◎博士论坛◎

面向文档分类的 LDE 和简化 SVM 方法研究

王自强[1],钱 旭[1],孔 敏[2]
WANG Zi-qiang[1],QIAN Xu[1],KONG Min[2]

1.中国矿业大学(北京)机电与信息工程学院,北京 100083
2.山东省曲阜市职业中等专业学校,山东 曲阜 273100
1.College of Mechanical Electronic and Info. Eng.,China University of Mining and Technology(Beijing),Beijing 100083,China
2.Qufu Vocational School of Shandong Province,Qufu,Shandong 273100,China
E-mail:wzqbox@yahoo.cn

WANG Zi-qiang,QIAN Xu,KONG Min.LDE and simplified SVM method for document classification.Computer Engineering and Applications,2009,45(22):1-3.

Abstract: To rapidly and accurately perform document classification task,an efficient document classification algorithm is proposed by using Local Discriminant Embedding(LDE) and Simplified Support Vector Machine(SSVM) in this paper.The essential idea of the algorithm is as follows.The high dimensional document data are first projected into the lower dimensional feature space with LDE algorithm,then the SSVM algorithm is applied to classify documents in the lower dimensional feature space.Experimental results show that the proposed algorithm is of higher classification accuracy and faster running speed.
Key words: document classification;local discriminant embedding;simplified support vector machine;data mining

摘 要:为了快速准确地对文档进行分类,提出了一种基于局部鉴别嵌入 LDE 和简化 SVM 的高效文档分类算法。该算法首先利用 LDE 算法把高维文档数据投影到低维特征空间,然后在低维特征空间利用精简 SVM 进行分类。实验结果表明该算法具有分类准确率高和运行速度快的优点。
关键词:文档分类;局部鉴别嵌入;简化支持向量机;数据挖掘
DOI:10.3778/j.issn.1002-8331.2009.22.001 **文章编号**:1002-8331(2009)22-0001-03 **文献标识码**:A **中图分类号**:TP181

1 引言

文档分类的主要任务是在预先给定的类别标记集合中,根据文档内容判定它的类别。由于文档分类在搜索引擎、垃圾邮件过滤、协同过滤、个性化数字图书馆中具有广泛的应用[1],因此该研究日益成为机器学习、数据挖掘和模式识别等众多领域的热点研究问题之一。

传统的文档分类算法在处理高维大规模数据集时,存在着测试运行时间过长,分类准确率不高的不足。考虑到高维文档空间引起的"维数灾难"问题,应当首先把高维文档空间投影到低维特征子空间,然后在降维后的低维特征空间利用高效分类器进行分类判别。事实上,高维文档数据空间的内在维数是很低的,通过维数降维,可以有效地发现高维数据空间内在的结构特征、避免"维数灾难"、提高后继分类器的性能和计算效率[2]。基于上述考虑,提出了基于局部鉴别嵌入 LDE(Local Discriminant Embedding)[3]和简化 SVM(Simplified Support Vector Machine)的文档分类算法。LDE 是一种最近提出的既能保持局部几何结构又显式地考虑数据类别标记的面向分类问题的流形学习算法,它能够有效地对高维文档空间进行维数降维。精简 SVM 是一种优化提高的 SVM,它有效地克服了传统 SVM 分类器存在的测试时间过长及分类准确率不高的不足。为了充分发挥两者的优势,提出:首先利用 LDE 算法把高维文档投影到低维特征空间,然后在降维后的低维特征空间利用精简 SVM 进行分类。实验结果表明该方法是可行的,且具有分类准确率高、运行速度快的优点。

2 技术基础

2.1 局部鉴别嵌入算法 LDE

局部鉴别嵌入 LDE[3]是最近提出的一种用于数据降维的流形学习算法。由于它能够在计算数据映射的过程中同时考虑局部数据几何结构和类别标记信息, 因此高维数据经 LDE 映射到低维特征空间后,来自相同类别标记的数据点仍然能够保持它们内在的邻居关系,而来自不同类别标记的点被远远地分离开。这恰好体现了分类算法的主要目的:保持相同类别内的紧凑性和不同类别间的分离性。因此 LDE 算法是一种面向分类

基金项目:教育部科学技术研究重点资助项目(The Key Science Foundation of Ministry of Education of China under Grant No.107021)。
作者简介:王自强(1973-),男,博士研究生,主要研究方向:机器学习与模式识别;钱旭(1962-),男,博士,教授,博士生导师,主要研究方向:机器学习与信息融合;孔敏(1976-),女,讲师,主要研究方向:数据挖掘。
收稿日期:2009-04-15 **修回日期**:2009-05-18

任务的流形学习算法。所以,在文档分类系统中采用 LDE 算法对高维文档数据进行降维,可以大大提高后继分类器的性能和计算效率。下面给出 LDE 的算法描述:

步骤 1 构建邻接图。设 G 和 G'表示一个具有 n 个顶点(数据点)的无向图,且每个顶点 $\boldsymbol{x}_i$ 的类别标记为 l_i。构造图 G 的方法如下:如果顶点 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 具有相同的类别标记(即,$l_i=l_j$),且 $\boldsymbol{x}_j$ 是 $\boldsymbol{x}_i$ 的 k 个最近邻居之一,则在顶点 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 构建一条边。图 G'的构造方法如下:如果顶点 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 具有不同的类别标记(即,$l_i\neq l_j$),且 $\boldsymbol{x}_j$ 是 $\boldsymbol{x}_i$ 的 k'个最近邻居之一,则在则在顶点 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 构建一条边。

步骤 2 计算图中边的权重。在图 G 中,如果顶点 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 有一条边,则该边的权重 w_{ij}=1,否则,w_{ij}=0。图 G'中边的权重确定方法和图 G 相似:如果图 G'中顶点 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 有一条边,则该边的权重 w_{ij}'=1,否则,w_{ij}'=0。

步骤 3 计算局部鉴别嵌入。计算如下特征方程中前 p 个最大特征值所对应的特征向量:

$$\boldsymbol{X}(\boldsymbol{D}'-\boldsymbol{W}')\boldsymbol{X}^{\mathrm{T}}a=\lambda\boldsymbol{X}(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{X}^{\mathrm{T}}a \tag{1}$$

其中,$\boldsymbol{W}$ 和 $\boldsymbol{W}'$分别表示图 G 和 G'中的边权重矩阵,$\boldsymbol{D}$ 和 $\boldsymbol{D}'$表示对角矩阵,其定义为:

$$\boldsymbol{D}=\sum_j w_{ij},\boldsymbol{D}'=\sum_j w_{ij}' \tag{2}$$

设 $\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_p$ 是式(1)的特征向量,并按照其对应特征值由大到小的顺序排列,即:$\lambda_1>\lambda_2>\cdots>\lambda_p$。则由高维数据到低维数据的局部鉴别嵌入表示如下:

$$\boldsymbol{x}_i\rightarrow\boldsymbol{y}_i=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x}_i \tag{3}$$

其中 $\boldsymbol{A}=(\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_p)$。

从以上 LDE 的算法过程可以得出如下结论:(1)由于 LDE 算法在寻找由高维数据空间向低维数据空间映射的过程中,同时考虑了局部数据结构和类别标记线,因而能够使得映射后的数据更加适合后继分类器分类任务的要求。(2)由于 LDE 算法采用了线性映射的方法,从而很好地解决了传统流形学习算法无法直接得到新测试数据的低维嵌入表示问题。(3)由于低维嵌入向量是通过求解稀疏矩阵的特征向量得到的,因而具有计算量小,运行速度快的优点。

2.2　简化 SVM

为了快速准确地对经过 LDE 降维后的高维文档数据进行分类,采用了具有良好泛化能力的支持向量机(SVM)[4]作为分类器。其算法实现过程如下:

设训练样本集$\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^{n}$,样本 $\boldsymbol{x}_i\in R^N$,$\boldsymbol{y}_i\in\{-1,+1\}$为类标记。SVM 通过求解下式找到一个具有最大间隔的超平面:

$$\min_{\boldsymbol{w},b,\xi}\ \frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}+C\sum_{i=1}^{n}\xi_i \tag{4}$$

$$\text{s.t. } y_i[(\boldsymbol{w}\cdot\varphi(\boldsymbol{x})+b)+\xi_i-1]\geqslant 0$$

其中,C 表示用于控制误差的惩罚常数,ξ_i 为非负松弛变量,$\varphi(\boldsymbol{x})$表示从原始低维空间到高维特征空间的映射函数。

利用拉格朗日乘子法,可以把式(4)转化为如下对偶形式:

$$\max\ W(\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i,\boldsymbol{x}_j) \tag{5}$$

$$\text{s.t. }\sum_{i=1}^{n}\alpha_i y_i=0,0\leqslant\alpha_i\leqslant 1,i=1,\cdots,n$$

其中,$K(\boldsymbol{x}_i,\boldsymbol{x}_j)$表示核函数,用来计算特征空间中的函数内积:$K(\boldsymbol{x}_i,\boldsymbol{x}_j)=(\varphi(\boldsymbol{x}_i)\cdot\varphi(\boldsymbol{x}_j))$,而无需知道映射函数 $\varphi(\boldsymbol{x})$的具体形式。

于是,SVM 分类器的最优判别函数为:

$$f(\boldsymbol{x})=\mathrm{sgn}\left(\sum_{i=1}^{n_{SV}}\alpha_i y_i K(\boldsymbol{x}_i,\boldsymbol{x})+b\right) \tag{6}$$

其中,n_{SV} 表示支持向量的数目。

由式(6)可知:SVM 分类器的测试速度主要取决于支持向量的数目。对于大规模高维数据(如文本数据、图像数据、基因数据等)而言,其支持向量的数目是相当大的,因而导致 SVM 的测试速度很慢。为了克服 SVM 的这一不足,在简化集算法[5-7]的基础上,采用了如下高效的简化 SVM 分类器算法,该算法不仅有效地克服了传统 SVM 分类器测试速度慢的缺点,而且具有很好的分类准确率。

根据再生核空间理论,特征空间 $\boldsymbol{F}$ 中的任一向量 $\boldsymbol{\psi}\in\boldsymbol{F}$ 都是由空间 $\boldsymbol{F}$ 中的所有样本线性组合而成的。于是可得:

$$\boldsymbol{\psi}=\sum_{i=1}^{n_s}\alpha_i\varphi(x_i) \tag{7}$$

其中,$\alpha_i\in R$,$\varphi(x_i)\in\boldsymbol{F}$ 表示原始输入数据 $\boldsymbol{x}_i\in R^N$ 到特征空间 $\boldsymbol{F}$ 的映射。为了降低运算复杂度,简化集算法通过寻找一组简化向量的线性组合来近似来 $\boldsymbol{\psi}$,即:

$$\boldsymbol{\psi}'=\sum_{i=1}^{n_z}\beta_i\varphi(z_i) \tag{8}$$

其中,$n_z<<n_s$,$\beta_i\in R$,$\varphi(z_i)\in\boldsymbol{F}$ 表示简化向量 $\boldsymbol{z}_i\in R^N$ 到特征空间 $\boldsymbol{F}$ 的映射。于是,上述最小化近似误差可以通过求解如下最小化目标函数获得:

$$\min_{z_i,\beta_i}\|\boldsymbol{\psi}-\boldsymbol{\psi}'\|^2=\min_{z_i,\beta_i}\left\{\sum_{i,j=1}^{n_s}\alpha_i\alpha_j K(\boldsymbol{x}_i,\boldsymbol{x}_j)+\sum_{i,j=1}^{n_z}\beta_i\beta_j K(\boldsymbol{x}_i,\boldsymbol{x}_j)-2\sum_{i=1}^{n_s}\sum_{j=1}^{n_z}\alpha_i\beta_j K(\boldsymbol{x}_i,\boldsymbol{z}_j)\right\} \tag{9}$$

该方法的关键思想在于:上述优化目标函数的求解只需通过计算核函数 $K(,)$即可,而无需知道映射函数 $\varphi()$的具体形式。采用文献[5]提出的迭代算法求解式(9)的目标函数来获得系数 β_i 和向量 $\boldsymbol{z}_i$。于是,给定简化向量数目 n_z,将式(8)代入式(6)可得到简化 SVM 分类器的最优判别函数:

$$f(x)=\mathrm{sgn}\left(\sum_{i=1}^{n_z}\beta_i K(\boldsymbol{z}_i,\boldsymbol{x})+b\right) \tag{10}$$

与式(6)相比,由于在式(10)中的待测试向量数目 $n_z<<n_{SV}$,因而简化 SVM 分类器大大地提高了计算效率,后面的实验结果有力地证明了这一点。

3　基于 LDE 和简化 SVM 的文档分类算法

设给定文档样本集 $\boldsymbol{X}=\{x_1,x_2,\cdots,x_n\}$,其中 $\boldsymbol{x}_i\in R^N$。采用向量空间模型(VSM)[8]中常用的项权重向量来表示每个文档 $\boldsymbol{x}_i$,并将其长度归一化。基于 LDE 和简化 SVM 的文本分类算法的主要思想为:首先利用局部鉴别嵌入 LDE 把高维文档映射到低维特征空间,然后利用简化 SVM 分类器对降维的高维文档数据进行分类判决。算法的具体实现过程如下:

步骤 1 利用向量空间模型 VSM 中的 TF-IDF 项权重向量来表示每个文档 $\boldsymbol{x}_i$。

步骤 2 构建邻接图。设 G 和 G'表示一个具有 n 个顶点的无向图,其中顶点 i 对应于文档 $\boldsymbol{x}_i$,并设顶点 i 的类别标记为 l_i。构造图 G 的方法如下:如果顶点 i 和 j 具有相同的类别标记(即,$l_i=l_j$),且顶点 j 是顶点 i 的 k 个最近邻居之一,则在顶点 j

和 i 构建一条边。图 G'的构造方法和 G 的构造方法类似:如果顶点 i 和 j 具有不同的类别标记(即,$l_i \neq l_j$),且顶点 j 是 i 的 k' 个最近邻居之一,则在则在顶点 i 和 j 构建一条边。

步骤 3　文档向量预处理。为了确保后面的 LDE 优化目标(即:式(1))不包含平凡解,首先通过去掉对应于零特征值的分量,将每个文档 $\boldsymbol{x}_i$ 投影到 PCA 子空间来消除平凡解。设 $\boldsymbol{W}_{PCA}$ 表示 PCA 的变换矩阵,则经过 PCA 投影后,文档 $\boldsymbol{x}_i$ 变为$\tilde{\boldsymbol{x}}_i$。

步骤 4　计算图中边的权重。在图 G 中,如果顶点 i 和 j 相连接,则该边的权重 $w_{ij}=\tilde{\boldsymbol{x}}_i^{\mathrm{T}}\tilde{\boldsymbol{x}}_j$,否则,$w_{ij}=0$。图 G'中边的权重确定方法和图 G 相似:如果图 G'中顶点 i 和 j 相连接,则该边的权重 $w_{ij}'=\tilde{\boldsymbol{x}}_i^{\mathrm{T}}\tilde{\boldsymbol{x}}_j$,否则,$w_{ij}'=0$。

步骤 5　LDE 映射。计算如下特征方程中前 p 个最大特征值所对应的特征向量:

$$\tilde{\boldsymbol{X}}(\boldsymbol{D}'-\boldsymbol{W}')\tilde{\boldsymbol{X}}^{\mathrm{T}}a=\lambda\tilde{\boldsymbol{X}}(\boldsymbol{D}-\boldsymbol{W})\tilde{\boldsymbol{X}}^{\mathrm{T}}a \tag{11}$$

其中,$\tilde{\boldsymbol{X}}=\{\tilde{x}_1,\tilde{x}_2,\cdots,\tilde{x}_n\}$,$\boldsymbol{W}$ 和 $\boldsymbol{W}'$分别表示图 G 和 G'中的边权重矩阵,$\boldsymbol{D}$ 和 $\boldsymbol{D}'$表示对角矩阵,其定义如下:

$$\boldsymbol{D}=\sum_j w_{ij},\boldsymbol{D}'=\sum_j w_{ij}' \tag{12}$$

设 $\boldsymbol{W}_{LDE}=[\boldsymbol{a}_1,\cdots,\boldsymbol{a}_p]$表示依据特征值 $\lambda_1>\lambda_2>\cdots>\lambda_p$ 的排列次序依次得到的式(11)的解,于是由高维文档到低维特征空间的映射可表示为:

$$\boldsymbol{x}_i\rightarrow\boldsymbol{y}_i=\boldsymbol{W}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i \tag{13}$$

$$\boldsymbol{W}=\boldsymbol{W}_{PCA}\boldsymbol{W}_{LDE} \tag{14}$$

步骤 6　利用简化 SVM 分类器进行分类判决。将原始高维文档集 $\boldsymbol{X}=\{x_1,x_2,\cdots,x_n\}$经 LDE 投影后变成的低维文档特征集 $\boldsymbol{Y}=\{y_1,y_2,\cdots,y_n\}$输入到如下简化 SVM 分类器的判决函数来确定测试文档所属的类别:

$$f(\boldsymbol{y}_i)=\mathrm{sgn}\left(\sum_{j=1}^{n_z}\beta_j K(\boldsymbol{z}_j,\boldsymbol{y}_i)+b\right) \tag{15}$$

4　实验结果

为了测试所提出的基于 LDE 和简化 SVM 的文档分类算法(以下简称:LDE-SSVM)分类性能,将 LDE-SSVM 算法与其他基于维数降维和传统 SVM 的文档分类算法进行了比较:(1)直接在原始文档空间利用传统 SVM 进行文档分类,以下简称:SVM;(2)首先采用潜在语义索引(LSI)[9]对文档进行降维,然后利用传统 SVM 进行分类的算法,以下简称:LSI-SVM 算法;(3)首先采用线性鉴别分析(LDA)[10]对文档进行降维,然后利用传统 SVM 进行文档分类的算法,以下简称:LDA-SVM 算法。简化 SVM 和传统 SVM 中的核函数采用高斯核函数,模型参数采用 10 次交叉验证法来确定。由于 SVM 是用于二分类问题的,而文档分类是多分类问题,因此采用常用的一对剩余法(One-vs-Rest)[11]进行多类分类。

测试数据采用了文档分类领域两个著名的测试集Reuters-21578 和 WebKB。对于 Reuters-21578 数据集[12],选择文档最多的 10 个类别进行实验,其中包括了 7 194 个训练文档和 2 788 个测试文档。对于 WebKB 数据集[13],遵循先前研究的常用方法,也选用了其中的 4 个类别(course、faculty、project 和 student)进行实验。其中每个类别中的一半数据作为训练集,另一半作为测试集。文档分类算法的性能指标采用常用的准确率(Accuracy)、$F1$ 值和均衡点(Break-Even Point)值进行比较,这三个评价指标的值越大,说明分类算法的性能越好。它们的定义如下:

$$\text{准确率}=\frac{\text{正确分类的测试文档数}}{\text{测试文档总数}} \tag{16}$$

$$F1=\frac{2\times\text{查准率}\times\text{查全率}}{\text{查准率}+\text{查全率}} \tag{17}$$

$$\text{均衡点}=(\text{查准率}=\text{查全率})\text{时所对应的点} \tag{18}$$

$$\text{查准率}=\frac{\text{正确分为某类的文档数}}{\text{测试集中分为该类别的文档总数}} \tag{19}$$

$$\text{查全率}=\frac{\text{正确分为某类的文档数}}{\text{测试集中属于该类别的文档总数}} \tag{20}$$

表 1 和表 2 分别给出了 4 种文档分类算法在测试数据集 Reuters-21578 和 WebKB 上的实验结果。从中可以看出:提出的 LDE-SSVM 分类算法的准确率、$F1$ 值和均衡点值都普遍高于常用的 SVM、LSI-SVM 和 LDA-SVM 分类算法。原因在于:传统的 SVM 分类算法是在原始高维文档空间直接进行的,而没有采用任何维数降维方法,因而存在着训练和测试速度较慢、分类效果不太理想的问题。LSI-SVM 算法虽然采用了 LSI 算法对高维文档空间进行降维,但是 LSI 是一种无监督的维数降维方法,旨在揭示高维文档的表示(representative)特征而不是用于分类目的的鉴别(discriminative)特征,因此 LSI 在降维的过程中并不能真正将不同类别的文档区分开。LDA 是一种能够有效利用文档类别信息的有监督维数降维方法,虽然它在一定程度上能将不同类别的文档区分开,但是其不足之处在于存在"奇异值"问题,因此 LDA-SVM 算法的分类性能有待进一步提高。LDE 算法在降维的过程中能够保持相同类别内的紧凑性和不同类别间的分离性,因此高维文档经 LDE 降维后大大提高了后继分类器 SSVM 的性能和计算效率。另外 SSVM 分类器算法不仅有效地克服了传统 SVM 分类器测试速度慢的缺点,而且具有较好的分类准确率,从而保证了 LDE-SSVM 算法具有很好的分类性能和很快的测试运行速度。

另外,为了说明所提出的 LDE-SSVM 分类算法的高效性,还对这 4 种分类算法的测试运行时间进行了比较,实验结果如表 3 所示。从中可以看出:提出的 LDE-SSVM 算法的测试运行时间普遍低于常用的 SVM、LSI-SVM 和 LDA-SVM 分类算法。原因在于:LDE 算法有效地避免了高维文档引起的"维数灾难"问题,缩小了 SSVM 分类器的搜索范围。另外,与传统 SVM 分类器相比,简化 SVM(SSVM)分类器大大减少了所需要测试的支持向量数目,从而保证了 LDE-SSVM 具有很快的测试运行速度。

5　结论

针对传统文档分类算法在处理高维大规模数据集时存在

表 1　在 Reuters-21578 上的分类性能比较

算法	准确率	$F1$	均衡点
SVM	0.876 1	0.865 3	0.813 4
LSI-SVM	0.883 5	0.867 2	0.821 6
LDA-SVM	0.916 4	0.872 6	0.825 1
LDE-SSVM	0.957 8	0.924 5	0.842 7

表 2　在 WebKB 上的分类性能比较

算法	准确率	$F1$	均衡点
SVM	0.864 2	0.852 8	0.863 5
LSI-SVM	0.876 9	0.863 1	0.859 4
LDA-SVM	0.921 8	0.894 5	0.878 3
LDE-SSVM	0.963 5	0.912 3	0.902 6

表 3　测试运行时间比较　s

算法	Reuters-21578	WebKB
SVM	5.73	4.58
LSI-SVM	4.52	4.34
LDA-SVM	2.31	2.26
LDE-SSVM	0.45	0.38

(下转 6 页)