, 测试文本 900 篇, 每种类别的训练和测试文本各 100 篇。首先采用上面所述的 TFIDF 方法对文本预处理, 提取特征词,

选取特征向量维数为 256 维, 构造得到两个的特征向量矩阵. 然后根据实验要求设计概率神经网络, 网络分为四层, 第一层为输入层, 输入层神经元的数目的确定和输入的文本特征向量维数大小有关, 按照预处理的文本特征向量的大小将输入层的神经元的数目设为 256; 第二层为隐含层, 第三层为求和层, 根据实验过程的调整分别设为 200, 85, 隐含层的传输函数是一个径向基函数, 阈值 $b = [-\log.5]^{1/2} / spread$, $spread$ 为径向基函数的扩展系数, 本实验确定为 0.7, 学习函数采用默认设置, 求和层将由对应与训练样本中同一类的模式层传来的输入进行累加; 第四层分类层, 这是一个竞争层, 无阈值, 神经元的传输函数是一个竞争型传输函数, 参照待分类的文本类数目, 该层的神经元数目设定为 9, 9 个神经元中只有一个神经元的输出等于 1, 此分量设为 y_i , 其它神经元的输出都为 0. 网络结构的设计完成后, 对网络进行训练, 文本特征向量为 900×256 的向量, 目标向量为 900×9 的向量, 训练步骤分为三步, 第一步进行无监督学习算法训练, 第二步调整学习竞争层的学习函数, 将学习函数设为 LVQ1, 重新对网络进行训练, LVQ1 为 LVQ 第一种学习算法, 权值调整算法对应式(5)或式(6); 第三步再调整竞争层的学习函数, 将学习函数设为 LVQ2, 再对网络进行训练, LVQ2 是改进的 LVQ 学习算法, 它是在 LVQ1 训练的基础上对网络进行训练, 调整权值的算法对应式(8), 至此网络训练完成, 将训练好的网络进行实验^[8].

考虑评价和测试文本的自动分类算法的性能常用 IR 中使用的查全率(recall)和查准率(precision)^[9]:

1) 查全率(recall) = $\frac{T_n}{N}$, T_n 为通过分类算法被正确分类为 U_i 类的文本数目; N 为分类之前属于 U_i 类的文本的数目.

2) 查准率(precision) = $\frac{T_n}{U_n}$, T_n 为通过分类算法被正确分类为 U_i 类的文本数目; U_n 为通过分类算法被分为 U_i 类的文本数目.

表 1 概率型神经网络(无监督学习)分类结果

类别	概率神经网络 (无监督学习)				
	实际文本数	正确分类的文本数	分到该类的文本数	查全率 (%)	查准率 (%)
外语	100	96	99	96	96.97

计算机	100	91	98	91	92.86
军事	100	90	104	90	86.54
法律	100	95	102	95	93.14
教育	100	84	90	84	93.33
体育	100	95	100	95	95.00
娱乐	100	91	94	91	96.81
政治	100	92	105	92	87.62
科技	100	97	108	97	89.82
平均的查全率 (%)	92.3333	平均查准率 (%)	92.4531	训练时间	0.625s

表 2 概率型神经网络(LVQ 学习算法)分类结果

类别	概率神经网络 (LVQ 学习算法)				
	实际文本数	正确分类的文本数	分到该类的文本数	查全率 (%)	查准率 (%)
外语	100	97	99	97	97.98
计算机	100	93	99	93	93.93
军事	100	91	101	91	90.10
法律	100	94	104	94	94.00
教育	100	89	94	89	94.68
体育	100	95	100	95	95.00
娱乐	100	91	95	91	95.79
政治	100	93	103	93	90.29
科技	100	98	109	98	89.91
平均查全率 (%)	93.4444	平均查准率 (%)	93.5209	训练时间	0.0790s

实验中对无监督学习和 LVQ 学习算法的概率型神经网络在文本的分类性能进行了研究, 首先训练已经建立好的网络, 网络训练后将网络用于实验, 从实验结果分析可知概率型神经网络对文本具有较好的分类能力. 在没有经过 LVQ 学习算法前的网络对文本的分类正确率在 86.54%~96.97%之间, 平均查准率为 92.4531%, 平均查全率为 92.3333%. 经过 LVQ 学习算法学习后, 网络对文本的分类正确率在 89.91%~97.98%之间, 平均查准率为 93.5209%, 平均查全率为 93.4444%, 网络的平均查准率, 查全率都提高了 1%以上, 说明经过 LVQ 学习算法学习后, 网络对文本的分类性能得到了优化, 这是因为经过 LVQ 学习算法学习后, 网络较好的解决了聚类样本的测试数据交叠的问题. 实验结果还表明经过 LVQ 学习算法学习的网络的学习效率得到较好的改善, 这使的网络的收

敛速度得到了大大改善,从而缩短了网络训练的时间,训练时间约缩短了 8 倍.从表 1 和表 2 中,还知道被正确分类最少的是教育类,这说明在所选取的文本中教育类提取的特征信息和其它类的特征信息有较多的相关性,不好被区分出来.实验中被分到科技类的最多,这说明在所选取的文本中其它类提取的特征信息和科技类特征信息具有一定的相似性,因而使得有些自身特征信息不明确的文本被分到科技类.因而对这些文本在进行特征信息提取的时候需要进行适当的处理,使得各类之间的特征信息具有较好的可区分性,能够输入到网络中被正确分类.

5 结语

本文探讨了基于概率型神经网络和 LVQ 学习算法相结合的文本分类方法,实验结果表明该方法具有较好的文本分类效果,经过 LVQ 学习后的网络的训练效率、查全率和查准率都得到了改善.进一步的研究将在适合于概率神经网络的文本特征提取、文本权重计算、分类模型的完善方面开展深入的研究与改进.

(上接第 60 页)

结合实现;其它锁的帮助类,通过简单的锁成员对象封装类便可实现.

4 结语

本文设计与实现的通用多线程库,可以跨 Linux 和 Windows 平台,并且通过简单的本地线程库封装,也可跨更多的软硬件平台,为采集、处理、解释一体的地震勘探软件的开发提供了简易、高效的多核并行开发库.并且通过进一步的搜集和监控 CPU 等硬件信息,能简易的实现线程池,从而使一个功能性的组件,变成兼顾功能和性能双重性的组件,为云计算平台上的一体化勘探软件的部署提供高效的并行运行条件.

参考文献

- 1 李智,孙立铸.一种基于进化 BP 神经网络的文本分类算法.佳木斯大学学报,2009,27(2):165-167.
- 2 蔡继亮,叶微.基于概率神经网络的空袭目标识别.火力与指挥控制,2010,35(12):25-27.
- 3 蔡曲林,刘普寅.一种新的概率神经网络有监督学习算法.模糊系统与数学,2006,20(6):83-87.
- 4 周开利,庄耀红.神经网络模型及其 matlab 仿真程序设计.北京:清华大学出版社,2005.
- 5 张义忠,赵明生,梁久祯.基于自组织特征映射的网页分类研究.测试技术学报,2002,16:1433-1438.
- 6 隋长富.过程神经网络在网页中的自动分类中的应用研究.人工智能及识别技术,2008,2(12):533-536.
- 7 徐建民,刘清江,付婷婷,等.基于量化同义词关系的改进特征提取方法.河北大学学报,2010,30(01):97-101.
- 8 张德丰.matlab 神经网络应用设计.北京:机械工业出版社,2009.50-180.
- 9 陈显毅,朱倩.文本挖掘原理.北京:科学出版社,2010.14-45.

参考文献

- 1 赵改善.地球物理软件技术发展趋势与战略研究.勘探地球物理进展,2010,2.
- 2 汤子瀛.计算机操作系统(修订版).西安:西安电子科技大学出版社,2001.
- 3 杨开杰.线程池的多线程并发控制技术研究.计算机应用与软件,2010,1.
- 4 王晶.多核多线程结构线程调度策略研究.计算机科学,2007,9.
- 5 Butenhof DR. Programming with POSIX threads. Addison-Wesley, 1997.
- 6 Beveridge J. Multithreading Application in Win32. Addison-Wesley, 1996 年 10 月
- 7 鲁建壮.面向线程级前瞻的线程划分方法浅析.计算机科学,2006,5.