申请上海交通大学硕士学位论文

# 基于距离度量学习的文本分类研究


**硕士生：**彭凯

**导　师：**杨煜普

**班　级：**B1003292

**学　号：**1100329079

**学　科：**控制理论与控制工程



**上海交通大学电子信息与电气工程学院**

**2013 年 2 月**

**A Dissertation Submitted to Shanghai Jiao Tong University for**

**Master Degree of Engineering**

# The Research of Text Classification Based on Distance Metric Learning

**Author:** Peng Kai

**Advisor:** Prof. Yang Yupu

**Student ID:** 1100329079

**Specialty:** Control Theory and Control Engineering

School of Electronic Information and Electrical Engineering

Shanghai Jiao Tong University Shanghai, P.R. China

February, 2013

# 上海交通大学
# 学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期：2013 年 2 月 22 日

# 上海交通大学
# 学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密□，在___年解密后适用本授权书。

本学位论文属于

不保密☑。

（请在以上方框内打“√”）

学位论文作者签名：[signature]　　指导教师签名：[signature]

日期：2013 年 2 月 22 日　　日期：2013 年 2 月 22 日

# 基于距离度量学习的文本分类研究

## 摘 要

文本分类技术作为作为现代互联网信息科技的重要分支在过去的二十年中有了长足的发展，然而随着互联网上 Web 页面数量的指数增长，互联网信息的多样性也呈现出越来越复杂的态势。如何改变传统的文本分类算法使其适应现代 Web 信息类别多样、低区分度等特性成为现在文本分类亟待解决的问题。距离度量学习算法是一类围绕样本之间相似度的度量模式来进行研究的机器学习算法，由于目前基于统计和机器学习的文本分类算法已经比较成熟，在分类精度方面很难再有更大的提高，因此如何改变样本的距离度量模式使其达到更好的分类效果，是当前的一个研究热点。此方面的研究已经在图像识别、分类领域有了比较成功的应用。

本文主要针对距离度量学习在文本分类中的应用展开研究，首先在广泛调研文献的基础上总结了目前已有的本领域相关工作，并介绍了几种常见的距离度量学习算法，其次介绍了文本分类的具体流程，并对其中关键算法进行了分析，最后根据文本分类的特点结合已有的距离度量学习算法根据在实际应用中出现的问题提出了一系列改进方案。

本文的主要工作有：

（1）在引入距离度量学习的基础上考虑到其对样本密度的影响，提出了改进方案。新的方案设计了一个密度函数与 K 近邻分类器相结合来平衡距离度量学习算法对样本数据的影响。

（2）在大边界最近邻（LMNN）算法的启发下，提出了一种新的基于余弦距离度量的学习算法（CS-LMNN），该算法更加适用于经典的向量空间模型下的文本分类。

（3）最后在上述理论基础上，实现了整个文本分类系统，包括预处理模块，特征选择模块，距离度量学习模块，分类模块以及评价模块。

**关键字：**文本分类，距离度量学习，密度，余弦，向量空间模型，大边界最近邻

# The RESEARCH OF TEXT CLASSIFICATION BASED ON DISTANCE METRIC LEARNING

## ABSTRACT

As an important branch of modern information technology text classification techniques has made great progress in past two decades, however, with the growth in the exponential of the number of Web pages on the Internet, the diversity of the Internet information is also showing more and more complextrend. How to change the traditional text classification algorithm to adapt them to diverse modern Web information categories, low discrimination characteristics become the most problems to be solved. Distance metric learning algorithm is a class of machine learning algorithms that research around the the sample similarity metric mode.In terms of text classification algorithm based on statistical and machine learning are already quite mature, to have a greater improve of classification accuracy becomes much more difficult. As a result, changing the sample distance metric mode to reach better classification results is a current research focus. This research has identified in the field of image and classification a successful application.

This article research is mainly expanded from distance metric learning for

text classification. First, based on the extensive research literature we summarized the existing work of the field,and introduced several common distance metric learning algorithms, Secondly, we introduced the text classification process, and analysed some key algorithms, At last, we combined some distance metric learning algorithms with existing text classification algorithms and proposed a series of improvements based on the problems in practical applications.

The main work of the article:

(1) Applied distance metric learning based on taking into account the impact of sample density, introduced improved scheme. The new scheme designed a density function combining with the K-nearest neighbor classifier to balance the bad impact of distance metric learning algorithm.

(2) Inspired by the large margine nearest neighbor (LMNN) algorithm, we proposed a new learning algorithm based on cosine distance metric called CS-LMNN, the algorithm is more suitable for the classic vector space model.

(3) Finally, basedon the theory mentioned before,we realized the entire text classification system, including classification module, pre-processing module, feature selection module, the distance metric learning modules, as well as evaluation module.

**KEYWORDS:** Text Classification Distance Metric Learning, Density,

Cosine, Vector Space Model, Large Margine Nearest Neighbor.

# 目 录

# 第一章 绪论

## 1.1 课题研究的背景及意义

随着 Internet 的迅猛发展，信息正在以爆炸式的方式增长，人们获得文本、图片、声音、视频等形式存在的信息也越来越容易，但是文本目前仍是信息的最主要载体。据统计，Google 早在 2008 年所索引的网页已经超过一万亿，而在十年前的 1998 年这一数目仅为 2600 万，所以如何管理和使用这样日益庞大的数据成为当前数据挖掘领域的最大难题。文本分类作为当前一种重要的数据挖掘手段被广泛应用于搜索引擎、数字图书馆、档案管理等涉及海量文本信息的系统中。它最早出现于上个世纪 50 年代，随着科技的发展其应用也越来越广泛例如在最近比较流行的情感分类、推荐系统中都有着文本分类技术的应用。文本分类过程可以分为手工分类和主动分类。前者最著名的实例是 yahoo 的网页分类系统，是由专家定义了分类规则，然后人工将网页分类。这种方法须要大批人力，现实中已经很少采用。自动文本分类（automatic text categorization）算法大致可以分为两类[1]：知识工程（knowledge engineering）方法和机器学习（machine learning）方法。知识工程方法指的是由专家为每个类别定义一些规矩，这些规矩代表了这个类别的特点，主动把符合规矩的文档划分到相应的种别中。这方面最有名的体系是 CONSTRUE。上个世纪 90 年代之后，机器学习方法成为主导。机器学习方法与知识工程方法相比，能够到达类似的准确度，但是减少了大量的人工参与。词匹配法是最早被提出来的文本分类算法，后来随着发展逐渐引入了基于统计的分类算法和基于机器学习的分类算法。

基于机器学习的文本分类技术由于其算法的稳定性和流程的标准性目前被普遍应用于科研和实践领域。它的主要流程包括文本的预处理、特征选择、分类模型建立等几个方面。随着机器学习的迅速发展，文本分类技术也相应有了很大的提高。然而这种分类方法的精度高度依赖于文本的预处理和特征选择，这也造成了目前文本分类领域的一个瓶颈。解决这一问题的思路大致分为两种，一是改进文本预处理和特征选择的算法尽量减少由于预处理和特征提取所造成的分类误差。在这一思路下[2][3]做出了一系列有意义的尝试。第二种思路独辟蹊径考虑样本间相似性的度量方式对分类算法的影响[4][5]，传统的欧式距离度量虽然简单方便，但是在很多情况下，无法准确度量两个样本间的相似性，通过特征变换找到两个样本间的更准确度

量可以一定程度提高分类准确率。在实际应用中可以使用机器学习的算法对特征提取后的训练样本进行学习，得到一个更加准确的度量方式，再进行分类，这在另一方面也弥补了文本预处理和特征提取阶段的误差。

目前，经过近几年的发展国内对于中文文本分类的研究已经有了长足的进步。但由于国内对于中文文本分类的研究起步较晚，一些新算法和相关技术的研究与国外还存在着一定的差距，特别是上述提到的使用距离度量学习的方式来解决目前文本分类领域遇到的瓶颈问题。在文本分类中，如何选择一种合适的距离度量，现有的距离度量学习算法是否存在不足与缺点，如何选择一种合适的特征选择算法，传统的特征选择算法的不足与缺点等都是亟待解决的问题。因此建立一套文本分类系统，并在此基础上对相关的技术问题进行探索和研究是十分必要的。

## 1.2 国内外相关的研究

### 1.2.1 文本分类算法研究现状

国外文本分类最早始于上个世纪五十年代。1959 年 IBM 的 H.P.Luhn 首次比较明确地提出了统计词频的思想[6]，同年 Maron、Kuhn 首次提出了自动分类技术[7]，确立了文本分类的基本流程，并且提出了概率标引模型，一直到 80 年代前相继出现了一批优秀的科学家对文本分类领域做出了巨大贡献，例如 Rosenblatt 设计了感知机[8]，Gerald Salton 在其有关信息检索的论文中提出了文本的向量空间表示模型[9]等。这一阶段内模式识别和信息检索有了长足的发展并且逐渐成为独立学科。同时，这也为以后文本分类的发展奠定了良好的理论基础。20 世纪 80 年代后随着文本数量的增长简单的依靠专家经验的人工分类方法已经无法满足实际应用需求。于是，基于机器学习的文本分类逐渐兴起，它一般讲样本分为训练集合测试集，通过对训练集进行学习得到分类规则，这种方法不依赖于特定专业知识分类结果准确性高，便于移植。这一时期的主要研究成果有 1990 年卡内基集团首先完成了自动文本分类系统 Construe.1994 年 Lewis、Ringuette 提出了决策树和贝叶斯分类器[10][11]。当前应用最广泛的支持向量机的自动文本分类算由 T. Joachims[12]等人提出，之后相继出现了最大熵模型和粗糙集等理论，并成功应用于文本分类领域。当前文本分类领域的研究方向主要集中在为文本分类的精度提高进行改进，这方面的研究可以分为以下几个方向：

（1）特征提取方面的研究，包括文档频率、信息增益（IG）、互信息（MI）、$\chi^2$

统计法等。

（2）分类模型建立方面的研究，其中包括决策树法，Rocchio[15]算法、K 近邻分类算法[14]、朴素贝叶斯算法[15]、支持向量机算法[12] [13]，神经网络算法[17]等。

（3）相似度度量方式方面的研究，这方面的研究目前主要集中在距离度量学习算法上,包括局部自适应距离度量学习算法（LAD）[18]、近邻元分析算法（NCA）[19]、大边界最近邻算法（LMNN）[4]等。

国内的文本分类研究起步比国外晚很多，直到上世纪 80 年代侯汉清教授才对文本分类做了系统的介绍[20]，由于中文文本的特点中文文本分类的预处理中要比英文文本分类多加一步分词阶段，这一方面中科院提出了基于多层隐马尔科夫模型的汉语词法分析系统（ICTCLAS）。黄萱菁等提出了一种基于机器学习的文本分类模型[21]，李荣陆等在最大熵模型的基础上对中文文本分类进行了研究[22]。董小国等根据句子重要程度的不同提出了一种新的特征项权重计算方法[23]。于一针对 KNN 算法搜索速度过慢的问题，提出了改进算法提高了搜索效率[24]。朱靖波等提出了一种混淆类的判别技术采用二阶段法来处理那些单一分类器无法准确分类的样本[25]。李文波等人根据目前较流行的 LDA(Latent Dirichlet Allocation)模型提出了一种附加类别标签的 LDA 模型用于文本分类[26]。罗长升等人提出了基于推拉策略的文本分类增量学习算法[27]。目前，中文文本分类尚没有一个标准的语料库，应用较多的主要有谭松波和王月粉共同整理的中文文本语料库 TanCorpV1.0 共包括 12 个大类总计 14150 篇，Sogou 实验室提供的基于搜狐分类目录手工编辑的 10 个类别近 8 万余篇文章。

从总体来看目前国内中文文本分类在各个方面从无到有在短时间内取得了交大的进展但相对于国外的研究我们仍处于融合国外不同算法的阶段，存在一定的差距，需要我们进一步的研究。

### 1.2.2 距离度量学习研究现状

特征空间中的距离度量是在模式识别领域研究的主要问题之一，R. A. Fisher. 在 1936 年就尝试了使用不同的度量方式来解决分类问题[28]。自此之后使用不同的距离度量方式被广泛应用于模式识别领域。在这一时期，W.L.G.Koontz 和 K. Fukunaga 提出了一种使用距离度量信息的非线性特征提取方法[29]。Y. LeCun 等根据一种新的距离变换提出了一种有效的模式识别方法[30]。T. Hastie 和 R. Tibshirani 提出了一种自适应判别的最近邻分类算法[31]。然而，距离度量学习真正的发展却是在 2000 年以后，在这一时期距离度量学习这一概念被正式提出，并且根据训练数据的有效性可以分为有监督和监督两类，有监督的距离度量学习又可以细分为全局的和局部的度

量学习算法[32]。其中，全局的度量学习算法如下，Eric P. Xing 等通过解一个有约束的凸规划提出了一种基于概率的全局距离度量学习算法（PGDM）[33]。Aharon Bar-Hillel 和 Tomer Hertz 利用等式约束条件通过攫取全局数据的结构提出了相关成分分析算法（RCA）[34]。Steven C.H. Hoi 等通过增加否定条件来改进 RCA 算法提出了有区别成分分析算法（DCA）[35]。Tomer Hertz 和 Aharon Bar-Hillel 在 boosting 框架下通过对二类分类器的边界训练得到距离函数提出了 DistBoost 算法[36]。局部的距离度量学习算法如下，Kilian Q. Weinberger 等人根据 K 近邻分类的特点，提出了通过解一种半正定规划问题来学习其马氏距离度量形式的算法称之为大边界最近邻算法（LMNN）[4]。Masashi Sugiyama 等人通过对近邻的连接设定更大的权值来改进线性判别分析算法（Linear Discriminant Analysis，LDA）提出了局部费舍尔判据分析算法（LFDA）[37]。Liu Yang 等在概率的框架下通过优化特征局部聚集和分离程度提出了局部距离度量学习算法（Localized Distance Metric Learning，LDM）[38]。Jacob Goldberger 等人通过估计样本的类条件概率分布提出了近邻元分析算法（NCA）[19]。无监督距离度量学习算法可以证明和线性流行学习算法是等价的，和大多数非线性流行学习算法在本质上也是有着相当大的关系的。其中,A. Bar-Hillel 等人通过解最能保持原数据结构的矩阵分解提出了主成分分析算法（PCA）[39]。这类算法还有，J.B.Tenenbaum 等提出的 ISOMAP 算法[40]，Mikhail Belkin 等人提出的拉普拉斯特征映射（Laplacian Eigenmap,LE) [41] ,S. Roweis 等人提出的局部线性嵌入 Locally Linear Embedding (LLE) [42] ,其中，ISOMAP 算法试图在子空间中寻找最能保持两点之间的测量距离，而 LE 和 LLE 算法侧重于保持原数据的近邻结构。

由上文可知，经过近十年的发展距离度量学习已经有相当大的发展但是仍然有一些问题没有解决列出如下：（1）目前，距离度量学习算法主要致力于寻找一种线性距离度量来调整原数据使其达到合适的聚集或分散，即对同类的数据要聚集，异类的数据要分离。然而通常在全局情况下这俩个条件往往是矛盾的，特别是在数据呈现多模分布时。（2）效率问题是困扰距离度量学习算法的一个重要因素，例如近邻元分析（NCA）和大边界最近邻算法（LMNN）都是将距离度量学习归结为解一个凸规划问题，这就导致计算量大大增加，并且训练数据往往又根据应用场合的不同而有所区别，在训练集有限或者数据维度过大时容易造成过拟合。因此，尽管应用距离度量学习已经有了很大的发展，但是继续研究它特别是研究将其应用到不同领域时容易出现的问题是特别有必要的。

## 1.3 本文的主要研究内容

本文重点研究距离度量学习算法，以及将其应用到文本分类领域时所遇到的问题，并提出了改进方法。首先，研究了文本分类的流程、主要算法和原理，包括中文分词、向量空间模型、分类方法和评价方法。之后研究了当前比较流行的距离度量学习算法包括有监督的 NCA 算法和 LMNN 算法，无监督的 PCA 算法和 RCA 算法等，并且提出了一种新的距离度量学习算法—余弦距离度量学习算法，称之为 CS-LMNN, 然后考虑到距离度量学习算法的初衷是为寻找分类聚类系统中数据之间的某种相似度度量，而在文本分类中 KNN 等依赖于相似度度量方式的分类算法的致命缺点就是针对不同的距离度量，分类效果相差巨大，所以提出将距离度量学习算法引入到文本分类系统，后又研究了将这类算法引入到文本分类时可能会出现的问题，对应提出了改进方案。最终在这些算法的基础上实现了文本分类系统，对实验结果进行了分析和对比。

## 1.4 论文的组织

本文围绕距离度量学习及其在文本分类领域的应用展开各章节内容安排如下：

第一章绪论，简明介绍了研究的背景，包括文本分类和距离度量学习的发展现状，并且提出了本文的研究内容。

第二章距离度量学习算法。介绍了距离度量学习的问题、分类和与流形学习之间关系，然后介绍了几种目前效果较好的算法，包括主元分析(PCA)、近邻元分析(PCA)，大边界最近邻算法(LMNN)等，并对这些算法进行了较为深入的比较探讨。

第三章文本分类的流程、原理以及实现，介绍了通用文本分类的流程，各个步骤中涉及的关键技术，并对这些技术进行了较为深入的比较探讨，最后编程实现了文本分类系统。

第四章基于大边界最近邻算法(LMNN)的文本分类，首先讨论了文本分类中 KNN 算法的两个缺点：1)分类效果高度依赖于相似度的度量方法，2)对数据密度分布较为敏感，并且根据这两点分别提出了，基于 LMNN 算法的 KNN 文本分类和基于密度加权的 K 近邻分类两种算法来对 KNN 文本分类系统做改进，最后实验证明改进后的算法是有效的。

第五章基于余弦距离度量学习的伪 K 近邻文本分类算法，首先分析了目前距离度量学习算法的一个缺点，然后对应此缺点提出了一种新的距离度量学习算法—余

弦距离度量学习，称之为 CS-LMNN 算法，在将其应用到文本分类时同样考虑密度和类偏斜带来的分类误差，又提出使用一种伪 K 近邻分类算法来作为分类器，最后实现了整个文本分类系统，采用现实语料库对实验结果进行了分析对比。

第六章总结和展望，总结了本文的研究内容，对文本分类算法与距离度量学习的结合进行了展望。

# 第二章 距离度量学习算法

本章介绍了距离度量学习算法思想、原理，以及几种效果比较流行的距离度量学习算法。

## 2.1 距离度量学习算法概述

样本之间的相似性度量是模式识别领域研究的核心问题之一，大量的机器学习方法如 K 近邻、径向基函数网络、支持向量机等性能的好坏主要由样本之间的相似度度量方式决定的[32]。目前，距离度量学习可以分为有监督的距离度量学习和无监督的距离度量学习算法两类，无监督的距离度量学习算法，没有样本类别标注的参与，多见于谱分析的降维算法中，如 PCA 等；有监督的距离度量学习算法利用样本之间的样本对约束(pairwise constraints)条件:同类样本之间形成的等价性约束(equivalence constrains)和异类样本之间的非等价性约束(inequivalence constrains)最小化马氏(Mahalanobis)距离意义下的同类样本之间的距离，最大化异类样本之间的距离来实现，这类算法有，近邻元分析(Nearest Component Analysis, NCA),大边界最近邻算法(Large margin nearest neighbor, LMNN)等。

### 2.1.1 问题的描述

距离度量学习在本质上是寻找一个度量函数 $D_M(\bullet,\bullet)$ 使得任意的三个向量 $\vec{a}$，$\vec{b}$，$\vec{c}$ 满足：

(1) 非负性 $D_M(\vec{a},\vec{b}) \geq 0$；

(2) 自反性 $D_M(\vec{a},\vec{b})=0$，当且仅当 $\vec{a}=\vec{b}$；

(3) 对称性 $D_M(\vec{a},\vec{b})=D_M(\vec{b},\vec{a})$；

(4) 三角不等式 $D_M(\vec{a},\vec{b})+D_M(\vec{b},\vec{c}) \geq D_M(\vec{a},\vec{c})$。

在现实应用中普通 n 维空间的欧式距离和更加广义的 Minkowski 距离，即 $L_K$ 范数，有着简单方便的特点，但是在 K 近邻分类中随着维数的增长样本与其近邻之间的距离和样本与其他样本之间的距离差异呈逐渐减小的趋势，而且对于分类问题中，

由于每一维对于分类的重要程度显然应该是不同的，直接采用 $L_K$ 范数意味着在计算相似度时平等对待每一维特征值，这样做显然是不合理的。

距离度量学习的目的是通过寻找一个合适的距离度量矩阵 $M$，计算样本 $d_i, d_j$ 之间的马氏距离(Mahalanobis)度量:

$$D_M\left(d_i, d_j\right)^2 = (d_i - d_j)^{\mathrm{T}} M (d_i - d_j) \tag{2-1}$$

其中, $M$ 是半正定对称矩阵可以表示为 $M = L^{\mathrm{T}}L$，这等价于寻找一个 $L$ 矩阵作为映射矩阵将原数据 $d$ 映射到一个新的分类空间 $L:\mathbb{R}^n \to \mathbb{R}^m$，即 $d^{'} \to Ld$ 上式可以改写为如下形式：

$$D_M\left(d_i, d_j\right)^2 = (Ld_i - Ld_j)^{\mathrm{T}} (Ld_i - Ld_j) \tag{2-2}$$

因此，欧式距离可以看成矩阵 $L$ 为单位阵时的特例。

### 2.1.2 距离度量学习的一般意义

对于 n 维样本 $d \in \mathbb{R}^n$，距离度量学习可以看成一个将输入数据从 n 维空间映射到 m 维空间的映射这里 $n \geq m$，对于线性变换有 $d^{'}=Ld$，这种映射一般包含两种操作，即旋转和尺度变换。下面分别讨论 $L$ 在不同形式下的变换意义[61]。

(1) 若 $L$ 为单位阵，则此时相当于欧式距离度量，样本不做任何变换。

(2) 若 $L$ 为为对角阵，此时向本制作尺度变换不做旋转。若每维对角线上的元素分别是相应维上特征的标准差 $\sigma_i$，即 $L = diag(1/\sigma_1, 1/\sigma_2 \cdots 1/\sigma_n)$ 时相当于对样本进行归一化，归一化得到的样本每维特征值的方差均为 1。若对角线上的元素非 0 即 1 时相当于对样本做了特征选择，对应维数为 0 的位在映射后将不起作用。

(3) 如果 $L$ 为正交阵，则只做旋转不做尺度变换（正交变换），不改变原先距离度量。这种变化的意义是，通过正交变换找到一个新的坐标系统，在该坐标系统下特征的分析更加明确。

(4) 若 $L$ 为方阵则，通常包含了旋转和尺度变换两种操作。若此方阵不为满秩的，则又包含降维操作。

## 2.2 距离度量学习与流形学习的关系

流形学习方法是模式识别中的基本方法，分为线性流形学习算法和非线性流形学习算法，非线性流形学习算法包括等距映射（Isomap），拉普拉斯特征映射（LE），局部线性嵌入(LLE)等。而线性方法则是对非线性方法的线性扩展，如局部保持投影（LPP），邻域保持嵌入（NPE）等。假设数据是均匀采样于一个高维欧氏空间中的低维流形，流形学习就是从高维采样数据中恢复低维流形结构，即找到高维空间中的低维流形，并求出相应的嵌入映射，以实现维数约简或者数据可视化。它是从观测到的现象中去寻找事物的本质，找到产生数据的内在规律。距离度量学习是通过对原数据分析学习得到一种更能代表数据特征度量矩阵，使其在分类时能够将原数据映射到一个更优分类空间[62]。

### 2.2.1 距离度量学习与线性映射

定义在空间 $R^N$ 中两点 $x, y$ 的距离为 $d(x,y)^2=(x-y)M(x-y)^{\mathrm{T}}$，距离度量学习的典型问题就是学习得到矩阵 $M\in R^{N\times N}$，上式可以改写为：

$$\begin{aligned} d(x,y)^2 &= (x-y)M(x-y)^{\mathrm{T}} \\ &= (x-y)\mathrm{M}^{\frac{1}{2}}(\mathrm{M}^{\frac{1}{2}}(x-y))^{\mathrm{T}} \\ &= (Lx-Ly)(Lx-Ly)^{\mathrm{T}} \end{aligned} \tag{2-3}$$

其中，$L=\mathrm{M}^{\frac{1}{2}}$ 可以看出学习一个距离度量矩阵 $\mathrm{M}$ 等价于学习一个线性映射 $L$。因此，在流行学习中学习一种线性变换即线性映射，等价于学习 $\mathrm{M}^{\frac{1}{2}}$。换句话说，线性流行学习算法就可以等价为距离度量学习。

### 2.2.2 距离度量学习与非线性流行学习

对于非线性流行学习算法，学习结果一般没有一个明确的映射矩阵，只是产生一个特定的嵌入。不同的算法由于初衷不同，所采用的约束条件也就不同，因此想要了解不同非线性流行学习算法与距离度量学习之间的关系，必须对具体分析每一种算法的约束。一般距离度量学习算法使用二进制度量，即 1 表示必要性连连接(must-link)，0 表示无关连接(cannot-link),具体应用有对约束条件(pairwise constraints)

和团簇(chunklest)。在流行学习中考察 Isomap 算法，它通过计算数据子格的最短路径来得到保持原数据测量的流形。LLE 算法通过计算局部邻域的线性组合来得到流形表示。这两种方法最后的数据均可以用$(x_i - x_j)\mathrm{M}^{\frac{1}{2}}$来表示。具体分析如下：

定义$W$为权值矩阵，$X = \{x_1, x_2, \cdots x_N\} \in R^{M \times N}$为数据矩阵，它在原始特征空间包含 N 个数据，$Y = \{y_1, y_2, \cdots y_N\} \in R^{m \times N}$为得到的非线性嵌入矩阵其中，$m \le M$。LLE 算法通过最小化损失函数$\Phi(Y) = \sum_i \left\| y_i - \sum_{i=1}^{K} W_{ij}^{*} y_{ij} \right\| = \left\| Y^{\mathrm{T}} M Y \right\|^2$来构造近邻保持流形，其中$M = (I - W^{*})(I - W^{*})^{\mathrm{T}}$，$W^{*} = \arg\ \min \sum_i \left\| x_i - \sum_j W_{ij} x_j \right\|^2$，近邻保持嵌入(Neighborhood Preserving Embedding, NPE)引入了线性映射$B \in R^{M \times m}$，有$Y = B^T X$。因此上述最优化问题可以归结为寻找$B = \arg\min_{B^{\mathrm{T}} X X^{\mathrm{T}} B = 1} B^{\mathrm{T}} X M X^{\mathrm{T}} B$，又等价为求解一般特征值问题：$XMX^{\mathrm{T}} B = \lambda XX^{\mathrm{T}} B$，矩阵$M$在流形上提供了一个近似的离散拉普拉斯贝特拉密操作，这就是说 NPE 也在流形上提供了一种拉普拉斯贝特拉密操作(Laplace Beltrami operater)特征方程的线性近似。同理可以证明 LE 算法也可以通过某种线性近似和距离度量学习联系起来。

综上所述，线性的流行学习算法和距离度量学习算法所求解的问题具有一致性，在一定意义上可以说两者是等价的；非线性的流行学习算法在本质上也和距离度量学习有着某种关系。

## 2.3 几种常见的距离度量学习算法

按照分类本节将依次介绍几种无监督的距离度量学习算法和有监督的距离度量学习算法。

### 2.3.1 无监督的距离度量学习

无监督的距离度量学习在本质上说与流形学习是一致的，基本思想是根据现有的训练数据找到一种最能描述原数据流形的低维表达。问题可以描述为对于给定的训练集$D = \{x_1, x_2, \cdots, x_n\} \in R^M$，找到一个合适的描述$Y = \{y_1, y_2, \cdots, y_n\} \in R^m$使得$M \ge m$

成立。这之中较为著名的是 PCA 算法。

主成分分析算法(Principal Component Analysis, PCA)，以样本点在空间中的变化最大方向，即方差最大的方向，作为判别标准来描述原训练数据。从统计观点可知，一个随机变量的方差越大，该随机变量所包含的信息就越多，如果一个变量的方差为 0 时，该变量为常数。所谓的主成分就是原始数据经矩阵 $L$ 映射后的最能反映其变量方差的 m 个变量。主成分分析的目标是找到一组相互正交的投影方向或一列正交的线性投影矩阵 $L \in R^m$，使得投影后的低维嵌入表示 $y_i = L^{\mathrm{T}} x_i$ 具有最大的方差。如上文表述，可求训练样本协方差矩阵为：

$$S_t = \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^{\mathrm{T}} = XHX^{\mathrm{T}} \tag{2-4}$$

其中，$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ 为样本均值，$H = I - \frac{1}{n} ee^{\mathrm{T}}$ 为中心化矩阵，$I$ 为单位阵，$e \in R^n$ 是元素全为 1 的列向量。然后，可求低维嵌入表示的协方差矩阵为：

$$\sum_{i=1}^{n} (y_i - \bar{y})(y_i - \bar{y})^{\mathrm{T}} = YHY^{\mathrm{T}} = L^{\mathrm{T}} XHX^{\mathrm{T}} L = L^{\mathrm{T}} S_t L \tag{2-5}$$

其中，$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ 表示低维嵌入的样本均值，PCA 的目标函数可以表示为下述形式：

$$\begin{aligned} &\arg\max \quad tr(L^{\mathrm{T}} S_t L) \\ &S.t. \quad\quad L^T L = I \end{aligned} \tag{2-6}$$

该目标函数的最优解 $L$ 可以通过对原始训练数据的协方差矩阵 $S_t$ 进行谱分解或特征分解来求解。即假设 $S_t$ 的谱分解为：

$$S_t = U \Lambda U^{\mathrm{T}} \tag{2-7}$$

其中，$\Lambda = diag(\lambda_1, \cdots, \lambda_D)$ 是由 $S_t$ 的特征值组成的对角矩阵，且满足 $\lambda_i \geq \lambda_{i+1}$，$U = [u_1, u_2, \cdots u_D]$ 为对应的特征向量，且满足 $U^{\mathrm{T}} U = I$。综上，PCA 的映射过程其实就是一个坐标变换过程，即将高维观测数据往方差最大方向组成的坐标系投影，从而做到方差最大化和投影误差最小化。

这一类方法的特点是一般都有较好的可解释性，计算流量较小，但是对于特定结构的数据在应用与文本分类时仅仅根据数据的某一特征进行映射往往不能得到更优的分类空间。

### 2.3.2 有监督的距离度量学习

在文本分类领域中往往有这中想法：使同类的数据点尽量集中，异类的数据点尽量远离,这样在应用一些分类算法,如 KNN、SVM 时可以有效的降低分类误差。基于这种思想，有监督的距离度量学习利用上述的对约束条件(Pairwise Constraints)来构造凸规划问题进行求解。所谓的对约束条件可以描述为:对于训练集 $D=\{x_1,x_2,\cdots,x_n\}\in R^M$ 有：等价性约束集合 $S_s=\{(x_i,x_j)\,|\,x_i和x_j属于同类\}$，非等价性约束集合 $S_d=\{(x_i,x_j)\,|\,x_i和x_j属于异类\}$。并且定义两训练数据 x,y 之间的距离为: $\mathrm{d}_M(x,y)=\sqrt{(x-y)^{\mathrm{T}}M(x-y)}$,其中,$M\succeq 0$ 为距离度量矩阵。这一类算法主要有基于凸规划的全局距离度量学习算法(Global Distance Metric Learning by Convex Programming, GDMLCP)、近邻元分析(Neighborhood Component Analysis, NCA)、相关量分析(Relevant Component Analysis, RCA)、大边界最近邻算法(Large Margin Nearset Neighbour, LMNN)等几种。

(1) 基于凸规划的全局距离度量学习算法(Global Distance Metric Learning by Convex Programming, GDMLCP) [33]是比较早的一种有监督距离度量学习算法，它的初衷就是根据样本的类标签先验知识，来确定对约束条件使学习到的距离度量矩阵可以反映这种条件限制关系。定义样本集合{X,C}，其中 X 表示文本集合，C 表示类别集合，定义等价性约束子集 $S_s=\{(x_i,x_j)\,|\,c_i=c_j\}$，非等价性子集 $S_D=\{(x_i,x_j)\,|\,c_i\neq c_j\}$，距离度量矩阵可以通过解以下凸规划问题得到：

$$\begin{aligned}&\min_M \quad \sum_{(x_i,x_j)\in S_s}\left\|x_i\text{-}x_j\right\|_M^2\\ &S.t \quad M\succeq 0\\ &\qquad \sum_{(x_i,x_j)\in S_D}\left\|x_i\text{-}x_j\right\|_M^2\geq 1\end{aligned}\tag{2-8}$$

其中，$\left\|x_i\text{-}x_j\right\|_M^2=(x_i\text{-}x_j)^{\mathrm{T}}\mathrm{M}(x_i\text{-}x_j)$ 表示样本 $x_i$ 和 $x_j$ 在距离度量矩阵为 $M$ 时的马氏距离(Mahalanobis distance)。上式解法可以根据预期得到的矩阵 $M$ 的不同可以有不同的解决方案，一般意义下，我们希望求得的是对原数据的另一种

表达，包括旋转和尺度变换。即希望求得一个普通矩阵，此时可以采用“梯度下降+逐次映射”的方法求解，具体求解方式如下图 3.1 所示：

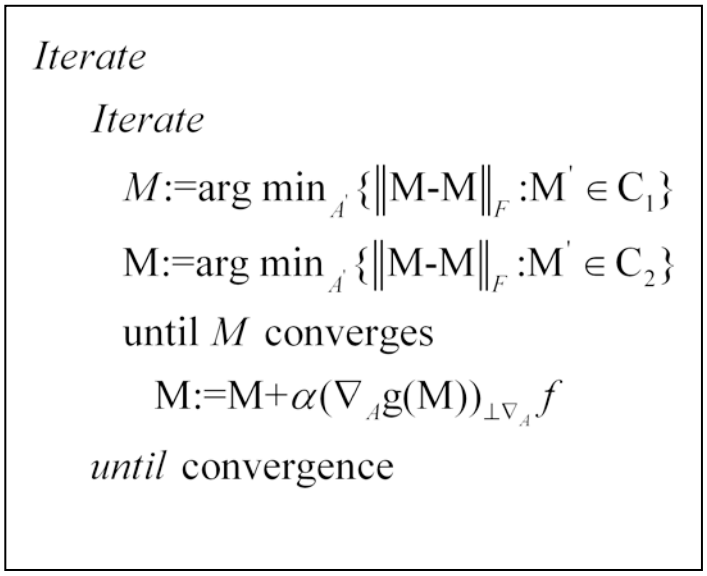


图 2-1 梯度下降+逐次映射求解过程示意图

Fig.2-1 Gradient descent + Successive mapping process diagram

(2) 近邻元分析(Neighborhood Component Analysis, NCA)是由 Goldberger 等人提出的一种基于概率的距离度量学习算法，NCA 算法一般在应用时还被用来做降维和数据可视化，这样可以有效的减小计算机存储负担并且可以更加形象的展示数据之间的关系。该算法目前主要应用在人脸识别、语音识别等领域，在这些领域中分类算法往往是基于近邻判别的，它可以有效地解决近邻元之间的距离度量，并且在一定程度上实现降维降低矩阵计算的复杂度。令距离度量矩阵 $M = L^{\mathrm{T}}L$，通过学习 $L$ 来得到 $M$ 以保证学习得到的矩阵为对称半正定矩阵，对于每个训练样本 $x_i$，可以认为除了自身以外的所有训练样本都以一定的概率称为它的近邻，这个概率可以通过下式计算得到：

$$p_{ij} = \frac{\exp\left(-\left\|Lx_i - Lx_j\right\|^2\right)}{\sum_{k\neq i}\exp\left(-\left\|Lx_i - Lx_k\right\|^2\right)}, p_{ii} = 0 \tag{2-9}$$

其中，$p_{ij}$ 定义为：点 $i$ 将另外一个点 $j$ 作为其近邻并且获取其列标签 $c_j$ 的概率。那么 $i$ 点被正确分类的概率就是：

$$p_i = \sum_{j \in C_i} p_{ij} \tag{2-10}$$

其中，$C_i = \{ j \mid c_i = c_j \}$,则总的分类正确率在概率意义下可以表示为：

$$f(L) = \sum_i \sum_{j \in C_i} p_{ij} = \sum_i p_i \tag{2-11}$$

这里，与实际意义下的分类正确率函数不同，概率意义下的分类正确率是可微的，$f(L)$的导数推导可以表示为下式：

$$\frac{\partial f(L)}{\partial L} = \frac{\partial}{\partial L} \sum_i \sum_{j \in C_i} \frac{\exp\left(-\left\| Lx_i - Lx_j \right\|^2\right)}{\sum_{k \neq i} \exp\left(\left\| Lx_i - Lx_k \right\|^2\right)} \tag{2-12}$$

$$\frac{\partial}{\partial L} \exp\left(-\left\| Lx_i - Lx_j \right\|^2\right) = \frac{\partial}{\partial L} \exp\left(-L^2 x_{ij}^2\right) = -2Lx_{ij}^2 \exp\left(-L^2 x_{ij}^2\right) \tag{2-13}$$

$$\frac{\partial}{\partial L} \sum_{k \neq i} \exp\left(-\left\| Lx_i - Lx_j \right\|^2\right) = -2L \sum_{k \neq i} \left( x_{ik}^2 \exp\left(-L^2 x_{ik}^2\right)\right) \tag{2-14}$$

$$\exp\left(-L^2 x_{ik}^2\right) = p_{ik} \sum_{k \neq i} \exp\left(-L^2 x_{ik}^2\right) \tag{2-15}$$

$$\frac{\partial f}{\partial L} = -2L \sum_i \sum_{j \in C_i} p_{ij} \left( x_{ij} x_{ij}^T - \sum_k p_{ik} x_{ik} x_{ik}^T \right) \tag{2-16}$$

其中，$x_{ij} = x_i - x_j$。

在具体计算时，写成更容易计算的表达式：

$$\frac{\partial f}{\partial L} = 2L \sum_i \left( p_i \sum_k p_{ik} x_{ik} x_{ik}^T - \sum_{j \in C_i} p_{ij} x_{ij} x_{ij}^T \right) \tag{2-17}$$

最终，距离度量矩阵$L$可以采用基于梯度下降的最优化方法对下式求解得到：

$$\arg_L \ \max \sum_{i=1}^{n} \log\left( \sum_{j \in C_i} p_{ij} \right) \tag{2-18}$$

(3) 相关量分析(Relevant Component Analysis, RCA)是一种试图寻找并且降低数据中不需要特征的方法。它通过数据集中的“相关量”赋大权值，“不相关量”赋小权值实现全局的线性变换。定义“团簇”(Chunklet)是数据集中某一未知类别属于同一类的一个子集(可以由等价性约束定义)，这些相关量可以由这些“团簇”来说明。RCA 的目的是降低数据的凌乱性，使其在映

射后的空间中能够更加容易分类。其具体算法流程如下;

1) 对于每一个团簇中的每一样本减去该团簇的样本均值。

2) 根据团簇中的每一个点计算协方差矩阵，假设在第 $k$ 个团簇中含有 $p$ 个点，并且在团簇 $j$ 中含有这些点 $\{x_{ji}\}_{i=1}^{n_j}$，其均值为 $m_j$。相应矩阵计算公式如下：

$$C=\frac{1}{p}\sum_{j=1}^{k}\sum_{i=1}^{n_j}(x_{ji}-m_j)(x_{ji}-m_j)^{\mathrm{T}} \tag{2-19}$$

3) 利用协方差矩阵来计算转换矩阵 $W=C^{-\frac{1}{2}}$，对元样本做白化变换 $x^{new}=Wx$，或者也可以使用 $C$ 的逆矩阵作为马氏距离度量矩阵。

在文献[34]中将 RCA 算法归结为一种团簇约束下信息最大化的最优化求解过程，它们主要基于信息论的判别准则，通过寻找对应输入数据 x 的相应模式转化 y,在某种一致性约束下最大化互信息 $I(X,Y)$。定义在空间 $R^n$ 中有数据集 $X=\{x_l\}_{l=1}^{n}$，经模式 $f$ 映射后对在空间 $R^m$ 中应有 $Y=\{f(x_l)\}_{l=1}^{n}$，其中模式转换 $f$ 使团簇中的点满足同类相互靠近的约束。上述问题可以归结为以下最优化问题：

$$\begin{aligned}&\max \quad I(X,Y)\\ &S.t \quad \frac{1}{p}\sum_{j=1}^{k}\sum_{i=1}^{n_j}\left\|y_{ji}-m_j^{y}\right\|^2\le K\end{aligned} \tag{2-20}$$

其中，$m_j^{y}$ 表示映射后的第 j 个团簇中数据点的均值，K 为阈值为常数，对于给定的 $f$ 每一个 X 都对应着确定的 Y，所以求 $I(X,Y)$ 的最大值可以等价为求 Y 的熵的最大值 $H(Y)$。定义 $|J(x)|$ 为 $x$ 的 Jacobian 变换，有 $p_y(y)dy=p_x(x)/|J(x)|dx$，因此 $H(Y)$ 可以表示为如下形式：

$$\begin{aligned} H(Y) &= -\int_y p(y)\log(p(y))dy = -\int_x p(x)\log\frac{p(x)}{|J(x)|}dx \\ &= H(x) + \left\langle \log|J(x)| \right\rangle_x \end{aligned} \tag{2-21}$$

在 $I(X,Y)$ 中，影响变换矩阵 $L$ 的只有 Jacobian 项，因此对应公式(2-19)可以写成下式：

$$\begin{aligned} &\max_L |L| \\ &S.t \quad \frac{1}{p}\sum_{j=1}^{k}\sum_{i=1}^{n_j}\left\| x_{ji} - m_j^y \right\|_{L^{\mathrm{T}}L}^2 \le K \end{aligned} \tag{2-22}$$

记相应的马氏距离度量矩阵为 $M = L^{\mathrm{T}}L$，此时 $M$ 为正定的，并且有 $\log(|L|) = \frac{1}{2}\log(|M|)$。上式可以最终表示为：

$$\begin{aligned} &\max_M |M| \\ &S.t \quad \frac{1}{p}\sum_{j=1}^{k}\sum_{i=1}^{n_j}\left\| x_{ji} - m_j^y \right\|_M^2 \le K, M \succ 0 \end{aligned} \tag{2-23}$$

接拉格朗日算子可得最终结果 $M = \frac{K}{N}C^{-1}$，其中 N 为特征空间的维数，C 为团簇(chunklet)的协方差矩阵。

(4) 大边界最近邻算法(Large Margin Nearset Neighbour, LMNN)是由 Weinberger 等人提出的一种应用于近邻分类算法的距离度量学习算法。它的动机非常简单对于 K 近邻分类，根据对约束条件(pairwise constraints)，在训练集中目标样本周围 K 个近邻中每个邻居中与目标样本类标签相同的点应尽量靠近，类别标签不同的点应与目标样本尽量远离。特别地，LMNN 算法只惩罚与目标样本标签相同但是距离目标样本较远和与目标样本标签不同但是距离目标样本较近的点。LMNN 算法与其它算法不同的地方是，它是针对近邻分类算法所提出的，在计算时需要训练集中每个样本的 K 近邻先验知识。假设，目标样本 $x_i$ 具有类标签 $c_i$ 在其 K 近邻点中有 $x_l$ 类标签为 $c_l$，定义噪声点为对任意目标样本 $x_i$ 有 $c_l \neq c_i$，满足：

$$\left\| L(x_i - x_l) \right\|^2 \le \left\| L(x_i - x_j) \right\|^2 + 1 \tag{2-24}$$

其中，$L$ 为距离度量矩阵。根据对约束条件,首先定义非等价性约束

(inequivalence constrains)：

$$\varepsilon_{push}(L)=\sum_{i,j\in K_pNN}\sum_{l}(1-y_{il})[1+D_L\left(x_i,x_j\right)-D_L\left(x_i,x_l\right)]_+ \tag{2-25}$$

其中，$D_L\left(x_i,x_j\right)=\left\|L(x_i-x_j)\right\|^2$ 表示映射后点 $x_i$ 和 $x_j$ 的距离度量；$j\in K_pNN$ 表示训练样本 $x_i$ 为测试样本 $x_j$ 的 K 近邻，此 K 近邻为先验知识以 $K_p$ 表示；$x_l$ 表示与处于 $x_i$ 最大边界内但又与测试样本类标签不相同的训练样本；$c_l$ 为 $x_l$ 的类标签；当 $x_i$ 的类标签 $c_i=c_l$ 时 $y_{il}=1$ 否则为 0；$[Z]_+=\max(Z,0)$。

定义等价性约束(equivalence constrains):

$$\varepsilon_{pull}(L)=\sum_{i,j\in K_pNN}D_L\left(x_i,x_j\right) \tag{2-26}$$

式中各要素定义如上文。最终，结合式(2-25)和(2-26)构造如下损失函数(loss function)：

$$\varepsilon(L)=(1-\mu)\varepsilon_{pull}(L)+\mu\varepsilon_{push}(L) \tag{2-27}$$

其中，$\mu$ 为权重系数一般取 0.5。可以看出惩罚函数的第一项 $\varepsilon_{pull}(L)$ 只惩罚与测试样本 $x_i$ 类标签相同但是距离处于最大边界之外的训练样本，第二项 $\varepsilon_{push}(L)$ 只惩罚与测试样本 $x_i$ 类标签不同但是又处于最大边界之内的训练样本，这样就保证了在求全局最优映射 $L$ 时只对影响 KNN 分类的点进行惩罚可以有效降低错误率和计算复杂度。公式(2-27)中所求线性变换 $\boldsymbol{L}$ 是非凸的使用梯度下降法求解时有可能陷入局部最优解，对于不同的问题给定的初始矩阵 $L$ 不同最终结果也不同，这对于某些问题可能不具有可重现性，应用性较差。通过对公式(2-27)重构，可以变为一个半正定规划问题。具体方法如下：首先，定义对称半正定矩阵 $M=L^TL$，使用矩阵 $\boldsymbol{M}$ 代替矩阵 $L$，公式(2-27)可以改写为：

$$\varepsilon(M)=(1-\mu)\sum_{i,j\in K_pNN}D_M\left(x_i,x_j\right)+\mu\sum_{i,j\in K_pNN}\sum_{l}(1-y_{il})[1+D_M\left(x_i,x_j\right)_M-D_M\left(x_i,x_l\right)]_+ \tag{2-28}$$

其中，$D_M\left(x_i,x_j\right)=(x_i-x_j)^{\mathrm{T}}M(x_i-x_j)$，且$M\succeq 0$；其余元素定义与上文相同，若将上式转换为凸规划问题求解，需要首先将其转化为更标准的形式，引入非负松弛变量$\{\xi_{ijl}\}$，来衡量式(2-24)的非等价性，构造以下半正定规划问题(Semidefinite Program, SDP)：

$$\begin{aligned}&min\ \ (1-\mu)\sum_{i,j\in K_PNN}(x_i-x_j)^{\mathrm{T}}M(x_i-x_j)+\mu\sum_{i,j\in K_PNN,l}(1-y_{il})\xi_{ijl}\\&(1)\ \ (x_i-x_l)^{\mathrm{T}}M(x_i-x_l)-(x_i-x_j)^{\mathrm{T}}M(x_i-x_j)\geq 1-\xi_{ijl}\\&(2)\ \ \xi_{ijl}\geq 0\\&(3)\ \ M\succeq 0\end{aligned} \tag{2-29}$$

在实际问题中大部分文本数据都可以很好的区分开，$\xi_{ijl}$就有很好的稀疏性，目标函数的不连续性对此半正定规划问题限制较小。所以，最终此最优问题可以通过次梯度下降法（Sub-Gradient Descent）来求解，具体过程如下：

$M_0:=I$ {Initialize with the identity matrix}
$t:=0$ {Initialize counter}
$\mathcal{N}^{(0)},\mathcal{N}_0:=\{\}$ {Initialize active sets}
$G_0:=(1-\mu)\sum_{i,j\in K_PNN}(x_i-x_j)(x_i-x_j)^T$
*while* (not converged) *do*
  *if* mod(t,someconstant) $=0\vee$ (almost converged) {we used someconstant=10} *then*
    *compute* $\mathcal{N}_{t+1}$ exactly
    $\mathcal{N}^{(t+1)}:=\mathcal{N}^{(t)}\bigcup\mathcal{N}_{t+1}$ (update active set)
  *else*

$$\text{compute}\mathcal{N}_{t+1} \approx \mathcal{N}_{t+1} \cap \mathcal{N}^{(t)} \{\text{only search active set}\}$$

$$\mathcal{N}^{(t+1)} := \mathcal{N}^{(t)} \{\text{keep active set untouched}\}$$

$$end\ if$$

$$G_{t+1} := G_t - \mu \sum_{(i,j,l)\in\mathcal{N}_t-\mathcal{N}_{t+1}} (C_{ij} - C_{il}) +$$

$$\mu \sum_{(i,j,l)\in\mathcal{N}_{t+1}-\mathcal{N}_t} (C_{ij} - C_{il})$$

$$\{C_{ij} = (x_i - x_j)(x_i - x_j)^{\mathrm{T}}\}$$

$$M_{t+1} := \mathcal{P}_S(M_t - \alpha G_{t+1}) \{\text{Take gradient step}$$

$$\text{and project onto SDP cone}\}$$

$$t := t + 1$$

$$end\ while$$

$$output\ M_t$$

图 2-2 半正定规划问题次梯度下降法求解过程示意图

Fig.2-2 SDP Sub-Gradient Descent process diagram

## 2.4 本章小结

本章详细介绍了距离度量学习的动机、问题描述和问题的一般意义。第二节又通过对几种流行学习算法的分析说明了距离度量学习与流形学习的异同。最后第三节详细介绍了两类距离度量学习算法:有监督的距离度量性学习算法和无监督的距离度量学习算法，并且重点对这两类算法中的典型算法在算法原理和算法流程等方面做了介绍和分析，为下一章中距离度量学习算法在文本分类中的具体应用做了铺垫。

# 第三章 文本分类的流程、原理以及实现

目前，主流的文本分类是一个有监督的机器学习过程,他根据一个已经被标注的训练文档集合，找到文档特征和文档类别之间的关系模型，然后利用这种学习到的模型对新的文档进行判别。数学上可以将这一过程描述为一个一对一或者一对多的映射。假设 $D=\{d_1,d_2,\cdots,d_n\}$ 为文档集合 $\vec{x}_j$ 为文本向量， $C=\{c_1,c_2,\cdots,c_m\}(m\geq 2)$ 为类别集合，目标模型 $T$ ，有：

$$T:D\rightarrow C \tag{3-1}$$

这里 $T(\vec{x})$ 是已知的，通过对训练集进行学习，目标是得到一个近似模型 $H$ ，有：

$$H:D\rightarrow C \tag{3-2}$$

对一个新的文档 $d$ ，分类结果 $H(d)$，对一个给定的评估函数 $f$ 有：

$$Min\ (\sum_{i=1}^{D} f(T(d_i)-H(d_i))) \tag{3-3}$$

本章主要围绕基于机器学习的文本分类展开，主要从流程、算法原理以及系统实现几个主要方面做了介绍。

## 3.1 文本预处理

文本作为信息的载体在非结构化的格式下是无法被计算机理解的，文本预处理的目的就是对非结构化的文本进行去除格式标记、分词、去停用词得到格式规范统一的文本数据为下一步分类做准备。

### 3.1.1 去除格式标记

在文本分类时需要处理的文本必须是纯文本，不附带任何格式标签的文本文档。随着信息化程度的加快获得各种格式文本的途径越来越多，如何去除文本的格式标签变得越来越重要。目前，应用较为普遍的文本样式是 web 页面文本，这类文本在处理时需要删除 html 标签，识别文档主副标题，正文内容，广告内容等。

### 3.1.2 分词

在去除格式标记后的文本中，要继续对文本进行处理需要对文本进行分词。英文中词与词之间都有空格分割，从语义的准确性和复杂性来说都比较易于处理。由于汉语语言的特点是以字组词，以词组句，以句成文，因此在处理中文文本时要先对文章进行分词，从句子中划出有独立意义的词作为识别语义的基本单位。

常用的分词分词算法主要分为三类[43]：基于字符串匹配的分词方法、基于理解的分词方法和基于统计的分词方法。按照是否与词性标注过程相结合，又可以分为单纯分词方法和分词与标注相结合的一体化方法。

(1) 基于字符串匹配的分词方法[44]。这种方法又叫做机械分词方法，它是按照一定的策略将待分析的汉字串与一个“充分大的”机器词典中的词条进行配，若在词典中找到某个字符串，则匹配成功。常用的几种机械分词方法如下：正向最大匹配法（由左到右的方向）；逆向最大匹配法（由右到左的方向）；最少切分（使每一句中切出的词数最小）；双向最大匹配法（进行由左到右、由右到左两次扫描）。

(2) 基于理解分词的方法[45]。这种分词方法是通过让计算机模拟人对句子的理解，达到识别词的效果。其基本思想就是在分词的同时进行句法、语义分析，利用句法信息和语义信息来处理歧义现象。它通常包括三个部分：分词子系统、句法语义子系统、总控部分。在总控部分的协调下，分词子系统可以获得有关词、句子等的句法和语义信息来对分词歧义进行判断，即它模拟了人对句子的理解过程。这种分词方法需要使用大量的语言知识和信息。由于汉语语言知识的笼统、复杂性，难以将各种语言信息组织成机器可直接读取的形式，因此目前基于理解的分词系统还处在试验阶段。

(3) 基于统计的分词方法[46]。词是稳定的字的组合，因此在上下文中，相邻的字同时出现的次数越多，就越有可能构成一个词。因此字与字相邻共现的频率或概率能够较好的反映成词的可信度。可以对语料中相邻共现的各个字的组合的频度进行统计，计算它们的互现信息。定义两个字的互现信息，计算两个汉字 X、Y 的相邻共现概率。互现信息体现了汉字之间结合关系的紧密程度。当紧密程度高于某一个阈值时，便可认为此字组可能构成了一个词。这种方法只需对语料中的字组频度进行统计，不需要切分词典，因而又叫做无词典分词法或统计取词方法。

目前，中文分词的难点主要有三点：

(1) 对于词和词组的模糊判别。现代汉语的基本表达单元虽然为“词”，且以双字或者多字词居多，但由于人们认识水平的不同，对词和短语的边界很难去区分。例如：“对随地吐痰者给予处罚”，“随地吐痰者”本身是一个词还是一个短语，不同的人会有不同的标准，同样的“海上”“酒厂”等等，即使是同一个人也可能做出不同判断，如果汉语真的要分词书写，必然会出现混乱，难度很大。

(2) 词与词之间的歧异判别。歧义是指同样的一句话，可能有两种或者更多的切分方法。主要的歧义有两种：交集型歧义和组合型歧义，例如：表面的，因为“表面”和“面的”都是词，那么这个短语就可以分成“表面 的”和“表 面的”。这种称为交集型歧义（交叉歧义）。像这种交集型歧义十分常见，前面举的“和服”的例子，其实就是因为交集型歧义引起的错误。“化妆和服装”可以分成“化妆 和 服装”或者“化妆 和服 装”。由于没有人的知识去理解，计算机很难知道到底哪个方案正确。

(3) 未登录词识别。命名实体（人名、地名）、新词，专业术语称为未登录词。也就是那些在分词词典中没有收录，但又确实能称为词的那些词。最典型的是人名，人可以很容易理解。句子“王军虎去广州了”中，“王军虎”是个词，因为是一个人的名字，但要是让计算机去识别就困难了。如果把“王军虎”做为一个词收录到字典中去，全世界有那么多名字，而且每时每刻都有新增的人名，收录这些人名本身就是一项既不划算又巨大的工程。

常用的分词工具主要有 IKAnalyzer、Paoding、Mmseg、Imdict、ICTCLAS、Phpanalysis、SCWS 等。其中应有最广泛的是中科院计算机所所研制的中文分词系统ICTCLAS,它基于N最短路径算法采用HMM模型进行分词,分词速度达到996KB/s，分词精度 98.45%，支持 Linux、FreeBSD 及 Windows 系列操作系统，支持 C/C++、C#、Delphi、Java 等主流的开发语言。

### 3.1.3 去停用词

在对文本分词之后，得到的词串并不都是有用的，需要自动过滤掉某些字或词，这些字或词即被称为 Stop Words(停用词)。这些停用词一般都是人工输入，生成后的停用词会形成一个停用词表。但是，并没有一个明确的停用词表能够适用于所有的工具。甚至有一些工具是明确地避免使用停用词来支持短语搜索的。对于一个给定的目的，任何一类的词语都可以被选作停用词。通常意义上，停用词大致分为两类。一类是人类语言中包含的功能词，这些功能词极其普遍，与其他词相比，功能

词没有什么实际含义，比如'the'、'is'、'at'、'which'、'on'和中文中的‘这’、‘那’、‘的’、‘得’、‘地’等。另一类词包括词汇词，比如'want'和‘后’，‘应’，‘到’，‘某’，‘后’，‘个’等。这些词应用十分广泛，但又无明显的类别意义。如果不将其去除很容易被选为特征词影响分类准确率。

目前，对于文本分类领域，停用词表的研究还比较少，G.W.art 在研究中发现英文段落中所有词的 50%可以包含在一个具有 135 个词的普通表中[47]，Van Rijsbergen 在[48]中认为这些词应该被视为噪声词，并且应该在文本分析中去除，这就构成了最早的停用词表。中文停用词表的研究起步较晚，顾益军在[49]中提出用信息熵的方法来代替简单的统计词频和文档频率，对分类效果有了较为明显的改善。目前，英文停用词表较为著名的是 Van Rijsbergen 发表的停用词表和 Brown Corps 停用词表，中文方面尚没有统一的停用词表。

## 3.2 特征提取

文档经过预处理后，由于每篇文章的长度、关键词都不相同无法拿来作比较，并不能直接拿来做分类。文本的特征选择是文本分类过程中的一个关键技术。它通过对文章进行统计分析来获得最能代表该篇文章的若干关键词，并以这些关键词代替该文章，这一过程成为文本分类的特征提取，它是一种应用于文本的特征降维方法。文本的特征选择过程大致可以分为四个部分[50]：产生文本特征子集、文本特征子集评估、终止条件和最优文本特征子集。如下图：

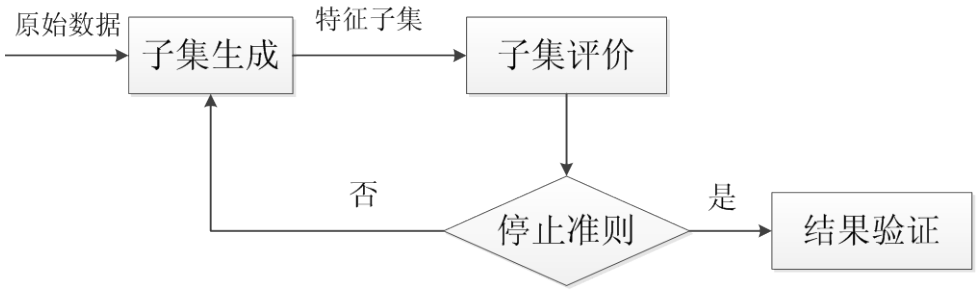


图 3-1 特征选择基本框架

Fig.3-1 Basic framework of feature selection

特征选择算法根据评价函数来分可分成四类：距离度量、信息度量、依赖性度量和一致性度量。

### 3.2.1 根据距离度量的特征提取算法

距离度量通常被认识为分离性、差异性或者辨识能力的度量。最为常用的距离测度有欧氏距离、*S* 阶 Minkowski 测度、Chebychev 距离、平方距离等。二分类问题中对于特征 X 和特征 Y,如果由 X 引起的两类条件概率差异大于 Y，则 X 优于 Y。因为特征选择的目的是找到使两类尽可能分离的特征。如果差异为 0，则 X 和 Y 是不可区分的，这一类算法中较为知名的是 Relief 算法。

Relief 算法[51]是由 Kira 和 Rendell 提出的一种权值搜索特征子集算法，它为每一个特征赋予一个权值，以权值表征特征与类别的相关性。定义假设间隔(hypothesis-margin)为在保持样本分类不变的情况下决策面能够移动的最大距离，可以表示为：

$$\theta = \frac{1}{2}(\|d - M(d)\| - \|d - H(d)\|) \tag{3-4}$$

其中 $M(d)$、$H(d)$ 分别为与 $d$ 同类和非同类的最近邻点。通过计算训练样本的假设间隔，可以近似地对特征进行关于分类价值的评价。Relief 算法正是利用这一特点给特征集中的每一特征赋予一定的权重，具体思想如下：

假设 $D = \{d_1, d_2, \cdots, d_n\}$ 是待分类的文本集，$d_i = \{x_1, x_2, \cdots, x_n\}$ 对应其第 i 篇文档，$x_j$ 表示第 j 维，即第 j 个特征值。Relief 算法从所有训练集中随机选择 m 个样本，计算样本在各个特征上的假设间隔，并累加起来作为该特征的权值，样本 d 更新特征 $x_i$ 的权值可以表示为：

$$W_{x_i}^{i+1} = W_{x_i}^{i} - diff(x_i, d, H(d))/m + diff(x_i, d, M(d))/m \tag{3-5}$$

由于文本特征为离散值，函数 $diff$ 定义为：

$$diff(x_i, d, d') = \begin{cases} 0, & d_{x_i} = d'_{x_i} \\ 1, & d_{x_i} \neq d'_{x_i} \end{cases} \tag{3-6}$$

从公式 3-6 可以看出当特征对分类贡献度更大时，意味着同类样本间距离较近，而非同类样本间距离相距较远，权值相应较大；相反，如果特征对分类贡献度较小时，权值经过大量样本计算应较小甚至趋于零。

### 3.2.2 基于信息度量的特征提取算法

信息度量通常采用信息增益(Information Gain, IG)[52]、互信息(Mutual Information, MI)[53]、CHI 统计[54]和期望交叉熵(Expected Cross Entropy, ECE)[55]来衡量。信息增益定义为先验不确定性与期望的后验不确定性之间的差异，它能够有效地选出关键特征，剔除无关特征,互信息描述的是两个随机变量之间的相互依存关系的强弱,CHI 统计描述特征与类别之间的独立性，期望交叉熵与信息增益类似，不同于信息增益的地方是不考虑单词在某一文档中未发生的情况。

(1) 信息增益

信息增益是机器学习中的概念，被用来计算决策树中特征的权值，它定义为类特征向量的平均值。该特征为分类所能提供的所有信息以不考虑该特征的熵和考虑该特征后的熵的差值表示。信息熵的定义如下：

$$E(C)=\sum_{i=1}^{n}P(c_i)\log P(c_i) \tag{3-7}$$

IG 的计算公式为：

$$IG(x)=E(C)-E(C|X)=-\sum_{i=1}^{n}P(c_i)\log P(c_i)+P(x)\sum_{i=1}^{n}P(c_i|x)\log P(c_i|x)+p(\bar{x})\sum_{i=1}^{n}P(c_i|\bar{x})\log P(c_i|\bar{x}) \tag{3-8}$$

$P(c_i)$表示类别$c_i$在训练集中出现的概率；$P(x)$为特征项 x 出现的概率，即训练集中包含 x 的文档的概率；而$p(\bar{x})$)表示训练集中不包含特征 x 的文档的概率;$P(c_i|x)$表示出现特征 x 时为类别$c_i$的概率，即类别$c_i$中出现特征 x 的文档数除以训练集中出现特征 x 的总文档数；$P(c_i|\bar{x})$为类别$c_i$中没有出现特征 t 的文档数除以训练集中没有出现特征 x 的文档数。

信息增益通过统计某个特征值在某篇文章中出现或者不出现的次数来预测文档的类别。某个特征的信息增益越大，贡献越大，对对分类贡献也就越大。信息增益的不足之处它考虑了特征未发生的情况，对判断文本类别贡献不大，引入了不必要额干扰，特别是在类别分布和特征值分布高度不均衡的情况下，若绝大多数类为负类，绝大多数的特征都不出现，函数结果由不出现的特征值决定，此时，信息增益算法的效果会大打折扣。

(2) 互信息

互信息是信息论中的概念，它用来度量一个信息中两个信号之间的相互依赖

程度。在特征选择领域中人们经常利用它来计算特征 $x$ 与类别 $c$ 之间的依赖程度，将特征 $x$ 与各个类别的互信息相互融合起来作为特征值的权重。特征 x 和类别 c 之间的互信息 $MI(t,c)$ 定义如下：

$$MI(x,c)=\log\frac{P(x|c)}{P(x)}=\log\frac{P(x,c)}{P(x)P(c)} \tag{3-9}$$

其近似计算公式为：

$$MI(x,c)\approx\log\frac{A\times N}{(A+C)(A+B)} \tag{3-10}$$

其中：A 为特征 x 和文档 $c$ 类同时出现的次数；B 为特征 x 出现而 c 类文档不出现的次数；C 为 $c$ 类文档出现而特征 x 不出现的次数；N 为文档总数。

当 x 和 $c$ 无关时，也就是说 x 和 $c$ 在分布上式独立的，那么 $MI(x,c)=0$；当特征项的出现依赖于某个类别时，与该类别的 MI 就会很大；当特征很少出现在某个类别时，它们的互信息为负数，即负相关。当存在多个类别时，可利用下式计算 x 对于 $c$ 的互信息：

$$MI_{AVR}(x,c)=\sum_{i=1}^{m}P(c_i)MI(x,c_i) \tag{3-11}$$

互信息在进行特征选择时更偏向于考虑稀有特征词，但是，由于稀有词所包含的类别信息可能会少于较常出现的特征词，这使得使用互信息法选取特征集合进行文本分类时的分类性能可能会较差。

(3) $\chi^2$ 统计量

$\chi^2$ (CHI-square)统计是用来统计特征项和类别之间独立性的缺乏程度的量，也就是说 $\chi^2$ 越大，两者之间独立性越小，相关性越大。它假设特征项 x 和类别 c 之间符合一阶自由度的 $\chi^2$ 分布，原假设为 x 和 c 无关，那么计算出来的 CHI 值越高，越应该选择备择假设，即相关性越大，类别信息量也越大。，特征项 x 对于类别 c 的 CHI 值，可由下式计算：

$$\chi^2 = \frac{N(AD-BC)^2}{(A+C)(B+D)(A+B)(C+D)} \tag{3-12}$$

其中，A 是特征 x 和类别 c 的共现频数，B 是在文本集中出现包含特征 x，且该文档不属于 c 类的文档数。C 是 c 类文档中不包含特征 x 的文档数，D 是文档中既不包含特征 x 也不属于类别 c 的文档数，N 为总文档数。$\chi^2$ 统计的优点是同时考虑了特征项存在与不存在的情况，所以在信息表征性上要优于互信息，但是，由于它统计文档中是否出现特征 x，并不重视其词频，这样就夸大了低频词的作用，对低频词分类效果并不明显。

(4) 期望交叉熵(Expected Cross Entropy, ECE)

期望交叉熵反映了文本类别的概率分布以及在出现某种特定情况下文本类别概率之间的距离。特征 x 的期望交叉熵越大，对文本分类别分布的影响就越大。

期望交叉熵计算公式如下：

$$ECE(x) = P(x)\sum_i P(c_i \mid x)\log\frac{P(c_i \mid x)}{P(c_i)} \tag{3-13}$$

其中，$P(c_i \mid x)$ 表示文本中出现词 x 时，文本属于类别 $c_i$ 的概率； $P(c_i)$ 是类别 $c_i$ 出现的概率； $P(c_i \mid x)$ 值较大时，类别与特征就是强相关的，若此时 $P(c_i)$ 较小时，那么该特征就对分类作用较大。但是，ECE 的缺点是它没有考虑文本特征没有出现的情况。

### 3.2.3 基于依赖性度量的特征提取算法

依赖性度量中 Hibert-Schmidt(HISC)依赖性准则经常被作为一个评价度量特征与类别相关性的关键方法，它的核心思想是一个好的特征应该最大化这个相关性。据此，特征选择问题可以看成组合最优化问题：

$$T_0 = \arg\ \max J(\mathrm{x}),\ \text{s.t.}\left|X\right| \le n \tag{3-14}$$

其中，n 为所选特征数上限，X 为已选特征的集合， $J(\mathrm{x})$ 为评价准则。这一类特征提取算法还经常用统计相关的系数，如 Pearson 相关系数、概率误差、Fisher 分数、

线性可判定分析、最小平方回归误差等来表达特征对于类别的可分离性。

### 3.2.4 基于一致性度量的特征提取算法

给定两个文本样本，如果他们的特征值均相同，但所属类别不相同，则称他们是不一致的；否则，称他们是一致的。一致性度量采用不一致性来度量，它的目的不是最大化类的可分离性，而是试图保留原始特征的辨识能力，即找到与全集有相同类别区分能力的最小子集，它具有单调、快速、去冗余特征等优点。但是，它对噪声数据敏感。这方面的典型算法有 LVF 、Focus 等。

LVF 算法[56]是一种使用概率的方法来进行特征选择的方法，它首先随机选择一个特征子集，然后用一致性判别准则来评价，实现特征选择。LVF 算法的核心是不一致准则，在具体应用时通过人工设定一个合适的不一致率 $\delta$，使得判别函数：$J(\mathrm{x}) \le \delta$ 成立。

LVF 算法具体流程如下：

（1） 在初始阶段指定一个最大选择次数 K 和不一致率 $\delta$，设定初始子集 X；

（2） 在 X 中随机选择候选自己 $X^{'}$；

（3） 若候选子集 $X^{'}$ 包含的特征数小于 X 所含的特征数则下一步，否则返回（2）。；

（4） 计算不一致率，若 $J(\mathrm{x}) \le \delta$，则替换最佳子集；

（5） 判断循环次数是否达到阈值，若达到则结束，否则返回（2）；

LVF 算法每一轮循环中都要从特征集 X 中随机选一个子集，并且判断是否满足不一致条件，它通过随机产生候选集的方法并没有遍历所有的特征，只能通过增加迭代次数来减小误差率。

### 3.2.5 小结

文本数据通常长度不同，维数巨大，在不进行特征提取的情况下很难进行文本分析处理，上述介绍的是文本分类中较常用的一类基于过滤（Filter）的方法，这一类方法通常以设定一种度量方式的方法来评判特征子集的优劣，另一类方法是用目前较流行的启发式学习算法，以分类准确率作为评判指标的方法，称之为 Wrapper 方法。常见的蜂群算法、基因算法、模拟退火算法都属于这一类，这一类算法通常效

率较低，在文本分类中由于分类过程复杂，数据维数高，采用这种方法往往得不偿失，因此在实际应用中通常较少使用。

## 3.3 文本的向量表示

文本要想被计算机所理解，要对其进行向量化，以数字的形式表示。这在本质上就是一个由非结构化的自然语言文本转换为结构化的向量文本的过程。在信息检索的概念被提出后，出现了一系列基于文档和查询之间的文本表示模型，具有代表性的有布尔模型、向量空间模型、概率模型等。这些模型分别从不同的角度出发，使用不同的方法处理特征加权和相似度计算等问题。

### 3.3.1 布尔模型

布尔模型(Boolean model, BM),是基于集合论与布尔代数智商的一种简单模型，主要应用于文本分析中。在布尔模型中文档 $d_i$ 中特征 $x_j$ 的权重 $w_{i,j}$，是二值的，即 $w_{i,j} \in \{0,1\}$。一篇文章被表示成文本集中出现的特征的集合，即一个特征空间中的一个向量，这个向量中，每个分量的权值为 0 或者 1。即文档 $d_i=(w_{i1},w_{i2}\cdots w_{in})$，这种表示方法由于过于简单并不能体现文章中不同特征词汇的区别，也不能体现文章上下文之间的联系，近几年仅在信息检索领域有了一定的发展。

### 3.3.2 向量空间模型

向量空间模型(Vector space model, VSM)以向量的形式来表示文档，目前，该模型是文本表示的主要模型。经典的向量空间模型有 Salton 等人于上世纪六十年代末提出，并且成功应用于著名的 SMART 系统，随着信息的膨胀，它被广泛的应用于信息检索、自动索引、文本分类。向量空间模型的一个基本假设是：文本所述的类别仅与该词条在该文档中出现的频数有关，而与这些单词或词组在该文本中出现的位置和顺序无关。它的一个基本思想是用文本的词袋(Bag of words, BOW)表示文本，将每个词条作为特征空间的一维，将文本看成特征空间中的一个向量，再以两个向量的模或者夹角来衡量两个文本之间的相似度。

在向量空间模型中比较重要的概念有：特征项、文本、特征项的权重、文本的相似度等。

文本(Document):泛指一般的文本或者文本的片段。一般为经过预处理后的纯文本，在下文中统一以文本代称纯文本对象，对于文档与文本不加以区别。

特征项(Term):文本的内容特之呢过常常用它所含有的基本语言单位(字、词、词组或短语等)来表示，即 $d=\{x_1,x_2\cdots x_n\}$，这些基本的语言单位被统称为文本的特征项。

特征项的权重(Term Weight):对于含有 n 个项的文本 $d=\{x_1,x_2\cdots x_n\}$,这 n 个特征项是经过特征提取降维之后的，这些特征项常常被赋予一定的权重 w 来表示他们在文本 d 中的重要程度，即 $d=\{x_1,w_1;x_2,w_2\cdots x_n,w_n\}$，简记为 $d=\{w_1,w_2\cdots w_n\}$。

文本相似度(Similarity):两个文本 $d_1$ 和 $d_2$ 之间的相关程度(Degree of Relevance)常常用他们之间的相似度 $Sim(d_1,d_2)$ 来衡量。当文本被表示为向量空间模型时，可以借助向量之间的距离表示他们之间的相似程度，目前经常被使用的相似度计算公式有：

（1）向量间内积

$$Sim(d_1,d_2)=\sum_{k=1}^{n} w_{1k}\cdot w_{2k} \tag{3-15}$$

两个向量间的内积越大，其相似度则越大。

（2）向量夹角余弦

$$Sim(d_1,d_2)=\frac{\sum_{k=1}^{n} w_{1k}\cdot w_{2k}}{\sqrt{\left(\sum_{k=1}^{n} w_{1k}^2\right)\left(\sum_{k=1}^{n} w_{2k}^2\right)}} \tag{3-16}$$

向量空间模型的特点是其忽略了特征项之间的相互顺序关系，这样也因此带来了表示、计算和处理上的优势，但是这样也不可避免的损失了大量关于文本结构和语义的信息。

### 3.3.3 特征项的权重计算

特征项的权重计算是向量空间模型的最重要组成部分，在给每个特征项赋予权重时，应该使文本中越能代表类别信息的特征项权重越大，但是应该避免当一个强的特征出现时把其它特征完全被其淹没，这就会造成一种信息的异常放大。计算特征项权重的方法最简单的是布尔权重，即特征项出现为 1 否则为 0，该方法特点是简单

易用，但是无法体现每个特征项在文档中的重要程度。另一种方法是由专家或者用户根据自己的经验与所掌握的领域知识，人为赋予权重，这种方法随意性比较强，而且效率也低，但是随着文本信息维度的增大，文本内容多样化，这种方法已经应用较少，但是在特种领域还有着应用。第三类方法是把特征项和所有类别中其他特征，文本进行比较从而得出其自身的权重，这类方法是目前研究和应用最多的一类方法，其代表就是 TF-IDF 算法。

所谓的 TF-IDF 算法是建立在这样一个假设之上的：对区别文档最有意义的词语应该是那些在文档中出现频率高，而在整个文档集合的其他文档中出现频率少的词语，所以如果特征空间坐标系取 TF (Term Frequency)词频作为测度，就可以体现同类文本的特点。另外考虑到单词区别不同类别的能力，TF-IDF 法认为一个单词出现的文本频数越小，它区别不同类别文本的能力就越大。因此引入了逆文本频度 IDF (Inverse Document Frequency)的概念，以 TF 和 IDF 的乘积作为特征空间坐标系的取值测度，并用它完成对权值 TF 的调整，调整权值的目的在于突出重要单词，抑制次要单词。对于在某一特定文件里的词语 $x_i$ 来说，它的重要性可表示为：

$$\mathrm{tf}_{i,j}=\frac{n_{i,j}}{\sum_k n_{k,j}} \tag{3-17}$$

上式中，$n_{i,j}$ 表示该特征词 $x_i$ 在文本 $d_j$ 中出现的次数分母为所有特征词数目之和。

逆向文件频率是一个词语普遍重要性的度量。某一特征词的 IDF 可以由总问本数目除以包含该词语的文本数目之和再将得到的商取对数，如下式：

$$idf_i=\log\frac{\left|\mathrm{D}\right|}{\left|\{j:x_i\in d_j\}\right|} \tag{3-18}$$

其中，$\left|\mathrm{D}\right|$ 为语料库中文本总数目，$\left|\{j:x_i\in d_j\}\right|$ 表示包含特征词 $x_i$ 的文本数目总数。但是，由于在某一篇文章中经过特征提取后可能会出现某词的出现频数为 0 的情况，因此在一般的情况下分母使用 $1+\left|\{j:x_i\in d_j\}\right|$ 代替。所以 TF-IDF 权重公式为：

$$W_{i,j}=tf_{i,j}\square idf_i \tag{3-19}$$

通常为了消除文档中权值相差过大和其他一些因素的影响，往往要对向量进行归一化处理，如下式：

$$W_{i,j}=\frac{tf_{i,j}\square idf_i}{\sqrt{\sum_{j=1}^{n}\left(tf_{i,j}\right)^2\square idf_i^2}} \tag{3-20}$$

在本质上 TF-IDF 是一种试图抑制噪声的加权 ，并且单纯地认为文本频率小的单词就越重要，文本频率大的单词就越无用，显然这并不是完全正确的。而且 TF-IDF 的简单结构并不能有效地反映单词的重要程度和特征词的分布情况，使其无法很好地完成对权值调整的功能，所以 TF-IDF 法的精度并不是很高。此外，在 TFIDF 算法中并没有体现出单词的位置信息。

## 3.4 分类算法

在经过上述步骤处理之后，文本数据本转换为向量数据，使得计算机可以对其进行处理，这就为后面的文本分类做了很好的基础。文本分类算法源于模式分类主要包括两种，一种是有监督的分类，即训练数据是有标签的，另一种是训练数据无标签的无监督分类也叫做文本聚类。本文的研究重点是有监督分类。

目前国内外比较流行的有监督分类算法，基本上可以分为三类，第一是基于统计的方法，如 KNN、朴素贝叶斯、回归模型、支持向量机、最大熵模型等，这一类方法目前应用最广，效果最好；另一种是基于连接的方法，如人工神经网等；最后一种是基于规则的方法，如决策树、关联规则等。最后两种方法在最近几年发展较慢，应用较少，所以下面简单较少几种基于统计的分类算法。

### 3.4.1 K 近邻(K-Nearest Neighbor)分类算法

K 近邻(K-Nearest Neighbor,KNN)算法，是由 Cover 和 Hart 在 1968 年提出的[57]，是最近邻算法的一个推广。基于最近邻的思想，取测试样本的 K 个最近邻，使用投票的原则作为分类决策，即 K 个最近邻中，多数样本的类别就是待测样本的类别。K 近邻算法是一种非参数的分类算法，具有简单、直观易用、易于实现等特点被广泛应用与分类、回归等模式识别领域中。

KNN 算法的实质是记忆知识，供需要时使用，不需要计算和推理，是一种最简单的无模型机器学习算法。K 近邻算法没有训练过程，只在测试时通过计算待测样本与训练样本之间的相似度，选择 K 个相似对最高的样本点，然后通过投票规则来判断这个测试点的类别。所以 K 近邻分类的两个重要问题是，一是相似度的计算，另一种是投票规则，即权重计算。对于文档集 D 中的每个文本向量，K 近邻分类的

目标是找到测试文本 x 的 K 个最相似近邻，这个相似度一般采用余弦相似度来衡量，判别公式如下：

$$p(x,c_i)=\sum_{x_j\in KNN} Sim(x,x_j)\sigma\left(x_j,c_i\right) \tag{3-21}$$

其中，$Sim(x,x_j)$ 表示测试点 $x$ 和它的 K 近邻点 $x_j$ 的余弦相似度，用来判断 $x_j$ 是否属于类别 $c_i$ 若是则为 1，否则为 0，即：

$$\sigma\left(x_j,c_i\right)=\begin{cases}1, x_j\in c_i\\ 0, x_j\notin c_i\end{cases} \tag{3-22}$$

最后将测试文本归入到结果最大的那一类中。

### 3.4.2 朴素贝叶斯（NaiveBayes）分类算法

朴素贝叶斯(NaiveBayes，NB)分类算法[58]是以贝叶斯原理为基础，假设在给定的文档语境下，文本特征是相互独立的。假设 $d_i$ 为任一文档，它属于类别 $c_j$，根据贝叶斯原理有：

$$P(c_j\mid d_i)=\frac{P(c_j)P(d_i\mid c_j)}{P(d_i)} \tag{3-23}$$

$$P(d_i)=\sum_{j}^{n}P(c_j)P(d_i\mid c_j) \tag{3-24}$$

对于测试文本 $d_i$，按上式计算 $P(c_j\mid d_i)$，分类规则为概率值最大的那个类就认为是 $d_i$ 的类别：

$$d_i\in c_j \quad if \quad P(c_j\mid d_i)=\max_{i=1}^{n}\{P(c_j\mid d_i)\} \tag{3-25}$$

根据 $P(c_j\mid d_i)$ 计算方式的不同，可以将朴素贝叶斯方法分为最大似然模型、多项式模型和泊松模型等。

### 3.4.3 支持向量机（Support Vector Machine）

支持向量机(Support vector machine, SVM)是由 Vapnik 在 1995 年提出的基于统计学习理论的一种机器学习算法[59]，它基于结构风险最小化原理，将原始数据压缩到

只有原数据 3%-5%大小的支持向量集中，以这些支持向量构成的超平面作为分类决策面。支持向量机在解决小样本、非线性以及高位模式识别问题中表现出了良好的性能。SVM 的基本思想是寻找一类样本点时的不同类别之间由这些支持向量构成的超平面之间分开距离最大。对于一个二分类问题，分类线性方程为 $x \cdot w + b = 0$，对其进行归一化处理，使得对线性可分的样本集 $(x_i, y_i)$，满足：

$$y_i((x \cdot w) + b) \geq 1 \tag{3-26}$$

此时分类间隔等于 $2/\|w\|$，满足公式 3-26 并且使得 $\|w\|^2$ 最小的平面，即为所求的最优分类面。构成这个超平面的向量即为支持向量。

## 3.5 分类性能评估

文本分类中一般使用信息检索中的查全率和查准率这些指标来衡量分类系统的性能[60]。文本分类根据文章类别之间的映射是否单射可以分为两类，第一类称为单类赋值，即每个文档只属于某一单一类别。第二类称为多类排序，即每个文档可以属于不同的类别，分类结果是给每个文档所属的类别打分并排序，排在最前面的类说明文档属于该类的可能性最大。

### 3.5.1 单类赋值

文本分类中最常使用的评估指标有查全率(Recall, R)、查准率(Precision, P)。对于一个文本的每一类别采用列联表来计算查全率和查准率。

表 3-1 单类赋值列联表

Table 3-1 Single Category Assignment Contingency Table

| | 真正属于该类的文档数 | 真正不属于该类的文档数 |
|---|---|---|
| 判断为属于该类的文档数 | A | C |
| 判断为不属于该类的文档数 | B | D |

此时，R 和 P 定义为：$\mathrm{R} = \frac{A}{A+C}$ 和 $\mathrm{P} = \frac{A}{A+B}$，对于一个分类系统来说 R 值和 P 值是相互影响的提高 R 必然会降低 P,因此为了更全面地反映系统的分类性能，一种将查全率和查准率结合起来的性能评估方法 F1 测量值经常被用于评估文本分类性能。其计算公式为：$F1 = \frac{P \times R \times 2}{P + R}$，当 P=R 时成为平衡点。

### 3.5.2 多类排序

对于这类问题，首先应给定一个阈值，以确定文档和类别之间的所属关系。对于每一个输入的测试文档都会返回一个排序后的文档列表。这时两个评价指标定义为：

$$P = \frac{\text{找到的该文档所属的正确类别数}}{\text{判断为该文档所属类的类别数}} \tag{3-27}$$

$$R = \frac{\text{找到的该文档所属的正确类别数}}{\text{该文档所属的所有类别数}} \tag{3-28}$$

## 3.6 文本分类系统实现

根据上述步骤本节设计实现了文本分类系统具体流程如下：

（1） 首先，对中文文本文本进行分词、去停用词等预处理。（本文采用的时张华平博士所共享的免费分词工具 ICTCLAS2011）

（2） 对文本进行特征选择，本文选用了 IG 这种常用的的特征提取算法来对文本进行特征提取。

（3） 构造向量空间模型(Vector Space Model，VSM)，本文所采用的是经典 TF*IDF 法。

（4） 采用 K 近邻分类算法进行分类。

（5） 分类评价。本文分别采用了查准率、查全率和 F1 值来对分类效果进行评估。

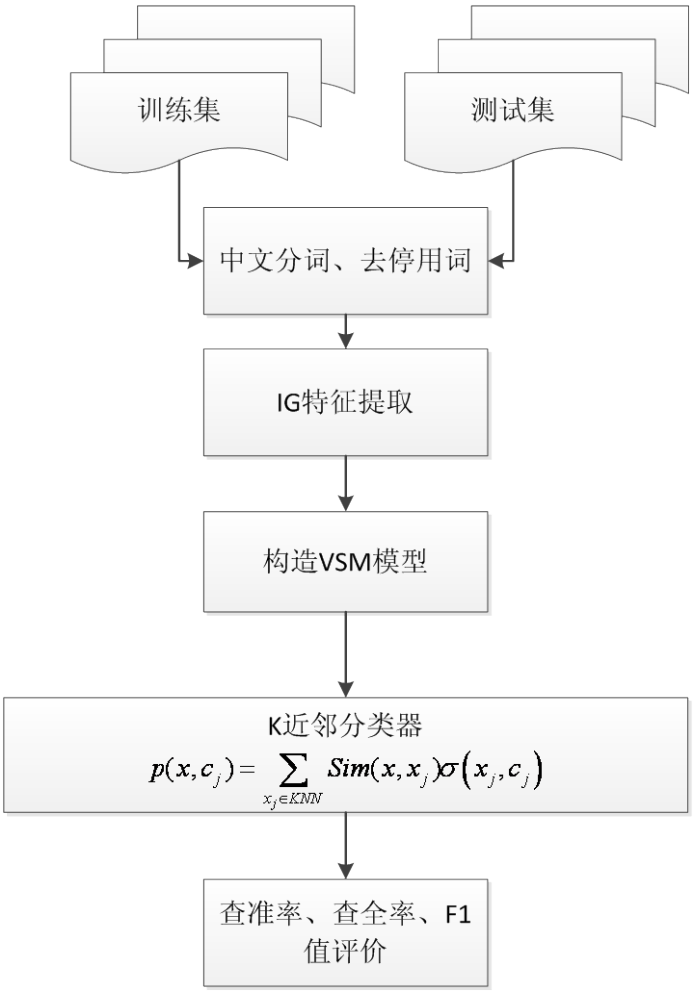


图 3-2 文本分类程序流程设计

Fig.3-2 Design of text classification program

### 3.6.1 系统程序设计

本系统采用 C++开发，包含预处理模块，特征提取模块、分类模块、评价模块四部分。首先对文本进行结构化处理，去除字符串首尾空白、宽窄字符转换：

```
void trim(string  &str,const string val);
wstring MultibyteToWideChar(string sResult);
```

然后根据北京理工大学张华平所提供的免费分词工具 ICTCLAS2011 进行对文章进行分词，其功能包含，添加词性，用户词典，文章，段，句子分词等，用户接口包含：

```
bool ICTCLAS_Init(const char * sInitDirPath=0, int encoding=GBK_CODE);
```

初始化分词工具；

```
unsigned int ICTCLAS_ImportUserDict(const char *sFilename);
```

导入用户词典；

```
const char * ICTCLAS_ParagraphProcess(const char *sParagraph,int bPOStagged=1);
```

段处理；

```
bool ICTCLAS_FileProcess(const char *sSourceFilename,const char *sResultFilename,int bPOStagged=1);
```

文章处理；

```
int ICTCLAS_AddUserWord(const char *sWord);
```

手工增加词性标注；

```
bool ICTCLAS_Exit();
```

退出释放内存。

第二步进行造停用词表、建立词袋子、计算特征词汇联排表、特征提取、构造VSM 模型：

```
set<string>MakeStopSet();
int ConstructDictionary(DICTIONARY& mymap,FUNCSEG seg,string tablename);
void GetContingencyTable(CONTINGENCY& contingencyTable,char *address);
vector<pair<string,double> >InformationGainFeatureSelectionForclassify(DICTIONARY&
mymap,CONTINGENCY& contingencyTable,string classLabel);
int VSMConstruction(DICTIONARY& mymap,DOCMATRIX& traingsetVSM,char* keywordsaddress);
```

最后，对特征向量规范化、进行KNN分类、计算查准率、查全率以及F1值:

```
vector<pair<int,double> > NormalizationVSM(vector<pair<int,double> > tempVSM);
void KNNclassifier(string tablename,DOCMATRIX& trainingsetVSM,DOCMATRIX&
testingsetVSM,vector<string>catigorization,int N,RESULTINFO& classifyResults);
double getPrecision(string classLabel,RESULTINFO classifyResults,string tablename);
double getRecall(string classLabel,RESULTINFO classifyResults,string tablename);
double getFscore(string classLabel,RESULTINFO classifyResults,string tablename);
```

### 3.6.2 实验结果

本文所采用的语料库是搜狗实验室所提供的新闻语料库，我们选取其中的 5 类，包括文化 2140 篇，教育 150 篇，娱乐 800 篇，历史 1300 篇，军事 1400 篇。按照 70%作为训练集，30%作为测试集分为两份。选取的特征数为 50，评价标准采用查准率、查全率和 F1 值。结果如下：

表 3-2 KNN 文本分类实验结果（K=26 时不同类别评价）

Table 3-2 The result of text classification using KNN(K=26)

| 类别 | 查全率 | 查准率 | F1 值 |
|---|---|---|---|
| 文化 | 0.721 | 0.799 | 0.757 |
| 教育 | 0.634 | 0.723 | 0.676 |
| 娱乐 | 0.715 | 0.784 | 0.747 |
| 历史 | 0.742 | 0.773 | 0.757 |
| 军事 | 0.733 | 0.780 | 0.755 |

表 3-3 KNN 文本分类实验结果（不同 K 值下的 F1 值）

Table 3-3 The result of text classification using KNN(Use Sougou's corpus)

| K 值 | 2 | 4 | 6 | 8 | 10 | 12 | 14 |
|---|---|---|---|---|---|---|---|
| F1 值 | 0.748 | 0.753 | 0.759 | 0.778 | 0.764 | 0.756 | 0.748 |
| K 值 | 16 | 18 | 20 | 22 | 24 | 26 | 28 |
| F1 值 | 0.752 | 0.754 | 0.750 | 0.761 | 0.753 | 0.768 | 0.746 |

从上述两表可以看出 K 近邻分类器在同一 K 值下对不同类别文档的分类性能并不相同，在不同 K 值的情况下随着 K 值的变化，分类性能也成抛物线形状，随着 K 值增加先升高后下降，这说明 K 近邻分类器的性能与文本特征和 K 值的选择有关。

## 3.7 本章小结

本章介绍了文本分类的具体步骤及流程，并对一些重点算法的原理进行了介绍，包括分词算法，文本的表示模型，特征提取算法，典型的分类算法以及性能评价方法。最后使用 C++实现了上述文本分类系统，并对 K 近邻文本分类算法做了分析。

# 第四章　基于大边界最近邻算法的文本分类

本章将距离度量学习中的大边界最近邻算法(LMNN)和文本分类相结合，研究了将距离度量学习引入到文本分类可能会出现的问题并且提出了改进方案，最后经过试验仿真验证算法的可行性。

## 4.1　背景和初衷

目前，机器学习中的 K 近邻(KNN)分类算法和支持向量机(SVM)被认为是处理文本分类的最好方法[63]，其中 K 近邻分类的优点有：第一，KNN 分类算法可以在类不平衡的情况下发挥作用，这是由于 K 近邻算法在执行分类的过程时只与目标样本周围的少数样本作比较，与类别整体的数量多少没有太大关系。第二，KNN 算法没有训练过程，是一种无参数的机器学习算法，它简单、易用而且效果特别好，是目前使用较为广泛的一种分类算法。但是，KNN 算法的缺点也很明显：第一，KNN 分类算法是基于近邻度量的一种模式分类算法，它高度依赖于数据间的相似度度量，简单的欧式距离度量在实际应用时，由于不考虑不同维度之间对分类的影响和输入数据维数越来越高往往不能取得良好的分类效果。第二，KNN 分类算法虽然可以一定情况下克服类偏斜带来的分类误差，但是这也是造成它对样本密度分布敏感的主要原因，当类间密度高度分布不均时分类效果也会有较大的影响。

由上述可知，在文本分类领域要想提高 KNN 分类的准确率，首先，要解决的就是距离度量的问题，而上文中提到的距离度量学习算法恰好是满足这样要求的一类算法。其中的大边界最近邻算法(Large Margin Nearest Neighbor, LMNN)和近邻元分析(Neighborhood Component Analysis, NCA)等是一类专门改进 K 近邻分类算法的距离度量算法。实验证明 LMNN 算法在大多数情况下要优于 NCA 算法[4]。因此，本章采用 LMNN 算法作为距离度量学习算法，它可以通过对训练集学习来得到一种原始数据的新度量，这种方法可以在一定程度上对原始数据分布进行重构，得到一个更加合理的数据分类空间。其次，要解决的就是样本密度分布不均的问题，在应用 LMNN 算法时我们注意到可能会加剧样本的密度分布不均衡，这就使得解决密度分布不均问题变得更加迫切。目前，解决这一问题的方法主要分文两类：第一种就是基于裁剪的方法将对样本进行裁剪，使得类间密度分布区域平衡，这一类方法主要

有[64]，另一类方法是基于填充的方法，它根据某种约束对训练样本进行填充使其密度分布达到或接近均衡。考虑到整体系统的计算复杂度的问题，本章所采用的方法是一种密度加权的方法可以归为第一类，它是一种非参数的函数加权算法，具有计算简单实用性好特点。本章内容将按照以上两点改进进行展开。

## 4.2 基于 LMNN 算法的文本分类

文本分类首先要对文本进行特征提取将待测试文本和训练文本表示成向量空间模型(Vector Space Model, VSM)，定义 $D=\{x_1,x_2,\cdots,x_n\}$ 表示训练文本集合，$C=\{c_1,c_2,\cdots,c_m\}(m\geq 2)$ 为类别集合，其中 $x_i=(d_1,d_2,\cdots,d_k)$ 表示第 $i$ 篇文章，$d_i$ 表示文本向量的第 $i$ 维，此处采用 IG 算法作为特征提权算法，然后采用 LMNN 方法对训练数据集进行重构，最后使用 K 近邻分类器来实现文本分类，评价标准使用 F1 值和查准率、查全率。如下图所示：

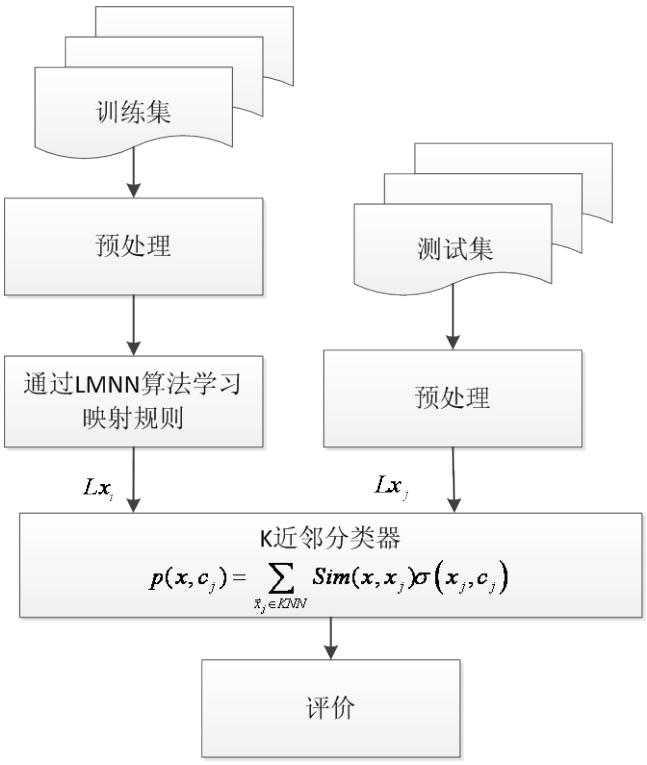


图 4-1 基于 LMNN 算法的文本分类程示意图

Fig.4-1 Text Classification progress diagram based on LMNN

根据第二章和第三章所述将样本集分为训练集和测试集两部分，依次进行预处理、特征提取、LMNN 算法距离度量学习得到映射矩阵 L、K 近邻分类器进行分类、评价等过程，由于上述过程前面已经有了详细介绍本章不再赘述。

### 4.2.1 基于 LMNN 的文本分类算法流程

大边界最近邻算法（LMNN）在应用于分类时对训练数据需要有每个训练数据的 K 近邻先验知识，然后根据构造出的代价函数采用留一法对训练误差求最小化，最终求得映射矩阵 $L$，而这个先验知识在文本分类数据中是没有的，因此我们先采用欧式距离对训练数据用留一法求得其每个数据的 K 个近邻并做好标签；然后利用上一章中提到的 LMNN 方法对训练数据求取映射矩阵 $L$；最后，结合 KNN 分类器，对中文文本进行分类。

其具体流程如下：

（1） 首先，对中文文本文本进行分词、去停用词等预处理。（本文采用的时张华平博士所共享的免费分词工具 ICTCLAS2011）

（2） 对文本进行特征选择，本文选用了 IG 这种常用的的特征提取算法来对文本进行特征提取。

（3） 构造向量空间模型(Vector Space Model，VSM)，本文所采用的是经典 TF*IDF 法。

（4） 对训练样本以欧氏距离用留一法计算出训练集中每个数据点的先验知识 K 近邻，并做好标签，设定此 K 值为 $K_p$。

（5） 利用 LMNN 算法对训练集进行学习，求出映射矩阵 $L$。

（6） 对训练样本和测试样本分别作映射 $x_i^{'} = Lx_i$。

（7） 跟据上一节中所提出的基于 LMNN 的文本分类算法对测试集进行分类。

### 4.2.2 实验仿真结果及分析

本文所采用的语料库是搜狗实验室共享的新闻语料库，我们选取其中的 5 类，包括文化 2140 篇，教育 150 篇，娱乐 800 篇，历史 1300 篇，军事 1400 篇。按照 70%作为训练集，30%作为测试集分为两份。选取的特征数为 50，评价标准采用 F1 值、查准率、查全率三种标准。实验程序是主要使用 C++语言编写，在程序算法的编写过程中，我们使用了 MatLab 与 C++语言的接口，在计算距离度量矩阵时调用 MatLab

程序实现 LMNN 算法的计算，大大简化的算法编写的复杂度。为了程序的独立性和可靠性，我们采用了本地 IO 读写的方式。

实验发现在先验 $K_p = 7$ 时的 LMNN 文本分类算法分类效果较好，因此列出此时 KNN 算法和 LMNN 算法在不同类别之间的查准率和查全率，具体如表 4-1 所示：

表 4-1 KNN 和 LMNN 文本分类实验结果（K=26, $K_p = 7$ 时不同类别评价）

Table 4-1 The result of text classification using KNN and LMNN(K=26, $K_p = 7$)

| 算法 | KNN | | | LMNN | | |
|---|---|---|---|---|---|---|
| 类别 | 查全率 | 查准率 | F1 值 | 查全率 | 查准率 | F1 值 |
| 文化 | 0.721 | 0.799 | 0.757 | 0.812 | 0.826 | 0.819 |
| 教育 | 0.634 | 0.721 | 0.676 | 0.821 | 0.882 | 0.850 |
| 娱乐 | 0.715 | 0.784 | 0.747 | 0.843 | 0.856 | 0.849 |
| 历史 | 0.742 | 0.773 | 0.757 | 0.810 | 0.823 | 0.816 |
| 军事 | 0.733 | 0.780 | 0.755 | 0.819 | 0.831 | 0.825 |

### 不同 K 值情况下 LMNN 算法和 KNN 算法的性能比较

由于 LMNN 算法是采用留一法对每一训练数据根据其先验的 K 近邻知识来判断样本点是否需要乘以惩罚系数，这里的先验知识为了区分后面的 K 近邻分类中的 K 值以 $K_p$ 表示，此处 $K_p$ 值不宜取的过大，否则计算量会非常大而且效果也不能保证。此处我们默认 $K_p$=7，下面对分类整体在 K 取不同值(此处在区间 0~30 隔一位取一次)情况下的查全率、查准率、F1 值作评价：

表 4-2 KNN 和 LMNN 在不同 K 值下实验结果（此时固定 $K_p = 7$ ）

Table 4-2 The result of KNN and LMNN with difference K( $K_p = 7$ )

| 算法 | KNN | | | LMNN | | |
|---|---|---|---|---|---|---|
| K 值 | 查全率 | 查准率 | F1 值 | 查全率 | 查准率 | F1 值 |
| 2 | 0.719 | 0.801 | 0.758 | 0.811 | 0.825 | 0.818 |
| 4 | 0.722 | 0.815 | 0.766 | 0.808 | 0.826 | 0.817 |

| 算法 | KNN | | | LMNN | | |
|---|---|---|---|---|---|---|
| K 值 | 查全率 | 查准率 | K 值 | 查全率 | 查准率 | K 值 |
| 6 | 0.713 | 0.816 | 0.761 | 0.799 | 0.827 | 0.813 |
| 8 | 0.731 | 0.784 | 0.757 | 0.814 | 0.828 | 0.821 |
| 10 | 0.732 | 0.782 | 0.756 | 0.819 | 0.821 | 0.820 |
| 12 | 0.720 | 0.798 | 0.757 | 0.812 | 0.826 | 0.819 |
| 14 | 0.730 | 0.802 | 0.769 | 0.817 | 0.831 | 0.824 |
| 16 | 0.743 | 0.803 | 0.772 | 0.822 | 0.830 | 0.826 |
| 18 | 0.742 | 0.802 | 0.771 | 0.823 | 0.833 | 0.828 |
| 20 | 0.735 | 0.804 | 0.768 | 0.817 | 0.837 | 0.827 |
| 22 | 0.721 | 0.799 | 0.758 | 0.821 | 0.837 | 0.829 |
| 24 | 0.724 | 0.793 | 0.757 | 0.823 | 0.833 | 0.828 |
| 26 | 0.732 | 0.807 | 0.768 | 0.818 | 0.832 | 0.825 |
| 28 | 0.722 | 0.793 | 0.756 | 0.811 | 0.827 | 0.819 |
| 30 | 0.717 | 0.784 | 0.749 | 0.827 | 0.833 | 0.830 |

由上表数据可以看出在固定 $K_p$ 时随着 K 值的不断增加 K 近邻分类器性能有一个先上升后缓慢下降的过程，但是整体性能比较平稳，而采用了 LMNN 算法后分类器性能有了较明显的提升，当然，分类效果的好坏并不是受单一因素的影响，比如分词算法的选择、向量空间模型中特征系数的确定、特征提取算法的不同都会对分类造成影响，这里只讨论固定上述情况下，KNN 算法和 LMNN 算法的比较。

**不同先验值 $K_p$ 情况下各类算法的性能**

由于此处无法预先确定一个合适的 K 值来考察不同的先验 $K_p$ 情况下 KNN 与 LMNN 算法的性能，我们只能根据上文实验获得的数据发现在 K 近邻分类中当 K=26 时分类效果较好，因此此处默认 K 取值为 26，评价标准依然采用查全率、查准率和 F1 测试值，此处由于针对不同的先验知识 $K_p$ 由于采用留一法对训练样本做标签故随着 $K_p$ 值的增大，计算复杂度也大大增加，加入 LMNN 算法的意义也就降低所以此处只比较在范围 3~11 时的算法性能：

表 4-3 KNN 和 LMNN 在不同 $K_p$ 值下实验结果（此时固定 K=26）

Table 4-3 The result of KNN and LMNN with difference $K_p$ (K=26)

| 算法 | KNN | | | LMNN | | |
|---|---|---|---|---|---|---|
| $K_p$ 值 | 查全率 | 查准率 | F1 值 | 查全率 | 查准率 | F1 值 |
| 3 | 0.732 | 0.807 | 0.768 | 0.812 | 0.832 | 0.822 |
| 4 | 0.732 | 0.807 | 0.768 | 0.809 | 0.833 | 0.821 |
| 5 | 0.732 | 0.807 | 0.768 | 0.815 | 0.835 | 0.825 |
| 6 | 0.732 | 0.807 | 0.768 | 0.817 | 0.835 | 0.826 |
| 7 | 0.732 | 0.807 | 0.768 | 0.818 | 0.832 | 0.825 |
| 8 | 0.732 | 0.807 | 0.768 | 0.820 | 0.830 | 0.825 |
| 9 | 0.732 | 0.807 | 0.768 | 0.817 | 0.831 | 0.824 |
| 10 | 0.732 | 0.807 | 0.768 | 0.811 | 0.835 | 0.823 |
| 11 | 0.732 | 0.807 | 0.768 | 0.823 | 0.831 | 0.827 |

LMNN 算法根据先验知识 $K_p$ 来确定需要迭代计算的点，如果 $K_p$ 过小的话不能够发挥该算法的效果，但如果取 $K_p$ 过大则会导致计算量大幅增加，根据上表显示在固定 K=26 的情况下 $K_p$ 在区间 3~11 间分类效果较为稳定，有一个小幅的上升下降趋势，但并不明显，因此这也说明了不宜将 $K_p$ 取得过大，否则会得不偿失。

## 4.3 基于密度加权的 LMNN 分类算法

由上文介绍可知在应用 K 近邻分类时 KNN 算法有一个较为明显的缺点就是对样本密度分布会比较敏感，当样本密度分布不均时分类效果会有较为明显的下降，而在文本分类中文章样本有着较严重的类偏斜情况，例如某网站中娱乐类新闻明显要比历史类新闻要多的多，这就有可能造成经特征提取后的数据点在某种度量意义下密度分布不均衡，特别地在应用 LMNN 算法来对样本点进行距离度量学习时，考察

以下公式：

$$\left\|L(x_i - x_l)\right\|^2 \le \left\|L(x_i - x_j)\right\|^2 + 1 \tag{4-1}$$

它描述了了在目标样本 $x_i$ 在其 K 个近邻中噪声点(impostor)的标准,并且以此定义非等价约束条件，对近邻中的异类点有一个推力作用，使其在马氏距离度量意义下远离目标样本，对于同类的样本在最优化公式(3-29)中第一项可以看到，等价为有一个拉力的作用，其在马氏距离度量意义下靠近目标样本，如下图所示:

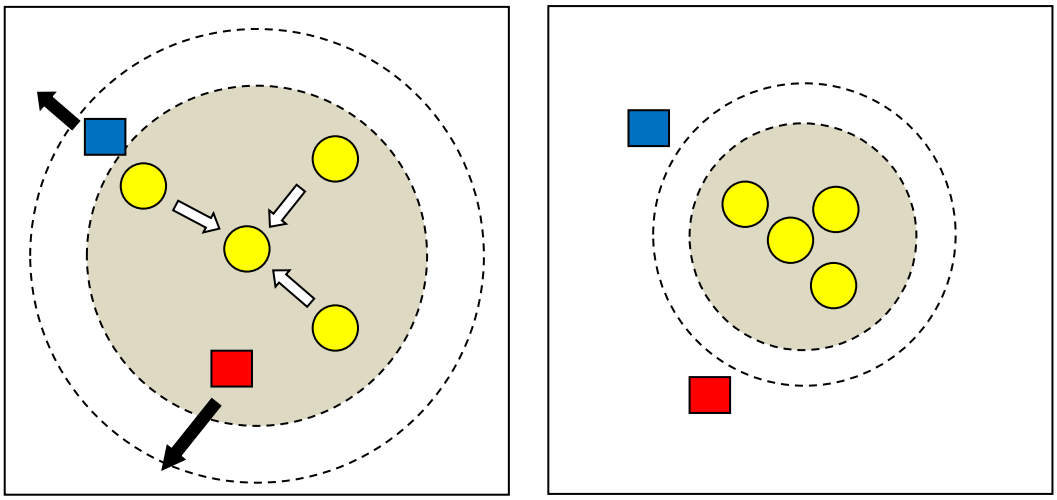

图 4-2 LMNN 算法作用示意图

Fig.4-2 LMNN algorithm's role diagram

上图左图为原始数据空间，右图为 LMNN 算法映射后的数据空间，可以观察到有可能导致在使用 LMNN 算法后使得数据更加趋于密度不均衡，从而限制文本分类的精度提高。

### 4.3.1 基于密度加权的 K 近邻分类算法

考虑到中文文本分类中不可避免的存在着由于语料库的选择、特征提取所造成的个别数据点远离类中心的情况，将距离学习算法 LMNN 应用到文本分类中可以很大程度上解决这种问题。然而，在将 LMNN 算法与 KNN 结合时由于 LMNN 算法的特点在很大程度上可能会增加样本密度分布不均匀，在应用于文本分类仍然会有较大的误差。

针对此，我们又提出了基于密度加权的 K 近邻分类算法和 LMNN 文本分类算法相结合，称之为 DLMNNC 算法，可以一定程度上解决上述由于 LMNN 的引入所造

成的密度分布更加不均匀问题。首先定义密度函数：

$$D(x_i,c_i)=\frac{n_i}{K}\sum_{x_i,x_j\in KNN}\frac{Sim(x_i,x_j)}{K} \tag{4-2}$$

其中， $x_i$ 为 $x_j$ 的 K 近邻点，$D(x_i,c_i)$ 表示 K 近邻中类标签为 $y_i$ 向量的密度，K 为最近邻数，$n_i$ 为类标签为 $y_i$ 的 K 近邻中向量个数，$Sim(x_i,x_j)$ 表示 $x_i,x_j$ 的余弦相似度为：$Sim(x_i,x_j)=\frac{\sum_{k=1}^{n}d_{ik}d_{jk}}{\sqrt{(\sum_{k=1}^{n}d_{ik}^2)(\sum_{k=1}^{n}d_{jk}^2)}}$，其中，$d_{ik}$ 表示 $x_i$ 的第 K 维向量。K 近邻决策公式可以表示为：

$$p(x,c_j)=\sum_{x_j\in KNN}Sim(x,x_j)\sigma\left(x_j,c_j\right) \tag{4-3}$$

$$\sigma\left(x_j,c_j\right)=\frac{1}{\exp(D(x_j,c_j)} \tag{4-4}$$

公式（4-2）定义了一个密度公式，对 K 近邻中所有样本点与给定选定的样本 $x_i$ 求平均相似度，可以在相似度程度上反映该类样本的密度，加权因子 $\frac{n_i}{K}$ 可以在数量程度上反映样本的密度。观察公式（4-4）如果样本密度低则权重系数 $\sigma\left(x_j,c_j\right)$ 一定程度上增大，如果样本密度高则其权重系数相应降低，因此，可以降低密度分布不均对分类造成的影响。

### 4.3.2 实验仿真结果及分析

本节实验条件与上一节一致，由于文本分类数据维数较大，无法实现可视化，因此为了验证上述推论的正确性，我们采用 iris 数据集选取其中 150 个数据类标签有 3 中分别标记为 1,2,3。并且，将其中的 112 个数据拿来做训练集，剩下 38 个作为测试集。然后使用 PCA 算法对其降维实现可视化，最后对其使用使用 LMNN 算法观察数据密度分布变化情况，此时先验知识 $K_p$ 取 3，此时训练的到的距离度量矩阵 $L$ 为：

$$L=\begin{bmatrix}-1.9811 & 1.5283\\ -1.3820 & 1.7915\end{bmatrix} \tag{4-5}$$

分类准确率，此时 KNN 算法在 K 取 3 时训练集的准确率为 92.11%,LMNN 算法的准确率为 94.74%，采用 DLMNNC 的分类准确率为%95.36，在测试集中 KNN 算法在 K 取 3 时的准确率 93.32%，LMNN 算法的准确率为 97.79%，DLMNNC 的分类准确率为%97.79。可见 DLMNNC 算法确实在一定程度上缓解了 LMNN 算法所引起的密度不均问题。下图分别展示了训练集未使用 LMNN 算法前的密度分布，使用 LMNN 算法后的密度分布，测试集未使用 LMNN 算法前的密度分布，使用 LMNN 算法后样本分布。

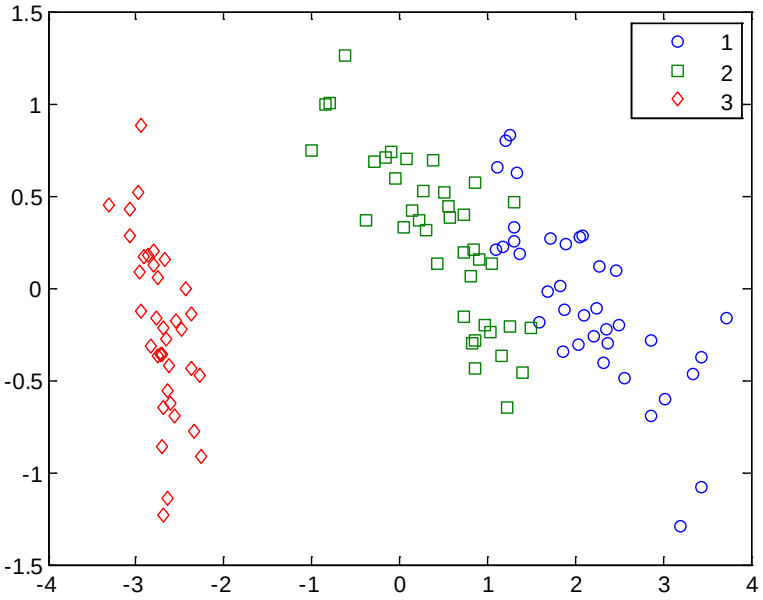


图 4-2 Iris 数据集中训练数据分布

Fig.4-2 Training data distribution of Iris

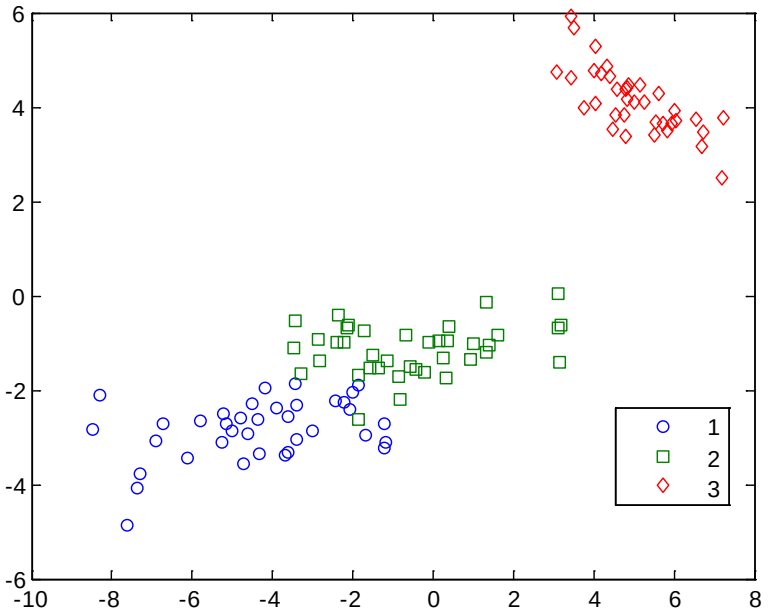


图 4-3 LMNN 算法映射后的训练数据集

Fig.4-3 Mapped Training data distribution using LMNN

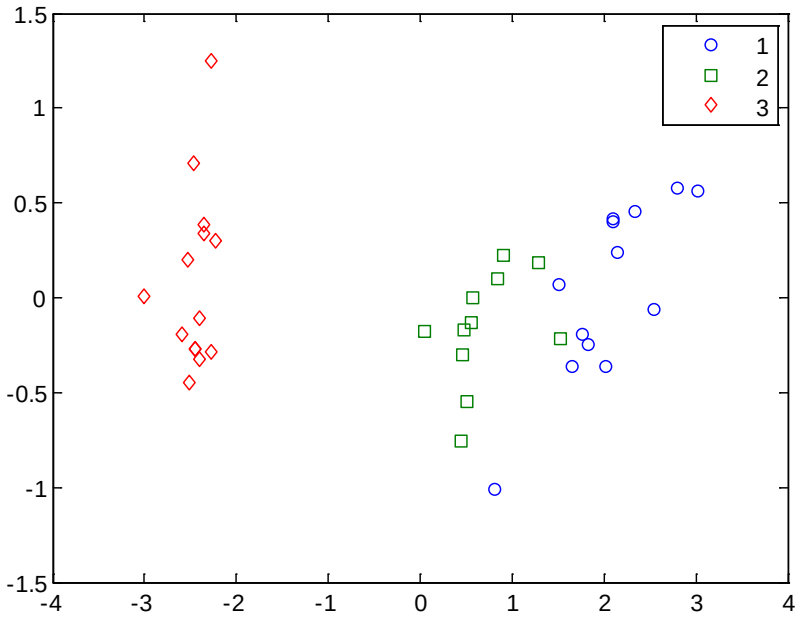


图 4-4 Iris 数据集中测试数据分布

Fig.4-4Testing data distribution of Iris

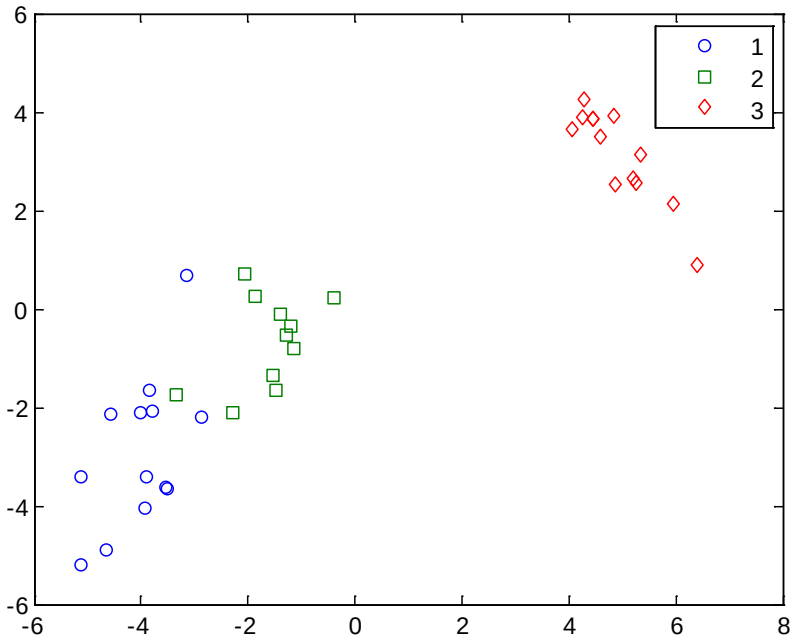


图 4-5 LMNN 算法映射后的测试数据集

Fig.4-5 Mapped Testing data distribution using LMNN

本节的主要思路是加上密度加权函数后的 K 近邻分类与 LMNN 算法相结合来做文本分类(DLMNNC)，上面一节验证了密度加权函数的必要性，在本节中将应用真实的文本分类数据集来验证 KNN 分类算法、LMNN 算法、DLMNNC 算法的性能，实验设定与上文一致。首先对固定 K=26 值和 $K_p$=7 情况下各个类别间的查准率、查全率做比较如下表：

表 4-4 KNN、LMNN 和 DLMMC 算法比较（此时固定 K=26，$K_p=7$ ）

Table 4-4 The result of text classification using KNN and LMNN(K=26, $K_p=7$ )

| 算法 | KNN | | LMNN | | DLMNNC | |
|---|---|---|---|---|---|---|
| | 查全率 | 查准率 | 查全率 | 查准率 | 查准率 | 查全率 |
| 文化 | 0.721 | 0.799 | 0.812 | 0.826 | 0.836 | 0.868 |
| 教育 | 0.634 | 0.721 | 0.821 | 0.882 | 0.862 | 0.880 |
| 娱乐 | 0.715 | 0.784 | 0.843 | 0.856 | 0.857 | 0.865 |
| 历史 | 0.742 | 0.773 | 0.810 | 0.823 | 0.830 | 0.844 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 军事 | 0.733 | 0.780 | 0.819 | 0.831 | 0.824 | 0.872 |

可以看到 KNN 算法在各个类别中查准率和查全率都明显低于 LMNN 算法，而 DLMNNC 算法则在一定程度上比 LMNN 算法有了更大的提升。

### 搜狗数据集中不同 K 值情况下三种算法性能比较

上文中已经比较了 KNN 算法和 LMNN 算法在不同 K 值情况下的算法性能，包括算法的查准率、查全率、F1 值，本节实验条件与上文一致固定 $K_p$=7，为了更加直观只在图中对比三种算法的 F1 测试值。

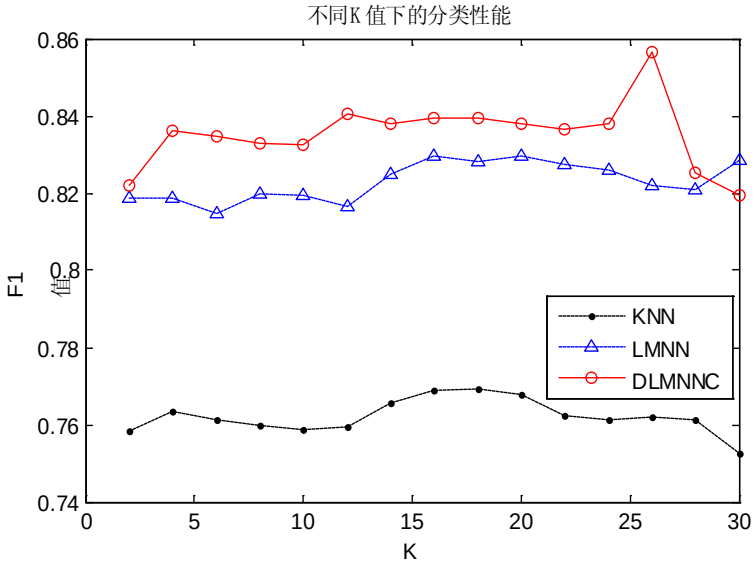


图 4-6 不同 K 值下三种分类器的性能

Fig.4-6 The performance of three classifier with different K values

由上图可见 DLMNNC 确实在一定程度上提高了分类的精度，观察图 4-6 可以看到在 K 取 2 到 30 的区间内 DLMNNC 算法明显优于 LMNN 和 KNN 算法，在 K 小于 5 和大于 30 附近时可以看到使用 DLMNNC 算法的分类器有明显的性能下降，这是由于 DLMNNC 算法是采用密度对近邻点进行加权，当 K 值取的较大或较小时均不能正确反映数据密度分布情况。

### 搜狗数据集中不同先验值 $K_p$ 情况下三种算法性能比较

由上述实验条件一致，调节 $K_p$ 观察在不同值情况下 LMNN 算法和 DLMNNC 算

法的性能改变情况，取 K=26(由于 KNN 算法不需要先验知识$K_p$，此时 KNN 算法 F1 值维持不变)。如下图所示：

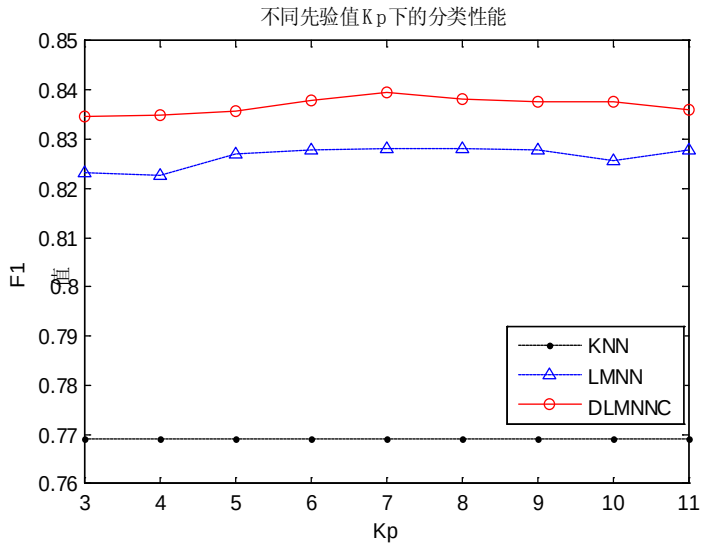


图 4-7 不同$K_p$值下三种分类器的性能

Fig.4-7 The performance of three classifier with different $K_p$ values

由于 LMNN 算法的特点$K_p$值不宜取的过大，否则计算量会非常大而且效果也不能保证。所以，此处我们在区间 3 到 11 之间进行试验。图 4-7 中可以看到 DLMNNC 算法在 K=26 时效果普遍优于 LMNN 和 KNN 算法，而且在$K_p$增大的过程中分类器性能没有明显的变化，但是随着$K_p$的增大计算时间却几何倍数增加，这也说明了$K_p$不宜取的过大。

## 4.4 本章小结

本章主要研究了一种大边界最近邻算法(LMNN)，并且将其应用到文本分类领域，然后考虑到 LMNN 算法在实际应用时可能会加剧文本数据密度分布不均的情况，提出了基于密度加权的 K 近邻算法来和 LMNN 算法结合使用，又使用 Iris 数据集，经 PCA 降维实现数据的可视化，某种程度上验证了上述推论，最后使用搜狗文本分类数据集对文中所提到的三种算法做了比较分析。

# 第五章 基于余弦距离度量学习的伪 K 近邻文本分类算法

本章主要研究了一种新的余弦距离度量学习算法，并且将其应用到文本分类中。然后，提出使用一种伪 K 近邻分类算法来和余弦距离度量学习算法相结合来克服将距离度量学习引入到文本分类所遇到的问题。最后，通过实验仿真表明，该算法有着较好的分类效果。

## 5.1 背景和初衷

上文已经介绍了在文本分类中一个较为重要的问题是如何解决样本的相似度度量问题，目前无论是商业上还是学术界都有一个共识就是对于文本数据余弦距离度量要比欧式距离度量要好一些，这主要因为两点：首先，在文章中特定的主题肯定使用特定的描述词。因此，同一类新闻一定是某些主题词用的比较多，另外一些词用的少。比如金融类新闻，股票、利息、基金、银行、债券这些词出现的频率就比较高，而类似二氧化碳、诗歌、宇宙等词就会出现的比较少。反映在每篇新闻的特征上，如果某两篇新闻是同一类的，那么在他们的特征向量的某几维上的度量值都比较大，而其他维度值比较小。这就是说，对于不同向量，方向性要比数值更加重要，而传统的欧氏距离度量标准只对数值敏感，并没有利用向量之间的方向性。而余弦相似度和欧式距离度量相比较，更加注重两个向量在方向上的差异，而非距离或长度。其次，在实际应用中如果采用欧式距离度量方式来对文本进行评价往往要对其进行归一化，否则无法避免不同长度的文章中某些词汇密度分布失衡造成的分类误差。这样做虽然解决了这一问题，但是忽视了在文本分类中数据维数往往都非常高，随着文本特征向量维度的提高，文本特征向量的欧氏距离度量也都会趋向于 1，这显然对于分类是非常不利的，而使用余弦距离度量由于其本身就相当于归一化的可以避开这一问题。

因此，本章提出一种基于余弦距离度量的距离度量学习算法来对文本数据进行距离度量。在上一章中提到了应用 LMNN 算法可能会造成数据密度分布不均衡，本章所提出的余弦距离度量学习算法与 LMNN 算法类似也是基于最大边界来构造对约束条件的，因此为了缓解因此造成的样本密度改变，提出使用一种伪 K 近邻分类算法来进行分类，这种算法在本质上说是属于基于数据填充的一中算法，而且，由于这种算法是在每一类中都取 K 个最近邻，所以这在一定程度上也缓解了文本分类中另

一普遍问题—类偏斜所带来的分类误差，使该算法更加适用于类偏斜数据集。

## 5.2 基于余弦的距离度量学习(CS-LMNN)算法

有监督的距离度量学习算法的一般框架是根据对约束条件(Pairwise Constraints)分别确定等价性约束(equivalence constrains)和非等价性约束(inequivalence constrains)，最后构造凸规划问题求解。所谓的对约束条件就是：1)类标签相同的点应尽量相近，2)类标签不同的点应尽量远离。根据这两个条件我们提出了一种新的基于余弦的距离度量学习算法，称之为 CS-LMNN 算法。该算法和 LMNN 算法类似，也需要训练集的 K 近邻先验知识同样以 $K_p$ 表示，它根据余弦夹角的性质，即任意夹角的余弦值不可能大于 1，这一条件来构造非等价性约束，然后，在最优化表达式中，通过最小化近邻同类标签样本的余弦距离来构造等价性条件。最终，将两条件改写为一个最优化问题进行求解。具体算法流程如下:首先，定义余弦距离度量,定义 1：在训练集 D 中任意两点 $x_i, x_j$ 间的余弦距离度量表达式：

$$CS_M(x_i, x_j) = \frac{x_i^T M x_j}{\sqrt{x_i^T M x_i}\sqrt{x_j^T M x_j}} \tag{5-1}$$

此处，$M = L^{\mathrm{T}}L$，$M \succeq 0$ 是一个对称半正定矩阵，$L$ 为所求距离度量矩阵。其中 $CS_M$ 满足一般余弦相似度的所有性质。下文中所有符号定义与上文保持一致。假设，目标样本 $x_i$ 具有类标签 $c_i$ 在其 K 近邻点中有 $x_l$ 类标签为 $c_l$，定义噪声点为对任意目标样本 $x_i$ 有 $c_l \neq c_i$，满足：

$$CS_M\left(x_i, x_l\right) \leq 1 - CS_M\left(x_i, x_j\right) \tag{5-2}$$

其中，$x_i$ 为输入向量，$x_l$ 为 $x_i$ 的 K 近邻点但是和 $x_i$ 类标签不同，$x_j$ 为 $x_i$ 的 K 近邻点且和 $x_i$ 的类标签相同。

根据式(5-2)，首先定义非等价性约束条件：

$$\varepsilon_{push}(M) = \sum_{i,j \in K_p NN} \sum_{l} (1 - y_{il})[CS_M\left(x_i, x_j\right) + CS_M\left(x_i, x_l\right) - 1]_+ \tag{5-3}$$

上式中，$j \in K_p NN$ 表示训练样本 $x_i$ 为测试样本 $x_j$ 的 K 近邻，此 K 近邻为先验知识以 $K_p$ 表示；$x_l$ 表示与处于 $x_i$ 最大间距内但又与测试样本类标签不相同的训练样本，即满足式(5-2)的样本；$c_l$ 为 $x_l$ 的类标签；当 $x_i$ 的类标签 $c_i = c_l$ 时 $y_{il} = 1$ 否则为 0；标记 $[Z]_+ = \max(Z, 0)$，这就保证了对 $x_i$ 的 K 近邻点且类标签不同的点中与 $x_i$ 相似度较小的不作处理，可以大大减少计算量。

然后定义等价性约束：

$$\varepsilon_{pull}(M) = \sum_{i, j \in K_p NN} CS_M\left(x_i, x_j\right) \tag{5-4}$$

公式(5-4)的目的是增大与 $x_i$ 为 K 近邻点的 $x_j$ 相似度,即使其余弦夹角变小。

最后，结合公式(5-3)和(5-4)构造如下损失函数：

$$\varepsilon(M) = (1-\mu)\varepsilon_{pull}(M) + \mu\varepsilon_{push}(M) \tag{5-5}$$

其中，$\mu$ 为权重系数一般取 0.5。可以看出惩罚函数的第一项 $\varepsilon_{pull}(M)$ 只惩罚与测试样本 $x_i$ 类标签相同但是距离处于最大边界之外的训练样本，第二项 $\varepsilon_{push}(M)$ 只惩罚与测试样本 $x_i$ 类标签不同但是又处于最大边界之内的训练样本，这样就保证了在求全局最优映射 $L$ 时只对影响 KNN 分类的点进行惩罚可以有效降低错误率和计算复杂度。若将上式转换为最优化问题求解，需要首先将其转化为更标准的形式，引入非负松弛变量 $\xi_{ijl}$ ,来衡量式(5-2)的非等价性，构造以最优化问题:

$$\begin{aligned} &\max \ (1-\mu)\sum_{i, j \in KNN} CS_M(x_i, x_j) + \mu \sum_{i, j \in KNN, l} (1 - y_{il})\xi_{ijl} \\ &(1) \ \ CS_M(x_i, x_j) + CS_M(x_i, x_l) \le 1 - \xi_{ijl} \\ &(2) \ \ \xi_{ijl} \ge 0 \\ &(3) \ \ M \succeq 0 \end{aligned} \tag{5-6}$$

其中，$\mu$ 为调节系数，一般取 0.5，$CS_M(x_i, x_j)$ 表示目标点 $x_i$ 和其 K 近邻点 $x_j$ 的余弦

距离度量，且 $x_i, x_j$ 类标签相同，$CS_M(x_i, x_j)$ 表示目标点 $x_i$ 其 K 近邻点 $x_l$ 的余弦距离度量，且 $x_i, x_l$ 的类标签不同，当目标点 $x_i$ 与其 K 近邻测试点类类标签相同时 $y_{il}$ 取 1，否则为 0。观察公式（5-6）可以看出目标是使 $CS_M(x_i, x_j)$ 增大，由于利用式（5-2）定义最优化条件，当 $CS_M(x_i, x_j)$ 增大时，$CS_M(x_i, x_l)$ 相应减小，$\xi_{ijl}$ 为松弛变量用来衡量式（5-2）的不平衡程度。这个最优化问题的复杂度为 $O(kn^2)$，采用一般求解最优化问题的工具包可能不太适合，此处采用一种交叉验证的算法来求解。

## 5.3 基于余弦距离度量(CS-LMNN)的文本分类算法

文本分类流程与 4.2 节所述基本一致，此处不再赘述，唯一不同的地方是将 LMNN 算法替换为上一节提出的余弦距离度量学习算法，算法具体流程如下图所示：

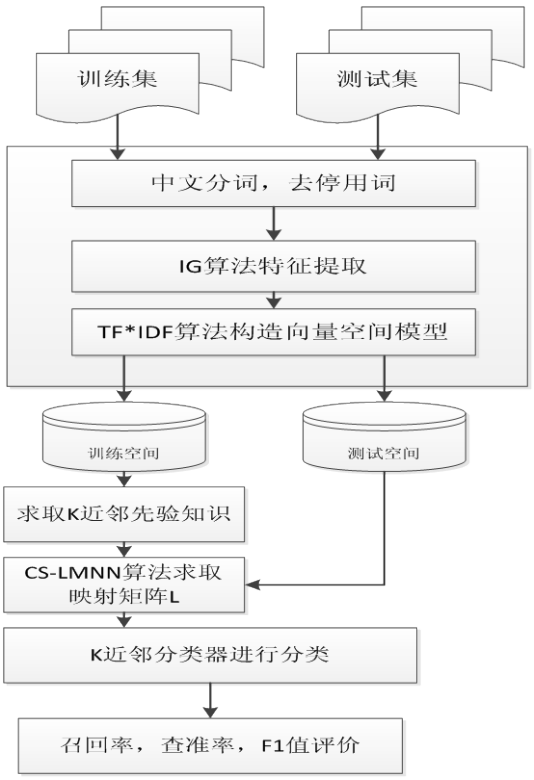


图 5-1 基于 CS-LMNN 算法的文本分类流程图

Fig.5-1 ext Classification progress diagram based on LMNN

### 5.3.1 实验结果与分析

为了下文中验证伪 K 近邻算法在具有类偏斜情况下的分类性能，我们特地在搜狗数据集中选取了选取了一组类偏斜较为严重的文本集作为训练和测试语料库，总计 5 类，其中包括文化 2000 篇，教育 50 篇，娱乐 800 篇，历史 1300 篇，军事 1400 篇。按照 70%作为训练集，剩下的 30%作为测试集分为两份。评价标准选取较为公认的，召回率（Recall, R），准确率（Precision, P）和 F1 测试值。

首先，固定在 $K_p$ =8，k=10 时对比各个类别的查准率、查全率和 F1 值,其中以 CS-LMNN-K 代表本章所提算法和 KNN 相结合的文本分类算法，LMNN-K 表示以 LMNN 算法和 KNN 算法相结合的文本分类算法，具体结果如下表：

表 5-1 KNN、LMNN-K 和 CS-LMNN-K 算法比较（此时固定 K=10，$K_p=8$）

Table 5-1 The result of text classification using KNN,LMNN-K and CS-LMNN-K(K=10, $K_p=8$)

| 算法 | KNN | | | LMNN-K | | | CS-LMNN-K | | |
|---|---|---|---|---|---|---|---|---|---|
| | 查全率 | 查准率 | F1 值 | 查全率 | 查准率 | F1 值 | 查准率 | 查全率 | F1 值 |
| 教育 | 0.691 | 0.815 | 0.747 | 0.824 | 0.912 | 0.865 | 0.872 | 0.910 | 0.890 |
| 文化 | 0.722 | 0.831 | 0.772 | 0.791 | 0.906 | 0.844 | 0.891 | 0.907 | 0.898 |
| 娱乐 | 0.773 | 0.782 | 0.777 | 0.872 | 0.902 | 0.887 | 0.852 | 0.899 | 0.874 |
| 历史 | 0.787 | 0.769 | 0.778 | 0.894 | 0.879 | 0.886 | 0.901 | 0.903 | 0.902 |
| 军事 | 0.792 | 0.773 | 0.782 | 0.880 | 0.931 | 0.905 | 0.893 | 0.903 | 0.897 |

可以发现采用 CS-LMNN 算法的文本分类中分类效果要优于或者接近 LMNN 算法的。

不同 K 值情况下各类算法的性能

此处由于 CS-LMNN 和 LMNN 算法需要有一个先验的 K 近邻知识 $K_p$，目前没有较好的方法来确定它的数值只能根据经验选取。由于 $K_p=8$ 时分类效果较好且计算量也不是很大，本节考察 $K_p=8$ 时不同的 K 值对分类 $K_p=8$ 效果的影响，此处 K 值取区间 1~30，每隔一位取一次。

表 5-2 KNN、LMNN-K 和 CS-LMN-K 在不同 K 值下实验结果（此时固定 $K_p = 8$ ）

Table 5-2 The result of KNN,LMNN-K and CS-LMNN-K with difference K( $K_p = 8$ )

| 算法 | KNN | | | LMNN-K | | | CS-LMNN-K | | |
|---|---|---|---|---|---|---|---|---|---|
| K 值 | 查全率 | 查准率 | F1 值 | 查全率 | 查准率 | F1 值 | 查准率 | 查全率 | F1 值 |
| 2 | 0.707 | 0.790 | 0.746 | 0.848 | 0.856 | 0.852 | 0.861 | 0.885 | 0.873 |
| 4 | 0.721 | 0.786 | 0.752 | 0.859 | 0.875 | 0.867 | 0.876 | 0.882 | 0.879 |
| 6 | 0.718 | 0.823 | 0.767 | 0.866 | 0.876 | 0.871 | 0.883 | 0.899 | 0.891 |
| 8 | 0.733 | 0.829 | 0.778 | 0.871 | 0.889 | 0.880 | 0.877 | 0.883 | 0.880 |
| 10 | 0.726 | 0.827 | 0.773 | 0.874 | 0.876 | 0.875 | 0.881 | 0.891 | 0.886 |
| 12 | 0.711 | 0.819 | 0.761 | 0.857 | 0.871 | 0.864 | 0.885 | 0.895 | 0.890 |
| 14 | 0.710 | 0.799 | 0.752 | 0.861 | 0.877 | 0.869 | 0.872 | 0.884 | 0.878 |
| 16 | 0.717 | 0.795 | 0.754 | 0.873 | 0.889 | 0.881 | 0.875 | 0.885 | 0.880 |
| 18 | 0.719 | 0.790 | 0.753 | 0.869 | 0.889 | 0.879 | 0.869 | 0.881 | 0.875 |
| 20 | 0.713 | 0.788 | 0.749 | 0.870 | 0.886 | 0.878 | 0.866 | 0.874 | 0.870 |
| 22 | 0.726 | 0.790 | 0.757 | 0.874 | 0.882 | 0.878 | 0.883 | 0.895 | 0.889 |
| 24 | 0.709 | 0.792 | 0.748 | 0.862 | 0.870 | 0.866 | 0.874 | 0.880 | 0.877 |
| 26 | 0.716 | 0.789 | 0.751 | 0.879 | 0.885 | 0.882 | 0.863 | 0.875 | 0.869 |
| 28 | 0.713 | 0.782 | 0.746 | 0.864 | 0.870 | 0.867 | 0.863 | 0.873 | 0.868 |
| 30 | 0.697 | 0.795 | 0.743 | 0.845 | 0.881 | 0.863 | 0.857 | 0.871 | 0.864 |

上述实验是在固定特征向量为 50 维的情况下进行的，可以看出随着 K 值的增加 KNN 算法、LMNN-K 算法和 CS-LMNN-K 算法的分类精度都在一定程度上波动，但是明显的是 LMNN-K 算法和 CS-LMNN-K 算法的性能要明显优于 KNN 算法，而在大多数情况下 CS-LMNN-K 的性能要优于 LMNN-K 算法。

**不同先验值 $K_p$ 情况下各类算法的性能**

由上述算法可知在 LMNN 算法和 CS-LMNN 算法中先验知识 $K_p$ 是非常重要的，本节主要讨论不同的 $K_p$ 值对分类器性能的影响，此处固定 K 值为 10, 由于 $K_p$ 的改变对计算量要求较大我们只在范围 2 到 10 中隔一位取一个进行实验，否则计算时间

过长失去意义。

表 5-3 KNN、LMNN-K 和 CS-LMN-K 在不同 $K_p$ 值下实验结果（此时固定 K=10）

Table 5-3 The result of KNN,LMNN-K and CS-LMNN-K with difference $K_p$ (K=10)

| 算法 | KNN | | | LMNN-K | | | CS-LMNN-K | | |
|---|---|---|---|---|---|---|---|---|---|
| $K_p$ | 查全率 | 查准率 | F1 值 | 查全率 | 查准率 | F1 值 | 查准率 | 查全率 | F1 值 |
| 2 | 0.726 | 0.827 | 0.773 | 0.863 | 0.881 | 0.872 | 0.871 | 0.885 | 0.878 |
| 4 | 0.726 | 0.827 | 0.773 | 0.865 | 0.881 | 0.873 | 0.873 | 0.889 | 0.881 |
| 6 | 0.726 | 0.827 | 0.773 | 0.869 | 0.879 | 0.874 | 0.876 | 0.888 | 0.882 |
| 8 | 0.726 | 0.827 | 0.773 | 0.874 | 0.876 | 0.875 | 0.881 | 0.891 | 0.886 |
| 10 | 0.726 | 0.827 | 0.773 | 0.868 | 0.876 | 0.872 | 0.884 | 0.917 | 0.900 |

观察上表可以发现，CS-LMNN-K 算法明显性能明显优于其他两种，而且随着 $K_p$ 的增大，各类算法性能并没有明显的改变，而时间消耗确大幅增加，这说明将 $K_p$ 值设为一个较大值所取得的算法性能提升对比算法时间消耗的增加是不值得的。

## 5.4 基于 CS-LMNN 的伪 K 近邻分类

### 5.4.1 伪 K 近邻分类算法

传统的 K 近邻分类算法的缺陷是没有考虑到样本的数量差异，现实情况中，在应用于文本分类时，文本类别间的数量差异往往很大，这时 K 近邻分类算法就倾向于选择数量较多的那一类，最终造成误判，这就是分中的类偏斜问题，而目前解决这一问题的方法主要有两类，一是对样本进行剪裁使得不同类别的样本数接近相同，二是对样本进行扩充，通过人工或者其他方式生成一部分样本以达到平衡各类别样本数量的目的。伪 K 近邻是一种类似于方式二的方法，它首先在不同类别中分别选取测试文本 $x$ 的 K 个近邻，然后使用加权平均的方式取得该文本在个类别的相似度加权平均值，最终将 $x$ 归入值最大的那一类实现分类，判别式如下：

$$p(x,y_i)=\frac{1}{K}\sum_{x_j\in KNN}^{c_i} Sim(x,x_j)\sigma\left(x_j,c_i\right) \tag{5-7}$$

其中，K 为 K 近邻数，$\sum_{x_j \in KNN}^{c_i}$ 表示 $x$ 的在 $c_i$ 类中的 K 近邻点，$Sim(x, x_j)$ 一般取余弦相似度，$\sigma\left(x_j, c_i\right)$ 为权重系数，一般有指数逆距离加权，倒数距离加权等几种方式。

分析上述算法可以发现，伪 K 近邻分类算法在每一个类别中取相同的数目的近邻来做分类判别，这种方法显然在处理类别边界处密度不均匀问题有着独到的优势，而在上一章中我们讨论了在引入 LMNN 算法时可能会造成的密度分布不均问题，并且提出了一种基于裁剪的方法来缓解 LMNN 算法对数据密度分布的影响，由于 CS-LMNN 算法在本质上与 LMNN 算法一致都是基于最大边界的，所以也有可能会引起数据的密度分布不均。因此，在本节中我们提出使用这种伪 K 近邻分类的方法来对 CS-LMNN 算法进行处理不仅可以克服一定程度上的类偏斜问题，还可以缓解由 CS-LMNN 算法带来的密度问题,此处简记此算法伪 PKNN。

### 5.4.2 基于 CS-LMNN 的伪 K 近邻分类流程

伪 K 近邻文本分类和 CS-LMNN 算法相结合的具体流程如下：

（1） 首先，对中文文本文本进行分词、去停用词等预处理。

（2） 对文本进行特征选择，本文选用了 IG 这种效果较稳定的的特征提取算法来对文本进行特征提取。

（3） 构造向量空间模型(Vector Space Model，VSM)，本章所采用的是经典 TF*IDF 法。

（4） 对训练样本根据余弦相似度用留一法计算出训练集中每个点的先验知识 K 近邻，并做好标签，设定此 K 值为 $K_p$。

（5） 利用 5.2 中提到的 CS-LMNN 算法对训练集进行学习，求出映射矩阵 $L$。

（6） 跟据 5.4.1 中所提出的伪 K 近邻分类算法对测试集进行分类，此处的 K 近邻个数记为 K。

（7） 根据分类结果评价其召回率，准确率，F1 测试值。

### 5.4.3 实验结果与分析

为了验证伪 K 近邻分类算法对类偏斜样本的处理能力本节采用的数据集与 5.3 节相同，为具有一定程度类偏斜的语料库。实验条件与上文所述一致，首先考察固定 K 值和先验 $K_p$ 值情况下,KNN、LMNN-K、CS-LMNN-K 和 CS-LMNN-PK 对不同类别样本的分类性能。

表 5-4 不同算法间召回率和准确率的比较（$K_p$=8，K=10）

Table 5-3 The comparison of recall rate and precision among different algorithms（$K_p$=8，K=10)

| | KNN | | LMNN-K | | CS-LMNN-K | | CS-LMNN-PK | |
|---|---|---|---|---|---|---|---|---|
| | 查全率 | 查准率 | 查全率 | 查准率 | 查准率 | 查全率 | 查准率 | 查全率 |
| 教育 | 0.691 | 0.815 | 0.824 | 0.912 | 0.872 | 0.910 | 0.911 | 0.934 |
| 文化 | 0.722 | 0.831 | 0.791 | 0.906 | 0.891 | 0.907 | 0.874 | 0.916 |
| 娱乐 | 0.773 | 0.782 | 0.872 | 0.902 | 0.852 | 0.899 | 0.886 | 0.881 |
| 历史 | 0.787 | 0.769 | 0.894 | 0.879 | 0.901 | 0.903 | 0.920 | 0.941 |
| 军事 | 0.792 | 0.773 | 0.880 | 0.931 | 0.893 | 0.903 | 0.926 | 0.921 |

不同 K 值情况下各类算法的性能

与上文一致此处固定 $K_p$=8，考察不同 K 值对几种分类算法的影响

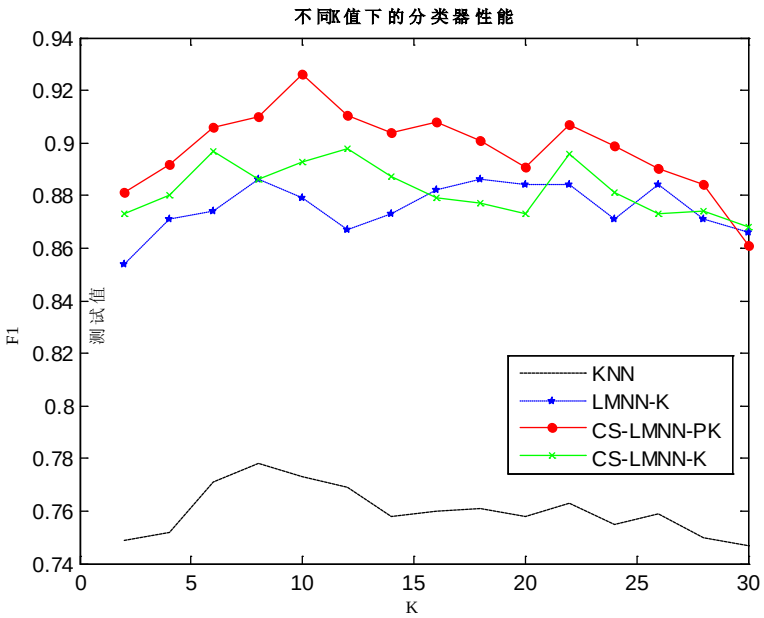


图 5-2 基于不同 K 值下各分类器的性能

Fig.5-2 The performance of different classifiers with different K

图 5-2 中 LMNN-K 算法指的是以先用 LMNN 算法进行度量学习再以 KNN 做分

类的一种分类算法，CS-LMNN-K 指的是以 CS-LMNN 算法为距离学习，以 KNN 算法做分类的一种分类算法，CS-LMNN-PK 指的是以 CS-LMNN 算法进行距离学习，结合伪 K 近邻分类的分类方法。参照标准为 KNN 算法，可以看出本章提出的 CS-LMNN 算法在和伪 K 近邻结合用来处理类倾斜较严重的语料库可以取得较好的效果。

### 不同先验值 $K_p$ 情况下各类算法的性能

为了考察算法的一般性和通用性，我们同时也对比了不同先验知识 $K_p$ 对分类效果的影响，此时固定 K=10。由于 $K_p$ 的改变对计算量要求较大我们只在范围 2 到 10 中隔一位取一个进行实验，否则计算时间过长失去意义。

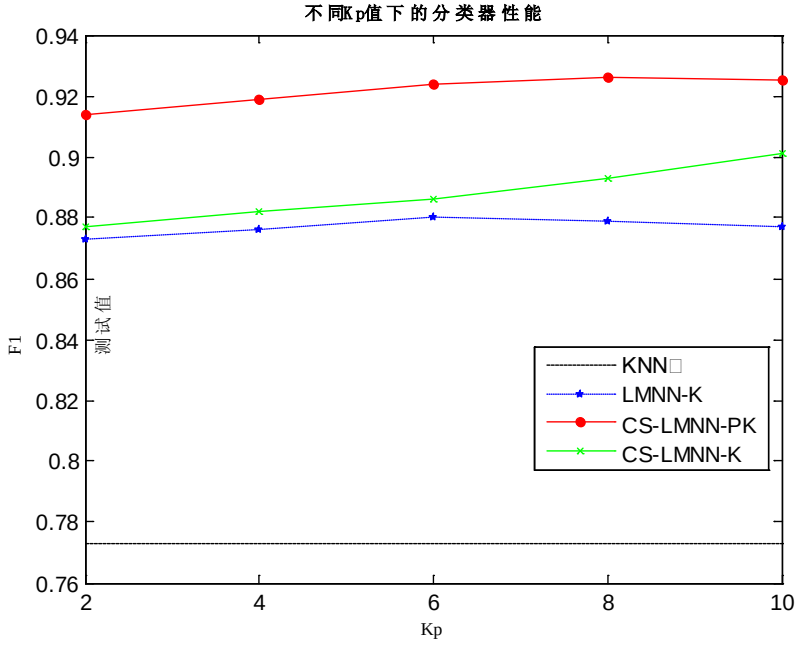


图 5-3 基于不同 $K_p$ 值下各分类器的性能

Fig.5-3 The performance of different classifiers with different $K_p$

图 5-3 中各曲线定义同图 5-2 中相同，可以看到 CS-LMNN-PK 的效果明显比其他几种算法好很多，且比较稳定，随着 $K_p$ 值的不断增大分类精度提高并不明显，这

也说明了 $K_p$ 不宜取的过大。

## 5.5 本章小结

本章首先提出了一种新的距离度量学习算法—余弦距离度量学习算法，对其原理做了详细介绍，然后和 KNN 算法结合将其应用到文本分类中，进行了大量对比试验，最后分析了该算法的缺点，并且提出了一种基于伪 K 近邻分类的改进方案，实验结果证明该算法和其改进算法具有良好的分类效果。

# 第六章 总结与展望

## 6.1 研究工作总结

本文对文本分类系统进行了比较深入的研究，并且尝试将距离度量学习引入到文本分类系统中，分析了现有距离度量学习的特点，提出了一种新的余弦距离度量学习算法。在将距离度量学习算法引入到文本分类的基础上，考虑应用这一类方法所带来的问题，分别提出了基于密度加权的文本分类改进方法和伪 K 近邻的分类算法最终设计实现了，基于 KNN 的文本分类系统，基于密度加权的 LMNN 文本分类系统，基于 CS-LMNN 的伪 K 近邻文本分类系统。具体成果包括：

1）考虑到 K 近邻等分类算法对样本距离度量方式的依赖性，提出使用目前较为流行的一类距离度量学习算法先对训练数据进行距离度量学习，来得到一种适合该分类算法的合适度量，再进行分类，在此基础上实现了基于 LMNN 算法的文本分类系统，并且应用搜狗数据集对算法性能进行了验证。

2）在将距离度量学习算法引入到文本分类系统后不可避免地会有一些问题产生，本文主要研究了一种大边界最近邻算法(LMNN),发现将其应用到文本分类系统时会引起样本密度的分布变化，有可能会加剧密度分布不均，这对 K 近邻分类器来说影响是比较大的，因此提出了一种基于密度加权的 K 近邻分类器来缓解引入距离度量学习算法带来的影响，最后实现了基于这种算法改进的分类系统，实验验证了该算法的有效性。

3）分析了目前距离度量学习算法普遍使用欧氏距离度量形式来学习数据的距离度量，在文本分类中，分类决策更加注重样本的方向性，而非数值。因此本文提出了一种基于余弦的距离度量学习算法，这种算法对向量的方向性更加敏感，在文本分类问题中更加适用，然而在将其和文本分类相结合时依然会遇到上述密度问题，为此，本文又提出了适用一种伪 K 近邻分类算法来对文本进行分类，这种算法不仅可以缓解因为密度改变带来的分类误差，还可以克服文本分类问题中另一个较为普遍的问题—类偏斜，最后我们使用一个具类偏斜的语料库来验证了这两种算法相结合的有效性。

## 6.2 未来的研究工作

文本分类是一个系统性的问题，它包括从预处理到分类器等一系列的问题。本文仅仅从预处理和分类器两个角度对文本分类做了一定的探索，提出将距离度量学习引入到文本分类，然而这类算法往往具有计算量大，计算复杂等特点，在实际应用时实用性还比较差，这将是未来研究改进的一个重点，另外本文只研究了一类使用与 K 近邻分类的距离度量学习算法，而目前分类算法的发展也比较迅速，例如比较著名的支持向量机(SVM)，朴素贝叶斯等都是分类效果比较好的，如何寻找使用于这一类分类算法的距离度量仍是目前研究的一个难点，也是非常值得研究的一个方向，未来我们将会将研究重点放在这一领域。

# 参考文献

[1] 庞观松,蒋盛益.文本自动分类技术研究综述[J].情报理论与实践,2012,35(2):123-128.

[2] Suen ChingY. N-Gram Statistics for Natural Language Understanding and Text Processing, IEEE Trans. on Pattern Analysis and Machine Intelligence , Vol. PAMI1, No. 2, April 1979, pp.164-172.

[3] 赵世奇，张宇.等基于类别特征域的文本分类特征选择方法[J]．中文信息学报，2005,19(6) : 21-27.

[4] Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification[J].Journal of Machine Learning Research 10:207-244,2009.

[5] R. A. Fisher. The use of multiple measurements in taxonomic problems[J]. Ann. Eugenics, 7:179–188, 1936.

[6] H P Luhn. Auto-encoding of documents for information retrieval systems. Modern Trends in Documentation, New York, Pergamon Press, 1959.

[7] M E Maron, J L Kuhn. On relevance, probabilistic indexing and information retrieval. ACM, 1960, 7(3): 216-244.

[8] Rosenblatt.F. The Perceptron:A Probalistic Model for Information Storage and Organization in the Brain[J].Psychological Review,65.1958,pp.386-408.

[9] G.Salton, A.Wong, and C.Yang. A vector space model for automatic indexing. Communications of the ACM, 1975,18(11):613–620.

[10] D．D．Lewis．Naive(Bayes)at forty：The Independence Assumption in Informationc Retrie.In Proceedings of the 10th European Conference on Machine Learning,New York,1998,4-15．

[11] Lewis, D. D. and Ringuette, M. A comparison of two learning algorithms for text categorization. In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval (Las Vegas, US, 1994), pp. 81–93.

[12] T Joachims. Text categorization with support vector machine: learning with many relevant features. Proceeding of the 10th European Conference on Machine Learning, 1998: 137-142.

[13] C J C Burges. A Tutorial on Support Vector Machines for Pattern Recognition. Data

Mining and Knowledge Discovery. 1998, 2: 121-167.
[14] S. Tan, An effective refinement strategy for KNN text classifier, Expert Systems with Applications 30 ,2006,290–298.
[15] Y Yang. An evaluation of statistical approaches to text categorization. Journal of Information Retrieval.1999, 1(1/2): 69-90.
[16] T Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. International Conference on Machine Learning (ICML), 1997.
[17] Behara, R. S., Fisher, W. W. and Lemmink, J. Modelling and Evaluating Service Quality Measurement Using Neural Networks, International journal of operations and production management.2002,22, 10, 1163-1185.
[18] T.-K. Kim, S.-F. Wong, B. Stenger, J. Kittler, and R. Cipolla. Incremental linear discriminant analysis using sufficient spanning set approximations. In Proc.Computer Vision and Pattern Recognition, 2007.
[19] J Goldberger, S Roweis, G Hinton, and R Salakhutdi-nov. Neighbourhood components analysis. In Advances in Neural Information Processing Systems 17, L. K. Saul, Y. Weiss, and L. Bottou, Eds: MIT Press ,2005: 513-520.
[20] 侯汉清. 分类法的发展趋势简论[J] .情报科学, 1981, (1) :58- 68,30.
[21] 黄萱菁,吴立德. 基于向量空间模型的文档分类系统[J].模式识别与人工智能 1998(2).
[22] 李荣陆,王建会,陈晓云,陶晓鹏,胡运发.使用最大熵模型进行中文文本分类[J].计算机研究与发展,2005,42(1):94−101.
[23] 董小国,丁冉.IDS 自适应特征选择算法-进化包装(Wrapper)算法分析[J].微计算机信息,2006,11-3:46-48.
[24] 于一.K-近邻法的文本分类算法分析与改进[J].火力与指挥控制,2008,33(4):143-145.
[25] 朱靖波,王会珍,张希娟等.面向文本分类的混淆类判别技术[J].软件学报,2008,19(3):630-639.DOI:10.3724/SP.J.1001.2008.00630.
[26] 李文波,孙乐,张大鲲等.基于 Labeled-LDA 模型的文本分类新算法[J].计算机学报,2008,31(4):620-627.
[27] 罗长升,段建国,郭莉等.基于推拉策略的文本分类增量学习研究[J].中文信息学报,2008,22(1):37-43.DOI:10.3969/j.issn.1003-0077.2008.01.006.
[28] Fisher, R.A. The use of multiple measurements in taxonomic problems. Annals of

Eugenics, 1936,7: 179-188.
[29] W. L. G. Koontz and K. Fukunaga. A nonlinear feature extraction algorithm using distance information.IEEE Trans. Computers, 1972,21(1):56-63.
[30] Y. LeCun P. Y. Simard and J. Decker. Efficient pattern recognition using a new transformation distance. InNIPS,1993,volume 6, page 50-58.
[31] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. IEEE Trans. Pattern Anal. Mach. Intell,1996, 18(6).
[32] L.Yang and R.Jin. Distance metric learning:A comprehensive survey.http://www.cse.msu.edu/~yangliu1/frame survey v2.pdf.2006.
[33] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning,with application to clustering with side-information, in Proceedings of Advances of Neural Information Processing Systems, 2003.
[34] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. InProc. Int. Conf. on Mach. Learn. 2003.
[35] S. C. H. Hoi, W. Liu, M. R. Lyu, and W.Y. Ma. Learning distance metrics with contextual constraints for image retrieval. InProc. Computer Vision and Pattern Recognition, 2006.
[36] Tomer Hertz , Aharon Bar-Hillel , Daphna Weinshall. Boosting margin based distance functions for clustering, Proceedings of the twenty-first international conference on Machine learning, p.50, July 04-08, 2004, Banff, Alberta, Canada.
[37] Masashi Sugiyama. Local fisher discriminant analysis for supervised dimension-ality reduction. InProc. Int. Conf. on Machine Learning, 2006.
[38] L. Yang, R. Jin, R. Sukthankar, and Y. Liu, An efficient algorithm for local distance metric learning, in Proc. AAAI Conf. Artificial Intell., Boston, MA, 2006, pp. 543–548.
[39] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. InProc. Int. Conf. on Mach. Learn., 2003.
[40] J.B. Tenenbaum, V. de Silva, and J. C. Langford. A globalgeometric framework for nonlinear dimensionality reduction. Science, 2000,290.
[41] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation.Neural Computation, 2003,15(6).
[42] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. InScience, 2000.

[43] 黄昌宁，赵海．中文分词十年回顾[J]. 中文信息学报,2007,21(3):8-19.DOI:10.3969/j.issn.1003-0077.2007.03.002.
[44] Wong, P. and Chan, C., Chinese word segmentation based on maximum matching and word binding force. in Proc. of the 16th conference on Computational linguistics, 1996.
[45] 杨晓恕,蒋维,郝文宁等.基于本体和句法分析的领域分词的实现[J].计算机工程,2008,34(23):26-28.
[46] 褚颖娜,廖敏,宋继华等.一种基于统计的分词标注一体化方法[J].计算机系统应用,2009,18(12):55-58.
[47] G.W. art. To decode short cryptograms [A]. Communications of the ACM[C]. New York , Association for Computing Machinery , 1994. 102 -108.
[48] Van Rijsbergen C J. Information retrieval [M]. London , Butterworths Scientific Publication , 1975.
[49] 顾益军,樊孝忠,王建华等.中文停用词表的自动选取[J].北京理工大学学报,2005,25(4):337-340.DOI:10.3969/j.issn.1001-0645.2005.04.014.
[50] 毛勇,周晓波,夏铮等.特征选择算法研究综述[J].模式识别与人工智能,2007,20(2):211-218.
[51] Kira.K and Rendell L.A. The feature selection problem Traditional methods and a new algorithm In Tenth National Conference on Articial Intel ligence. MIT Press. 1992a,129-134.
[52] 李凯齐, 刁兴春, 曹建军. 基于信息增益的文本特征权重改进算法[J]. 计算机工程, 2011, 37(1): 16-18.
[53] 徐峻岭,周毓明,陈林等.基于互信息的无监督特征选择[J].计算机研究与发展,2012,49(2):373-382.
[54] 单松巍,冯是聪,李晓明等.几种典型特征选取方法在中文网页分类上的效果比较[J].计算机工程与应用,2003,39(22):146-148.
[55] Koller, D. and Sahami, M. Hierarchically classifying documents using very few words. In Proceedings of ICML-97, 14th International Conference on Machine Learning (Nashville, US, 1997), pp. 170-178.
[56] Liu, H. and Setiono, R. A probabilistic approach to feature selection-a filter solution. In Proceedings of the Thirteenth International Conference on Machine Learning. 1996. pages 319-327.
[57] T. M. Cover. Estimation by the nearest neighbor rule. IEEE Trans. on Information

Theory, 14(1):50-55, 1968.
[58] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classication. In AAAI-98 Workshop on Learning for Text Categorization, 1998.
[59] Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A. and Vapnik, V. Support vector regression machines. Advances in Neural Information Processing Systems, 1997,9:155-161.
[60] 宋枫溪,高林.文本分类器性能评估指标[J].计算机工程,2004,30(13):107-109，127.
[61] 张巍．基于 K 近邻分类准则的特征变换算法研究，复旦大学博士论文，2007．
[62] Liu Yang. The Connection Between Manifold Learning and Distance Metric Learning. http://www.cs.cmu.edu/~liuy/lle_isomap_metric.pdf.2007.
[63] Fabrizio Sebastiani. Machine learning in automated text categorisation. Technical Report.IEI-B4-31-1999, Istituto di Elaborazione dell'Informazione, 2001.
[64] 李荣陆,胡运发.基于密度的 kNN 文本分类器训练样本裁剪方法[J].计算机研究与发展,2004,41(4):539-545.
[65] Weinberger, K. and Chapelle, O. Large margin taxonomy embedding with an application to document categorization [C].Vancouver, British Columbia, Canada: Advances in Neural Information Processing Systems 21, ,2009, pages:1737–1744.
[66] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification [J]. The Journal of Machine Learning Research, ,2009,10:207–244.

# 致谢

时光荏苒、岁月如梭，转眼间两年半的研究生生涯快要结束了，这两年半的时间了我学到了很多，在这里我首先要向我的导师杨煜普教授致以我最衷心的感谢。在硕士期间，从论文的选题、内容及撰写无一不渗透着导师的心血。他严谨的治学态度和负责的敬业精神让我深深敬仰。我很幸运遇到了这样一位恩师，他不仅在知识上指导我，更重要的是教我怎样去做人做事。他是一位智慧的长者，在我遇到困难的时候，总是能够帮我理清脉络，给我启发，他敏捷的思维、深入的分析总是会让我茅塞顿开。

我想再次感谢杨老师老师，感谢他给予我学习上的帮助和指导。感谢汪伟和沈键同学，我们有缘在这里相聚，一起度过了一段美好的时光，和你们的友谊是我一生的财富。感谢李祥宝、季睿、李皎洁、张伟在科研中给予我的指导，感谢李楠、魏延、王毓、张峰华在平日里给予我帮助，大家共同营造了实验室的良好氛围，我在这里学习生活都充满了乐趣。

感谢电信学院 B1003292 班的全体同学在学习和生活中给予我的帮助，我们来自四面八方，汇聚到一起，我们都有自己为之奋斗的理想，我们将继续前行。

最后，我要感谢我的父母，你们永远是我最坚强的后盾，你们教会怎么样做人，引导我走上人生的道路，我唯有用一生去回报你们的养育之恩。

感谢所有帮助过我的人！

彭凯　谨致

于上海交通大学闵行校区

# 攻读硕士学位期间的主要学术成果

已完成的论文

[1] 彭凯，汪伟，杨煜普. 基于余弦距离度量学习的伪 K 近邻文本分类算法. 计算机工程与设计, 2012（已录用，拟在 2013 年 8 月发表）

[2] 彭凯，魏岩，杨煜普. 一种基于密度的大边界最近邻文本分类方法.计算机应用与软件，2012（已录用，拟在 2013 年 6 月发表）