南开大学

硕士学位论文

排序学习中的批量主动学习问题研究

姓名：蒯宇豪

申请学位级别：硕士

专业：模式识别与智能系统

指导教师：黄亚楼

20090401

摘 要

随着 Internet 技术的突飞猛进，Web 信息量爆炸性增长，人们越来越习惯使用搜索引擎查找所关心的信息。但浩瀚的信息资源却给搜索引擎的发展提出了新的挑战。如何有效、快捷、准确地将查询结果返回给用户，提高 Web 信息检索效果，已变成一项迫切而有意义的研究课题。

在现阶段信息检索领域的研究中，基于监督学习的排序学习逐渐成为排序研究的热点。基于监督学习的排序学习需要大量的人工标注的样本，为了减少人工标注样本的标注量，产生了一些基于“选择最值得标注的样本进行标注”思想的所谓主动排序学习算法。通过主动排序学习算法，用户不需要一开始标注所有的样本，而是开始只标注一部分样本，先学习得到一个排序模型；然后每次从剩下的未标注样本中选择一个最值得标注的样本进行标注，把这个新标注的样本放入训练集中，重新训练得到新的排序模型；然后在剩下的未标注样本中再重新选择一个样本进行标注，加入训练集，如此类推直到得到最终的排序模型。主动学习减少了排序学习的样本标注量，但此方法有一个问题是每次只选择一个样本标注，之后又要重新训练，训练需要很多时间，同时标注人员标注下一个样本需要等待很长时间。如果每次可以选择多个样本，则可以减少整个主动排序学习的时间，降低标注人员的工作量，即标注代价，同时，如果有多个标注人员的话，还可以实现并行标注，提高主动排序的效率。

针对上述问题，本文提出批量主动排序学习的思想，主动排序学习的时候，一次能够找到多个值得标注的样本给用户标注，这多个标注的样本对排序模型性能的提升有很大的价值。

本文提出了两种批量主动排序学习算法，一种是基于夹角差异的批量主动排序学习算法，该算法通过加入批量选择的样本之间的夹角差异度量，来减少批量选择的样本之间的相似度，提高批量主动排序的性能。另一种是基于损失函数的批量主动排序学习算法，该算法直接从提高排序模型性能的损失函数入手，批量选择能够使损失函数达到最小值的那些样本进行标注。

本文在不同数据集上进行实验评价以上两种批量主动排序学习算法，同时与单样本主动排序学习算法，原始的批量主动排序学习算法（直接用单个主动

排序学习算法选择多个样本）等进行比较分析。实验结果表明，本文提出的批量主动排序学习算法具有很好的性能。

关键字：排序学习，主动学习，批量主动，夹角差异，损失函数

Abstract

Along with the rapid growth of Internet technology, there is much information in Web environment. People are used to gain information with search engine. But there is new challenge for search engine. How to rapidly and exactly return search results to users, and how to enhance the effect of information retrieve, have become research topics and hotspot with impendency and significance.

In recent years, supervised rank learning gradually becomes research hotspot. Supervised rank learning needs many samples manually labeled by person, for reducing the size of samples needs labeled, there bring some algorithms based on the idea of "select the most valuable sample to label", its named active learning. Active learning has one issue, it only select one sample to label, then needs put this sample to training set to retrain, retraining costs many time, and also person must wait retraining for much time to label next sample. If active learning once selects more samples with batch mode to label, that will reduce both the time of active learning and the workload of person for label, namely label expense. Additionally, if there exists more people for label, people can label sample in parallel, boosting the efficiency of active learning.

To solve the problems mentioned above, in this paper, the idea of batch active learning is proposed, it solve how to select more samples with batch mode to label, these samples have good value for rank model.

In this paper, two batch active rank learning algorithm are proposed, one is batch active based on angle difference, this algorithm reduce similarity of the samples batch selected by adding angle difference of the samples. Another is batch active based on loss function, this algorithm directly searches loss function to enhance the capability of rank model, the samples batch selected are those which can make loss function to minimum.

In this paper, test of two batch algorithm is performing on two different and authoritative data sets, comparing analysis is conducted between the proposed

algorithms and single select active learning, general batch active learning and so on. Experiment result shows the algorithms mentioned in this paper have nice capability.

Keywords: Rank learning, Active learning, Batch Active, Angle difference, Loss Function

南开大学学位论文版权使用授权书

本人完全了解南开大学关于收集、保存、使用学位论文的规定，同意如下各项内容：按照学校要求提交学位论文的印刷本和电子版本；学校有权保存学位论文的印刷本和电子版，并采用影印、缩印、扫描、数字化或其它手段保存论文；学校有权提供目录检索以及提供本学位论文全文或者部分的阅览服务；学校有权按有关规定向国家有关部门或者机构送交论文的复印件和电子版；在不以赢利为目的的前提下，学校可以适当复制论文部分或全部内容用于学术活动。

学位论文作者签名：

年　月　日

经指导教师同意，本学位论文属于保密，在　　年解密后适用本授权书。

指导教师签名：		**学位论文作者签名：**	
解　密　时　间：	年　　月　　日		

各密级的最长保密年限及书写格式规定如下：

内部　5年（最长5年，可少于5年）
秘密★10年（最长10年，可少于10年）
机密★20年（最长20年，可少于20年）

南开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，是本人在导师指导下，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，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：

年　　月　　日

第一章 绪 论

1.1 引言

随着 Internet 技术的突飞猛进，Web 信息量爆炸性增长，人们越来越习惯使用搜索引擎查找所关心的信息。但浩瀚的信息资源却给搜索引擎的发展提出了新的挑战。如何有效、快捷、准确地将查询结果返回给用户，提高 Web 信息检索效果，已变成一项迫切而有意义的研究课题。

信息检索的主要研究内容包括对信息的表示、存储、组织和访问，其目的在于让用户更加容易的访问到所需要或者感兴趣的信息。信息检索的过程可以简单地描述为：用户根据其信息需求，组织一个查询字符串提交给信息检索系统，信息检索系统在文档集中检索出与查询相关的文档子集返回给用户。由于信息量巨大，信息检索系统检索出的相关文档数量相当的多，为便于用户尽快地获得最相关的文档，大多数检索系统都把检索结果以其与用户提交的查询的相关程度进行排序后返回给用户。因此，如何准确、高效地为检索出的文档进行排序已成为信息检索研究的核心问题之一。

在现阶段的研究中，基于监督学习的排序学习逐渐成为排序研究的热点。基于监督学习的排序学习需要大量的人工标注的样本，为了减少人工标注样本的标注量，产生了一些基于“选择最值得标注的样本进行标注”思想的称为主动排序学习算法。通过主动排序学习算法，用户不需要一开始标注所有的样本，而是一开始只标注一部分，先学习得到一个排序模型，然后每次从剩下的未标注样本中选择一个最值得标注的样本进行标注，把这个标注的样本放入训练集中重新训练得到新的排序模型；然后在剩下的未标注样本中再重新选择一个样本进行标注，加入训练集，如此类推直到得到最终的排序模型。但此方法有一个问题是每次只选择一个样本标注，而后又要重新训练，训练需要很多时间，同时标注人员需要等待很长时间。如果每次可以选择多个样本，则可以减少整个主动排序学习的时间，降低标注人员的工作量，即标注代价，同时，如果有多个标注人员的话，还可以实现并行标注，提高主动排序的效率。

1.2 研究现状

本文的工作基于信息检索领域已有的很多关于信息检索，排序学习以及主动学习等的研究成果。在本节中，将对这些相关的研究成果做简单介绍。

1.2.1 机器学习

自从计算机问世以来，人们就想知道它们能不能自我学习，这在目前还并没有确切的答案，但一些针对特定学习任务的算法已经产生。人们开发出很多实践性的计算机程序来实现不同类型的学习，一些商业化的应用也已经出现。例如，对于语音识别这样的课题，迄今为止基于机器学习的算法明显胜过其他方法。随着对计算机认识的日益成熟，机器学习必将在计算机科学和技术中扮演越来越重要的角色。

机器学习的定义为：对于某类任务 T 和性能度量 P，如果一个计算机程序在 T 上以 P 衡量的性能随着经验 E 而自我完善，那么我们称这个计算机程序在经验 E 中学习。

从特殊的训练样例中归纳出一般的函数模型是机器学习的中心问题。要让机器可以进行学习，首先要获得已经经过适当的预处理并且被人为标注了的数据，这些数据的内容本身蕴含了数据的特征，大量数据形成的数据集往往代表了数据的客观分布，而所标注的结果正是机器学习所要归纳出的目标函数的函数值。在学习过程中，这些数据通常会被分为两部分，它们分别用来作为训练数据和测试数据。对于训练数据，计算机通过某种机器学习算法根据其内容和标注结果学习得到一个包含了目标函数的模型 M，此过程即学习过程。不同的学习算法所获得的模型是不同的，它们是对数据从不同侧面的描述。在得到了模型 M 之后，需要使用已获得的模型 M 应用目标函数去处理另一组用于测试的数据，以得到函数的结果，并通过把此结果与已通过人为标注所得的结果对比来评价模型的性能度量 P。如果对测试结果不满意，应该进行算法的参数或其他方面的调整，并重新进行学习，必要时可能还需要修正甚至更换学习算法。当得到了满意的模型之后，即可使用此模型对需要预测的数据进行处理了。

机器学习有多种常用的算法，每一种算法都有自己的适用环境，对于某一给定的数据集，使用适合它的算法可以得到令人较为满意的结果，而如果使用了不合适的算法可能会得到比较差的结果。这是由算法本身的性质决定的。由

不同的算法训练所得的模型从不同的角度去描述原始数据，而如果客观上从此角度并不能很好的区分数据，那么学习的结果也会比较差，性能不高。常见的算法有决策树、人工神经网络、贝叶斯、遗传算法、支持向量机等。这些算法都在各自的适用领域中有着成功的应用。

机器学习的方法在数据挖掘特别是分类问题中有着相当成功的应用，从包含设备维护记录、借贷申请、金融交易、医疗记录等信息的大型数据库中发现有价值的信息，都可以通过机器学习的方法来实现。本文中主要涉及的自然语言理解中文本信息提取的问题也将用到机器学习的方法。

1.2.2 排序学习

排序学习（Learning to Rank）旨在为目标对象按照某种规律确定一个等级顺序。排序学习在许多领域有着非常广泛的应用，例如在信息检索中，信息需要按照其与查询的相关程度进行排序；在金融银行中的信用等级的判定需要用到排序学习模型；在经济学领域中，各种经济学模型常常要用到排序学习；在传统的统计领域里面，排序模型也经常被用到。

解决排序学习问题的机器学习方法大致可以分为三个大类：基于回归的排序学习[22]、基于分类的排序学习[12]和基于顺序回归（Ordinal Regression）的排序学习。

解决排序学习问题最简单的方法就是把它看成传统的回归问题，把序列标号转化成实数。基于分类的排序学习是把排序问题分解为一系列嵌套的二值分类问题，这些二值分类问题的解中包含了排序信息，对这些解进行某种组织分析，从而得到最终的排序。基于顺序回归的排序学习算法是当前排序学习研究的热点，根据对训练数据处理手段的不同又可以分成三大类别：基于数据点（Point-wise）的排序学习算法、基于有序对（Pair-wise）的排序学习算法以及最新提出的基于列表（List-wise）的排序学习算法。

基于数据点的排序学习代表算法包括：Crammer 和 Singer 提出的基于感知机的排序感知机算法[9]（Perceptron Rank, PRank）。Harrington 对 PRank 算法进行改进，提出了其最大化边缘（Large Margin）的在线版本[15]。ShaShua 和 Levin 提出了一种推广的支持向量机版本[40]，替代感知机算法来解决排序学习问题。Chu 和 Keerthi 把阈值大小的约束融入了支持向量机的学习中，提出支持向量顺

序回归（Support Vector Ordinal Regression）算法[6]取得了更好的排序效果。

基于有序对的排序算法的思想是在有序对空间中构建排序模型。Herbirch等把对目标数据点的排序问题转换为基于有序对数据的二值分类问题，并且用结构风险最小求解分类问题的解[17]，从而得到排序模型。Joachims基于以上的排序学习思想，提出了用点击序列数据优化搜索引擎的新方法[21]，取得了很好的效果，其开发出的SVM^{light}工具包包含了上述排序模型，称为Ranking SVM。Freund等人基于Boosting思想，提出了RankBoost算法[14]。Burges等基于有序对数据，提出了基于交叉熵的损失函数，并用神经网络进行优化（RankNet）[3]。

基于列表的排序学习方法以一个列表为基本的学习单元，取得了比基于有序对的排序学习方法更好的效果。Cao等在2007年第一次提出基于列表的排序学习方法[5]，并给出其损失函数和对应的评价指标；Qin等在[33]提出使用余弦相似度度量列表之间的差异。

目前所有的排序学习方法都是基于有监督学习的方法，需要大量的人工标注样本。在信息检索中，数据的标注代价是非常昂贵的。这也限制了排序学习的发展和应用。

1.2.3 主动学习

传统的基于有监督学习的机器学习方法都需要大量的人工标注样本，而标注的代价是非常昂贵的。近年来，很多学者提出了诸多方法解决这一问题。有代表性的方法包括：自学习[50]，半监督学习[2][51][53]，主动学习[8][13][25][42]，多视图学习[28]，直推式学习[20]等等，这些方法在诸如文本分类、图像处理等领域都取得了良好的效果。

主动学习（Active Learning）是解决训练样本获取代价过大的一种有效方法。主动学习摒弃了传统的将机器学习系统视为纯粹被动的样本接受者的观点，认为学习系统能够利用其自身已有的信息主动的搜集或者查询新的样本以改善其性能。主动学习研究的重点在于学习系统如何利用自身的能力，以尽可能少的步骤和尽可能低的标注代价实现性能的有效提升。

主动学习的核心问题在于查询准则Q的设计和选择。现有主动学习的查询函数的设计思想主要可分为两类。第一类是利用统计学习的思想，Cohn等人提出基于统计学习（Statistical Learning）思想的查询函数最小化期望误差[8]。第

二类是通过查询函数挑选出信息量最大的样本交由人工标注。这类思想早期的文献如 Freund 等人提出基于委员会（Query by Committee）方法的主动学习算法[13]。Tong 等人于 2000 年提出基于版本空间（Version Space）的查询函数，使用支持向量机（Support Vector Machine, SVM）作基本学习模型的主动学习方法[42]，并应用于文本分类中。Lewis 和 Catlett 提出找最不确定（Least Certain）的样本作为查询函数要找的样本[25]。

目前，主动学习已经被广泛的应用于分类等机器学习问题中。但到目前为止，还未见有用于排序学习的主动学习成果发表。

1.2.4 批量主动学习

传统的主动学习的模式是一次只选择一个最值得标注的样本进行标注，然后把这个标注的样本加入到训练集重新训练得到新的排序模型，依次迭代，最终得到满意的排序模型。

这种主动学习模型需要的迭代次数多，每次迭代都需要对训练集重新学习得到新的模型，一般重新训练学习的时间都比较长，例如排序学习中用排序支持向量机算法训练一个规模几千条查询-文档对的训练数据需要几个小时，如果迭代次数很多的话，训练的时间代价将很大。

从标注人员的角度来看，标注人员一次只标注一个样本，然后需要等很长时间才可以标注下一个样本，这样主动学习的标注代价仍然很大。同时，如果存在多个标注人员的话，一次只有一个样本需要标注，也不能让多个标注人员并行标注，提高主动学习的效率。

为了解决上述问题，一些学者提出了批量主动学习的思想[57][58][59][60]，主动学习的时候一次可以选择多个值得标注的样本让标注人员标注，这样可以减少主动学习的迭代次数和学习时间，降低标注人员的标注代价，同时在条件允许的情况下，实现并行标注，提高主动学习的效率。

目前，在分类学习问题中有学者提出了一些批量主动学习的算法[57][58][59][60]，在排序学习问题中还没有学者提出批量主动的思想和算法。

1.3 本文主要研究内容

本文主要研究排序学习中的批量主动排序学习算法，提出并实现基于夹角差异的批量主动排序学习算法和基于损失函数的批量主动排序学习算法。具体讲，论文将在以下两个个方面开展研究工作：

1.3.1 基于夹角差异的批量主动排序研究

每次选择离 SVM 决策面最近的一个未标注样本进行标注，就可以把版本空间的矢量数量降低一半，从而提高模型的泛化性。但每次只能选择一个未标注样本进行标注，然后需要将人工标注过的样本加入训练集后重新训练求得 SVM 决策面，求 SVM 决策面需要的时间复杂度比较高，这样的话极大的影响了主动学习所发挥的作用。研究怎么能够一次选择多个未标注样本进行标注有很重要的意义。

现存在的选择多个未标注样本的方法有，选择离 SVM 决策面距离的和最小的多个未标注样本进行标注。但这样并不能保证版本空间的数量降低的最多，从而不能保证能够提高模型的泛化性。

论文将提出一种基于夹角差异的方法，找出夹角差异最大的样本作为“最值得标注”的样本，完成版本空间批量主动排序算法的研究与实现。

1.3.2 基于损失函数的批量主动排序研究

基于夹角差异的批量主动排序算法是一种启发式的批量选择算法，需要先选择一个，然后选择下一个，以此启发式的批量选择多个未标注样本。

本文继而提出一种基于损失函数的批量主动排序算法，直接从能够提高排序模型性能的损失函数入手，批量选择的样本理论上能够使排序模型的性能提升最高。基于损失函数的批量主动算法选择样本的依据在于使损失函数取得最小值，即批量选择的未标注样本能使损失函数达到最小值。基于损失函数的批量主动学习算法是一下子选择多个未标注样本，而不是启发式的，是一个新的思路，并且有很强的理论性。

为了验证本文提出的两种批量主动排序学习算法，设计和实现了主动排序学习实验算法。批量主动排序学习实验算法采用排序学习领域两个权威的数据集 OHSUMED 和 TREC.Gov，设计了批量主动学习对比实验流程，采用模型编程，实验时只要执行一遍算法，就可以得到算法的中间和最终对比结果，具有

很高的实验效率，方便了批量主动排序算法的研究和分析。

通过在两个大规模真实数据集上的实验表明，使用本文提出的算法可在保证排序模型性能的前提下，减少样本的标注量；在同等标注量的条件下，提高排序结果的正确率。

1.4 本文结构

本文共分为六章，各章节内容和结构安排如下：第一章是绪论，概括介绍本文的研究背景、研究内容和研究目标。第二章综述相关工作，介绍信息检索，排序学习，降低标注代价学习和主动学习模型。对信息检索，排序学习，降低标注代价学习以及主动学习等领域现有算法进行综述，作为后续章节工作的基础。第三章详细介绍启发式的批量主动排序学习算法，并在此基础上提出基于夹角差异的批量主动排序学习方法。描述基于夹角差异的批量主动排序学习的排序决策函数，查询函数，更新过程以及算法流程。第四章详细直接的批量主动排序学习算法，并在此基础上提出基于优化函数的批量主动排序学习方法。描述基于优化函数的批量主动排序学习的排序决策函数，查询函数，更新过程以及算法流程。第五章是总结与展望，总结本文工作，并展望下一步的研究工作。

第二章 相关工作综述

本文的研究得益于信息检索领域和机器学习领域的相关研究成果，本章详细介绍与本文相关的研究工作和成果，包括信息检索模型、排序学习模型、机器学习中降低标注代价方法和主动学习方法，作为后续章节工作的基础。

2.1 信息检索

信息检索是指从大量非结构化的文档集合中找出与用户给定查询相关的文档子集，是处理海量文本的重要手段。

一个文档检索的基本过程通常包括以下三个步骤：首先，用户可以从某一终端将其查询输入到检索系统中；之后，检索系统针对用户的查询，通过适当的算法，在已经建立了索引的文档集中进行检索，获得与用户查询相关的文档集；最后，检索系统为用户提供与其查询相关的文档集。通常，检索系统将所提供的相关文档集按照与用户查询的相关程度进行排序，最相关的文档排在最前面。

根据对相关文档判定方法的不同，信息检索模型可以分为以下五类经典模型：布尔模型、向量空间模型、概率统计模型、统计语言模型、基于有监督学习的检索模型等，以下将分别介绍这些检索模型。

2.1.1 布尔模型

布尔模型[24]（Boolean Model）是一种建立在集合论和布尔代数上的比较简单的检索模型。在经典布尔模型中，文档被表示成词项的集合。每个词项在每篇文档中只有两个值 1 和 0：“1”表示该词项出现在该文档中；“0”表示未出现

在该文档中。用户的查询用布尔表达式的形式来进行描述，支持逻辑与（AND）、逻辑或（OR）和逻辑非（NOT）的操作，只有满足该布尔表达式的文档才被认为是相关的文档，否则就是不相关的。

经典布尔模型存在着一些缺陷，主要问题包括：没有提供词项的权重信息，

检索系统无法区分文档中不同的词项对相关性的贡献；检索系统不能提供相关度的排名；对于相关性的二值判定过于严格等。

为解决经典布尔模型的诸多问题，研究者们提出了扩展布尔模型（Extended Boolean Model）。在该模型中，文档和查询的相关度不再是 0 和 1，而是区间[0,1]中的一个实数，从而使得对文档的相关度排名成为可能。

2.1.2　向量空间模型

向量空间检索模型[37][38]（Vector Space Model）是信息检索领域中广泛使用的一种信息检索模型。其基本思路是：在信息检索中，文档或者查询的基本含义都是通过其所包含的词（检索单元）来表述的，可以定义由检索单元组成的向量来描述每一篇文档和每一条检索，再通过计算文档与查询之间的相关程度来判断文档与查询是否相关，与某一特定的查询的相关程度越高者被认为是与该查询越相关的文档。对于向量空间检索模型，需要定义向量来描述文档和检索的含义。通常的做法是，以所有包含在文档和查询中的检索单元为检索空间，将文档和查询以向量的形式表示出来。

向量空间检索模型通常使用基于文档集合的统计频率的权值，也被称为 tf-idf 权值。tf-idf 权值由两部分组成，一部分是检索单元在文档中出现的频率（term frequency, tf），另一部分则被称为文档频率的反转（inverse document frequency, idf），通常，对于一个给定的检索单元 tf-idf 权值是 tf 与 idf 的乘积。

为了方便说明问题，作如下定义：

m:　整个检索空间Ω的大小。

d:　文档集合中文档的总个数。

tf_{ij}: 检索单元 t_j 在文档 d_i 中出现的次数（tf）。

df_j: 在整个文档集合中，包含检索单元 t_j 的文档的个数（df）。

则文档频率的反转定义为：

$$idf_j = \log\left(\frac{d}{df_j}\right) \tag{2.1}$$

对于给定的某一个文档，描述该文档的向量由 m 个元素组成，分别对应着文档中出现的 m 个检索单元。每一个元素的权值根据其所对应的检索单元在文档中出现的频率以及该检索单元在整个文档集中出现的频率两项因素共同决定

$$w_{ij} = tf_{ij} \cdot idf_j \tag{2.2}$$

使用公式（2.2）中的ω_{ij}作为向量中各元素的权值，对前面所述的向量进行进一步调整，这样的向量更精确地描述了文档和查询的内容。

对于向量空间检索模型，不仅需要定义向量来描述文档和查询的含义，还需要选择适当的方法来计算文档与查询的相关程度以判断文档与查询是否相关。

原则上讲，只要是能够判断出描述文档与查询的各个向量方向的接近程度各种计算方法都可以用来作为文档与查询相关程度的判断依据。很自然的，可以考虑使用向量夹角的余弦作为文档与查询相关程度的判断依据。如前所述，在检索空间Ω 中，定义文档D 和查询Q 的相似度（Similarity Coefficient, SC）为：

$$SC(q,d) = \frac{\sum_{i=1}^{m} w_{qi} \cdot w_{di}}{\sqrt{\sum_{i=1}^{m} w_{qi}^2 \times \sum_{i=1}^{m} w_{di}^2}} \tag{2.3}$$

其中，ω_{qi}和ω_{di}是对该文档含义的一系列描述。当检索单元t_i出现在文档D中时，$\omega_{d,i}$为1，否则为0；当检索单元t_i出现在查询Q中时，$\omega_{q,i}$为1，否则为0。ω_{qi}和ω_{di}都采用*tf-idf*权值。

2.1.3 概率检索模型

概率统计检索模型[35]（Probabilistic Retrieval Model）是另一种普遍使用的信息检索算法模型，它应用文档与查询相关的概率来计算文档与查询的相似度。通常，利用检索单元作为线索，通过统计得到每个检索单元在相关的文档集（对应于某查询）中出现和不出现的概率以及其在与该查询不相关的文档集中出现和不出现的概率，最终，利用这些概率值，计算文档与查询的相似度。

BM250 检索算法是一种经典的概率统计检索算法，由 Roberston 1994 年在 TREC-3 上提出，BM25 计算文档 D 和查询 Q 的相似性。对查询 Q 中的每一个检索单元ω_i，一共有三个权值U, V, W 与之相关：

$$U = \frac{(k_2+1)\psi}{k_2+\psi} \tag{2.4}$$

其中：k_2是由用户指定的参数，ψ是检索单元ω_i在 Q 中出现的频率 qtf（within query frequency）。

$$V=\frac{(k+1)\phi}{k(1-b+bL)+\phi} \tag{2.5}$$

其中：k 和 b 是用户指定的参数，ϕ是检索单元ω_i在 D 中出现的频率 tf（within document frequency），L 是正则化之后的文档长度，计算方法为原始文档长度除以文档集合中平均的文档长度。

$$W=\log\left(\frac{r+0.5}{(R-r)+0.5}/\frac{(n-r)+0.5}{(N-n)-(R-r)+0.5}\right) \tag{2.6}$$

其中：N 表示文档集合中文档的总数；R 表示与查询 q 相关的文档总数；n 表示含有检索单元ω_i的文档总数；r 表示与 q 相关的文档中，含有检索单元ω_i的文档数。

这样，在 BM25 公式中，查询 Q 和文档 D 的分值为：

$$SC(Q,D)=\sum_{\omega\in Q}UVW \tag{2.7}$$

近年来，Robertson[36]等提出了一种简单的基于 BM25 的改进，改进算法能够同时计算具有多个域的文档和查询的相似度，克服了 BM25 在这方面的不足。

2.1.4 统计语言模型

近些年来，统计语言模型（Language Model）在信息检索领域取得了令人瞩目的效果。1998 年，Ponte 和 Croft 在 SIGIR 会议上发表了一篇名为“A Language Modeling Approach to Information Retrieval”[31]的论文，由此开创了一个新的研究课题：统计语言模型在信息检索中的应用。随后几年，众多研究人员不断加入到该课题的研究工作中来，取得了丰硕的研究成果。许多实验结果显示：基于统计语言模型的方法在检索性能上普遍优于以前普遍采用的向量空间模型方法。

利用语言模型进行信息检索，查询 Q 一定时，检索出的文档 D 根据后验概率 $P(D|Q)$ 来排序。根据贝叶斯公式可得：

$$P(D\,|\,Q)\propto P(D)P(Q\,|\,D) \tag{2.8}$$

这便对应了一个信源-信道模型：信源模型 $P(D)$ 和信道模型 $P(Q|D)$，该方法认为：如果文档 D 和查询 Q 相似度越高，则通过观察信道的输出（Q）便能获得信源（D）的更多信息，这也是通过语言模型方法进行信息检索的一个基本假设。

利用语言模型进行信息检索的基本过程是：对每一篇文档均建立一个模型，计算每一个模型产生某主题（查询）的概率值（相当于其他模型中文档-查询相似度），然后对这些概率值进行排序，返回排序结果，即为该主题的检索结果。

基于有监督学习的排序学习模型近年来逐渐成为信息检索领域研究的热点，本文在下一节进行详细的介绍。

2.2 信息检索评价指标

在信息检索的研究工作中，检索性能的评价是一个非常重要的问题，它是衡量各种检索模型好坏的量化指标。

由于用户查询条件中所固有的模糊性，信息检索系统检索出来的文档集合不一定全是用户所希望的，因此有必要对这些文档集合根据其与用户查询条件的相关性进行排序，相关程度越高的文档排得越靠前为好，并以此来判定信息检索系统检索出的文档集合满足用户查询条件的程度，这种评测就是检索系统的检索性能的评测。

在本文的实验中，我们主要使用 MAP 和 NDCG 来评价排序结果序列的性能。

2.2.1 MAP

MAP（Mean Average Precision）是用来衡量算法对多个查询的平均排序结果。MAP 的计算公式为：

对于某一个查询 Q_i，其平均查准率计算公式为：

$$AvgP_i = \sum_{j=1}^{M} \frac{precision(j) \times pos(j)}{\text{number of documents relevant to } Q_i} \tag{2.9}$$

其中：j 表示排序的位置，M 是检索到的文档总数，Precision（j）是前 j 个检索到的文档的查准率，pos（j）是一个 0-1 函数，如果排在第 j 个文档是相关的，其值为 1，否则为 0。这样平均查准率的均值 MAP 的计算公式为：

$$MAP = \frac{\sum_i AvgP_i}{\text{查询的个数}} \tag{2.10}$$

在计算 MAP 时，由于其要求文档被标注成两个等级：相关和不相关，因此把标注为相关的文档（definitely relevant）看成相关的文档，其他两个级别的文档（部分相关（partially relevant）和不相关（not relevant））都看成不相关文档。

尽管 MAP 已广泛用作信息检索系统中检索算法的评测方法，但也有其限制：

1、MAP 将查询和文档的相关简化成为 0-1 关系，一个查询和一个文档要么相关，要么不相关。而实际上相关是一个程度的量，0-1 关系并不能准确的反映查询和文档的相关关系，例如在“相关”和“不相关”之间还可能存在着“部分相关”的文档。

2、在实际的检索中，用户往往只是浏览位于序列头部的结果，因此位于序列头部的结果对排序性能的影响越大。然而 MAP 并没有很好的解决这两个问题，因此，我们又使用 NDCG 来评价排序结果中顶部序列的准确性。

2.2.2 NDCG

NDCG（Normalized Discounted Cumulative Gain）对传统的评价标准做出了改进，这些改进基于以下两个原则：1、在信息检索中，相关可以分为多个级别，高度相关的文档比部分相关的文档更有价值，其在评价中应该赋予更大的权值；2、文档在序列中的位置越靠后，这个文档的价值越小，从用户的角度考虑，由于时间、精力以及从已经阅读过的文档中所得到了信息等原因，用户可能根本不会去看这些文档。NDCG 用来评价排序结果中顶部序列的准确性。

能根本不会去看这些文档。NDCG 用来评价排序结果中顶部序列的准确性。在这种评价方法中，每一个文档都对它所在的位置有一定的贡献，其贡献值与文档的相关度有关，然后，从 1 到 n 的所有的位置上的贡献值都被加起来作为最终的评价结果。这样，一个一定长度的文档序列被转换成了一个相关分值的序列。

给定一个排序后的文档序列，在第 r 位的 NDCG 值 NDCG@r 的计算公式为

$$NDCG@r = N_r \sum_{j=1}^{r} \frac{2^{r(j)}-1}{\log(1+j)} \tag{2.11}$$

其中：$r(j)$ 是第 j 个文档的级别，N_r 是归一化参数，它使得最优的排序的 NDCG@r 的值始终为 1；如果结果序列中文档的个数 n 要少于 r，则计算公式返回 NDCG@n 的值。

在计算 NDCG 时，我们把相关映射为数值 2、部分相关为 1、不相关映射为 0。

2.3　排序学习模型

排序学习问题和机器学习中的分类学习和回归学习有着密切的联系，但是排序学习又有自己的特点。排序学习介于分类学习和回归学习之间，与分类学习相比，排序学习的输出空间虽然也只包含了有限个元素，但是在元素之间定义了序关系，与回归学习相比，排序学习的元素之间没有定义度量。

2.3.1　排序学习形式化描述

下面，给出排序学习问题的形式化定义。给定一个输入向量 $\bar{x}$ 的集合 X

$$X = \{\bar{x}_1, \ldots, \bar{x}_m\} \subseteq \mathbb{R}^n \tag{2.12}$$

和其对应的标号

$$Y = \{y_1, \ldots, y_m\} \tag{2.13}$$

其中：m 表示训练样本的数目，n 表示输入向量的维度。S = (X, Y) 为某一分布 $P(\bar{x}, y)$ 的独立同分布（iid）的样本集合，也称为训练集合。独立同分布即意味着任意一个样本 $(\bar{x}_i, y_i)$ 既不依赖于其他样本，也不依赖于其下标 i。

排序学习的目的是寻找一个能够精确预测数据 $\bar{x}$ 的未知标号 y 的决策函数 $f: \mathbb{R}^n \mapsto Y$，也就是说，排序学习所学习的预测函数将最小化对排序的预测错误，预测错误的定义为 $f(\bar{x}) \neq y$ 的概率。

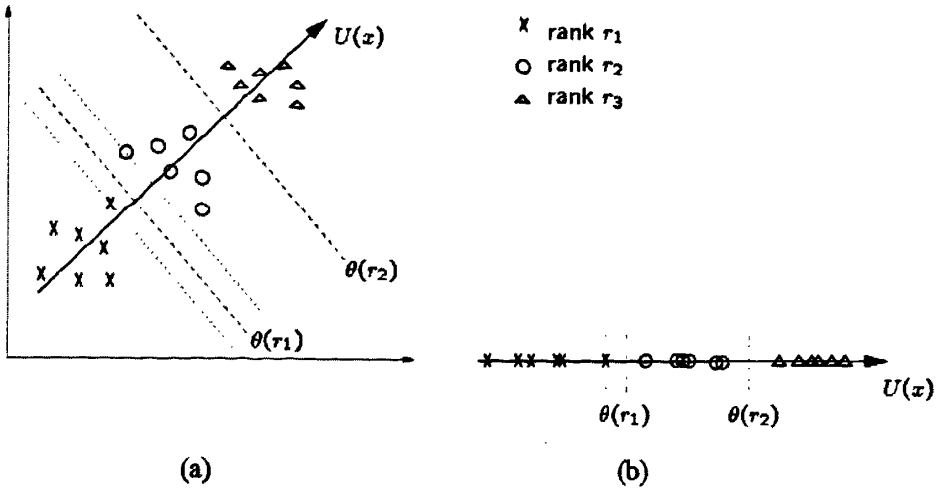

图 2.1 排序模型

图 2.1 （a） 用于排序的效应函数 $U(x)$，通过把数据点投影到 $U(x)$ 上可以得到对数据的排序，$\theta(r_i)$ 是各个序类别之间的边界。（b） 数据在 $U(x)$ 上的投影

如图 2.1 所示，在排序学习中，其对象按照效应函数（排序函数）$U(\vec{x})$ 进行排序：$U(\vec{x}):X\mapsto R$，每一个对象 $\vec{x}$ 的效应值为其在 $U(\vec{x})$ 上的映射值。

综上所述，排序学习中学习目的是从决策函数集合 $F=\{f:\mathbb{R}^n\mapsto Y\}$ 寻找一个最优决策函数 f^*，f^* 能够精确的预测数据点 $\vec{x}$ 的未知标号 y。

排序学习在近年来引起了机器学习研究者们的极大兴趣，有很多研究成果发表。用机器学习的方法来解决排序学习问题大致可以分为以下三类：基于回归的排序学习、基于分类的排序学习和基于顺序回归的排序学习。基于顺序回归的排序算法是当前排序学习研究的热点，根据对训练数据处理手段的不同又可以分成三大类别：基于数据点的排序学习算法、基于有序对的排序学习算法以及最新提出的基于列表的排序学习算法。

2.3.2 基于数据点的排序学习方法

基于数据点的排序学习方法的目标是找到一个排序函数，尽可能多的正确预测新样例的等级数。基于数据点的排序学习方法经过近几年的发展，相关算法研究已经有一定的基础，文献[9][15][40][6]都是基于数据点排序算法中的代表。

2001 年 Crammer 和 Singer 提出了用改进的感知机算法（Perceptron Rank, PRank）进行排序。排序感知机算法的学习目标是在特征空间中找到一个排序的方向和 k-1 个阈值，这 k-1 个阈值把空间划分成了 k 个连续的子空间，每一个子空间对应着一个序标号。在该算法的框架下，每一个样例都对应于一个等级，算法的目的就是要正确预测新样例的等级数。排序感知机算法是感知机模型的变形，但它包含了一系列用于相邻两个等级之间边界的偏置。之后 Harrington 在 2003 年用近似贝叶斯的观点对排序感知机算法提出了改进[15]，取得了很好的泛化性能。ShaShua 和 Levin 提出了一种推广的支持向量机版本[40]，替代了感知机算法来解决排序学习问题。Chu 和 Keerthi 把阈值大小的约束 $b_1 \le b_2 \le ... \le b_{k-1}$ 也融入到支持向量机的学习过程中，取得了更好的排序效果[6]。

虽然排序感知机算法及其扩展算法，在每个样例具有一个等级数的排序问题上取得了成功，但是它在处理类似文档检索排序时效果并不很好，限制了它们的应用。

2.3.3 基于有序对的排序学习方法

与基于数据点的排序算法不同，基于有序对的排序算法在有序对空间中构建排序模型，这种方法在信息检索等诸多领域得到了很好的应用。

Herbirch 等把对目标数据点的排序问题转换为基于有序对数据的二值分类问题，并且用结构风险最小求解分类问题的解[17]，从而得到排序模型。在这篇论文中提到的框架下，每一个训练样例与一个整数等级相关联，训练的目标函数就是要最大化相邻等级样例的间隔，并使用支持向量机计算最大化该间隔的线形函数；同时作者使用了有序样例对用于训练过程，也就是说如果有两个样例 $\overline{x_1}$ 和 $\overline{x_2}$，且它们各自的等级数为 i 和 i+1，那么 $\overline{x_1} - \overline{x_2}$ 就是正例，$\overline{x_2} - \overline{x_1}$ 就是负例。

Joachims 基于以上的排序学习思想，提出了用点击序列数据优化搜索引擎的新方法[21]，取得了很好的效果，其开发出的 SVMlight 工具包包含了上述排序模型，称为 Ranking SVM。

Burges 等基于有序对数据，提出了基于交叉熵的损失函数，并且用神经网络进行优化（RankNet）[3]，在信息检索的应用上取得了很好的性能。Freund 等

人基于 Boosting 思想，使用基于有序样例对的方法计算间隔，提出了 RankBoost 算法[14]。

2.3.4 基于列表的排序学习方法

基于列表的排序学习方法是最近一两年新提出的排序学习方法，是从基于有序对的排序学习方法发展而来的，其代表文献包括文献[5][32][33][48]。

基于列表的排序学习方法的初衷是：排序是一种关系的表现，不像以前比如分类、回归是一个物体或一个对象本身的属性。以网页处理为例：网页的分类问题，一个网页到底是讲新闻还是讲体育的是个绝对的事，拿到这个网页一切都知道了，是它的本身的属性。但网页排序是指这一个网页跟别的网页之间比较的一种关系。如果分类问题可以叫做一元学习，那么排序问题则是一个更高元的、更高阶的一个问题。

与之前基于有序对的排序学习方法不同的是，基于列表的排序学习方法是以一个列表为基本的学习单元。因为一个列表本身就包含了一些排了序的文档，某些关系已经嵌在这样的表达方式里，所以不需要像以前研究时的那种假设，文档之间会有相对大小的关系，这些都已经在以列表为单位学习单元里面了，这使得基于此的一些理论和实践都会比较顺畅，和以前有较大不同。

基于列表的排序学习方法之所以受到关注，是因为在评价排序结果好坏的时候，它把查询词对应的所有文档通盘考虑，全局衡量，而以前的工作把目光集中在单个文档或者一对文档之上；而且可以对文档之间的关系，如相似度等进行建模，因此可以定义更加有效的排序函数；另外，由于是列表级别，它可以充分利用文档在列表中的位置信息，因此可以更加强调排在前面的文档，而这与用户的体验更加一致。

基于列表的排序学习方法考虑了排序学习不同于原有机器学习问题的新特性，即不同查询对应的查询-样本对之间的差异是很大的，改变原有以有序对为基本样本单元的传统思想，以一个查询对应的所有查询-文档对列表为基本样本单元，在损失函数的构造上充分考虑了列表样本单元，取得了比基于有序对的排序学习方法更好的效果。Cao 等在 2007 年 ICML 会议上发表了一篇“Learning to Rank: From Pairwise Approach to Listwise Approach”[5]的论文，第一次提出基于列表的排序学习方法，并给出其损失函数和对应的评价指标；Qin

等在[33]提出使用余弦相似度度量列表之间的差异。这些算法都取得了比较好的排序效果。

排序学习的研究更多的是关心排序算法和排序模型的构建。比如在互联网搜索的时候，网页的重要度是一个重要的特征，但也要考虑相关度；只有重要度和相关度可能也不够，还要考虑其他的一些因素。人们已经慢慢意识到，有太多的因素会影响到排序结果。排序学习就是要把这些因素视作特征用一些方法综合考虑得出一个最合理的排序结果。

2.4 主动学习

2.4.1 主动学习模型

主动学习（Active Learning）就是在有监督学习时，从候选样本集中动态的选择样本用于训练。学习者用现在已有知识主动地选择最有可能解决问题的样本，而不是从指导者那被动地接受样本进行训练。这种允许学习者利用自身的知识动态的控制何种样本被选择，并指导搜索信息含量最大的样本的过程，就是主动学习的过程。一旦某种分类器被赋予这种功能，它就从一个被动的学习者变成了主动的学习者。

通常，主动学习模型由五部分组成（R, L, U, Q, S）。其中：R 是一个基本学习模型，L 是训练集中的已标注样本集，U 是训练集中的未标注样本集，Q 是查询函数，S 是可以为通过 Q 找出的未标注样本提供正确标签的指导者。

2.4.2 主动学习方法

目前主动学习的研究集中在查询函数 Q 的设计与实现上，即研究学习机器如何有效的搜集样本数据以提高自身的性能，使用查询函数，进行高效的查询，查找出那些“最值得标注”的样本，只对少量的查询的样本进行标注以减少代价。

2.4.2.1 基于统计的方法

基于统计的方法[8]（Statical Approach）主要研究学习模型的期望误差。因

为期望误差是不能直接计算的，所以就采用近似模型的方差代替。学习模型选择那些能最大降低方差的样本来学习。文献[8]提出了一种方法，它先假设分类器的偏置为 0，重点放在降低分类器的方差上，实现了一种在多层感知器神经网络（MLPNN）上估计方差的方法，从而选择哪些最能降低方差的样本训练分类器。

但此假设本身在现实中是不适用的，而且估计方差的方法也太复杂。

2.4.2.2 基于样本的不确定性方法

基于样本的不确定性[25]（Uncertainty Based Sampling）的方法有点类似于拿学习模型不易得出正确预测的样本进行训练的策略，但是它们之间还是有区别的。基于样本的不确定性要考虑这种情况，当这个样本的所属类别不清楚的时候，学习模型就要估计此样本是不是预测错了或者预测错的可能性。根据样本的不确定程度来选择样本，这个不确定程度又由学习模型决定。当一个样本的预测类别足够不确定的时候，就选择其加入训练集。学习模型通过这些不确定的样本学习的经验，估计下一个样本所属类别，继续选择最不能确定的样本。

这是因为此方法前提假设是，用最不能确定的样本训练学习模型，能使学习模型获益更多。这种方法仅仅适用于能够估计样本属于哪一类别的学习模型，并且能够提供一种机制衡量学习模型估计样本所属类别的可信度。

本文后面提出的两种主动排序学习算法的查询函数均基于样本的不确定性方法。

2.4.2.3 基于委员会的方法

基于委员会[13]（Query By Committee，QBC）的方法是多个学习模型组成一个委员会，采取投票表决。对于某个样本，委员对样本的类别进行投票，若某个预测结果得票最多，则这个票数代表这个样本可确定的程度。最后对于每个样本来说，都有一个票数，即都有一个可确定的程度值，选择最可确定的样本加入训练集。这种方法本质上是和样本的不确定性相对的，用投票的方法决定哪个样本是最确定的。它的假设是，用最确定的样本来训练学习模型可以使学习模型学到更多的知识。用同一个标志的训练样本训练这些学习模型，当预测未知样本时，若一半的学习模型分类其为正例，另一半分类为反例，则把这个样本拿出来询问专家，确定它属于哪一类别。它的预测性能的好坏，取决于

学习模型的个数，如学习模型趋于无穷，则每一个未知样本都有可能平分这个空间。但是具体组成一个委员会需要多个学习模型，目前为止没有一个客观的方法，所以 QBC 的性能依赖于主观经验。另外大量的学习模型增加了计算复杂度。

2.4.2.4 版本空间和边缘的方法

版本空间和边缘的方法[27]（Version Space and Margin Based Approach）就是说给定一个训练集和一个分类器，存在一个超平面集 H 划分这些数据。版本空间就是一个连续的超平面集组成的空间。在这个方法中，总是选择能把这个空间二等分的样本。如果，没有这样的样本，则选择那些最能够近似二等分的样本。有很多论文讨论了版本空间应用在 SVM 中[42]，在不同的情况下如何选择样本的问题。

但是版本空间仅仅适用于训练数据是线性可分的，如果不是线性可分，可以用 SVM 的核函数特征从低维映射到高维，但不能保证在获知所有样本的类标志之前，这些高维的特征空间是线性可分的。另外，众所周知，高维特征空间不仅增加了分类和计算复杂度，并且能够影响全局泛化误差。

边缘的思想是基于最大边缘分类器（Large Margin Classifier）的概念提出来的，选择距离当前分割数据空间的超平面最近的那个样本训练分类器。这是因为，距离最近的样本对分类器的分类能力影响比较大。还有些方法是以概率 c 选择当前边界的样本，c 表示选择的是最优样本的概率。但是这种方法只考虑了样本对当前边界或超平面的影响。它们仅仅适用于最大边界的分类器，如 SVM，对其他的分类器适用性就很差了。但是 SVM 是二分类器，仅仅适用于二分问题。另外，如果初始训练集选择不当，则其后的训练很难收敛。

2.5 批量主动分类学习

主动学习算法在机器学习领域研究比较多，但批量主动学习算法研究相对较少，在分类问题中研究的论文也只有 4，5 篇，而目前排序问题中则没有相关研究，因此批量主动排序算法具有创新性和研究价值。

批量主动分类学习算法在 ICML 和 NIPS 等顶级会议上都有研究，在 ICML2003 会议上，K. Brinker 把基于支持向量的批量主动算法应用到分类问题

中[56]。在 ICML2006 上,S.Hoi 用批量主动学习算法来解决图片分类的问题[57]。在 NIPS2007 会议上，Yuhong Guo 研究了有判别力批量模型的批量主动学习算法[59]。ICML2006 会议上，S. Hoi 研究了在大量文本分类问题中的批量主动算法[58]。

第三章　基于夹角差异的批量主动排序学习

3.1　引言

当前一些原始的批量主动排序算法都是基于单个主动算法的基础上，用选择单个样本的度量再来选择下一个样本以致多个样本，这样来达到批量选择的目的。这样有一个明显的问题就是选择单个样本的度量肯定不完全适合多个样本，选择的多个样本之间可能存在信息的耦合，相似性高的问题。即选择的第一个样本是最值得标注的样本，那标注后，再用次度量选择的下一个样本，未必是值得的标注的样本。这次选择的样本可能后上次存在相似性，标了上一个样本模型学习后，这次的样本模型就能判别了。

基于夹角差异的批量主动排序算法目的在于解决上述的问题，用样本之间的夹角差异度来解决批量选择的未标注样本之间的相似性问题，不会产生批量选择多个相似的样本的情况，提高批量选择的多个样本对排序模型的价值。基于夹角差异的批量主动算法的排序算法使用排序支持向量机，该算法也是排序学习领域用的很广泛的成熟算法。

3.2　排序支持向量机

基于夹角差异的批量主动排序算法用排序支持向量机算法（RSVM）作为基本的排序算法，排序模型由排序支持向量机算法学习得到。下面详细描述一下排序支持向量机算法。

假定存在一个输入空间 $X \in R^n$，其中 n 表示特征向量 X 的维数。那么对排序的输出空间可用标签 $Y = \{r_1, r_2, ..., r_q\}$ 表示，其中 q 表示排序的个数。从而可以假设肯定存在整个排序在 $r_q > r_{q-1} > ... > r_1$，其中>表示偏序关系，存在一个排序函数集合 $f \in F$，集合中的各个函数都能决定样本之间的偏序关：

$$\overline{x_i} > \overline{x_j} \Leftrightarrow f(\overline{x_i}) > f(\overline{x_j}) \tag{3.1}$$

假设我们从空间 $X \times Y$ 中给定一个排序样本集合 $S = \left\{(\overline{x_i}, y_i)\right\}_{i=1}^{l}$，那么我们

的目标就是从F集合中选择最好的函数f^*，能够在保证给定排序样本准确率的前提下，是给定的损失函数达到最小值。

Herbrich et al.提出了一直解决上述问题的描述，他把样本形成序对，从而把排序问题转化成分类问题。

首先，我们假设f 函数是一个线性函数：

$$f_{\vec{w}}(\vec{x})=\langle\vec{w},\vec{x}\rangle \tag{3.2}$$

其中$\vec{w}$表示权值矢量，$\langle\cdot,\cdot\rangle$代表作内积，把公式2代入公式1可得：

$$\vec{x}_i>\vec{x}_j \Leftrightarrow \langle\vec{w},\vec{x}_i-\vec{x}_j\rangle>0 \tag{3.3}$$

可以看出样本序对$\vec{x}_i$和$\vec{x}_j$的关系 $\vec{x}_i>\vec{x}_j$可以用新的特征向量$\vec{x}_i-\vec{x}_j$来表示。然后，我们对每一个样本序对建立新的特征向量和新的标签。假设$\vec{x}^{(1)}$和$\vec{x}^{(2)}$分别表示序对中的第一个样本和第二个样本，$y^{(1)}$和$y^{(2)}$表示他们的标签，从而我们可以得到：

$$\left(\vec{x}^{(1)}-\vec{x}^{(2)},z=\begin{cases}+1 & y^{(1)}>y^{(2)}\\ -1 & y^{(2)}>y^{(1)}\end{cases}\right) \tag{3.4}$$

因此，从给定的训练数据集S中，我们可以创建一个新的训练数据集S'，新的训练集有ℓ个已标注的向量。

$$S'=\{\vec{x}_t^{(1)}-\vec{x}_t^{(2)},z_t\}_{t=1}^{\ell} \tag{3.5}$$

然后，我们可以把新的训练集S'当作分类数据集，从而可以训练一个支持向量机模型，通过支持向量机模型可以预测每个序对向量$\vec{x}^{(1)}-\vec{x}^{(2)}$是正例$z=+1$还是负例$z=-1$。

排序支持向量机巧妙的定义了一个基于有序数据对的损失函数：给定两个任意的训练样本$(\vec{x}_i,y_i)$和$(\vec{x}_j,y_j)$，排序支持向量机在损失函数中区分了两种情况，$y_i\succ y_j$和$y_i\prec y_j$，在下列两种情况下：(i) $y_i\succ y_j$但$f(\vec{x}_i)\prec f(\vec{x}_j)$或者(ii) $y_i\prec y_j$但是$f(\vec{x}_i)\succ f(\vec{x}_j)$。因此，基于有序对，排序支持向量机为排序学习问题定义了一个合适的损失函数。

而支持向量机模型的构建就是解决下面列出的二次最优化问题。

$$\min_{w} M(\vec{w}) = \frac{1}{2}\left\|\vec{w}\right\|^2 + C\sum_{i=1}^{\ell}\xi_i$$

满足约束：$\xi_i \geq 0$，　　(3.6)

$$z_i\langle \vec{w}, \vec{x}_i^{(1)} - \vec{x}_i^{(2)} \rangle \geq 1 - \xi_i, i = 1,...,\ell$$

当$\lambda = \frac{1}{2C}$时，上述式子可以转化为以下这个式子。

$$\min_{w} \sum_{i=1}^{\ell}[1 - z_i\langle \vec{w}, \vec{x}_i^{(1)} - \vec{x}_i^{(2)} \rangle]_+ + \lambda\left\|\vec{w}\right\|^2 \qquad (3.7)$$

其中下标“+”表示正例部分。上述式子的第一项就是所谓的经验折页点损失(empirical Hinge Loss)，第二项是修正。

假定$\vec{w}^*$是支持向量机算法所得的权值矢量。从几何学上来说，矢量$\vec{w}^*$与排序支持向量机的超平面垂直相交。我们可以利用矢量$\vec{w}^*$来构建排序样本的排序函数$f_{\vec{w}^*}$。

$$f_{\vec{w}^*}(\vec{x}) = \left\langle \vec{w}^*, \vec{x} \right\rangle \qquad (3.8)$$

当把排序支持向量机应用到信息检索领域的文档排序问题中，排序支持向量机所需要的训练集中的每个样本对应于一个“查询-文档”对。样本中的每个特征是查询和文档的一个函数。比如，词频特征是指查询词在文档中的出现次数。所有查询对应的样本合并在一起训练。不同查询下的样本不用区别对待，因此不同查询下形成的序对也不用区别对待。

最后从几何的角度解释一下排序支持向量机，图（3.1）对排序支持向量机做出了几何上的解释，假设图中空心的圆点代表在训练数据中$z=+1$的有序对，实心圆点代表了$z=-1$的有序对。公式（3.6）的优化目标是找出能够最大化两类之间的边缘的分类超平面，由于两类的数据点在空间内关于坐标原点中心对称，因此超平面经过坐标原点。此超平面的法向量$\vec{w}^*$所指的方向即为所求的排序方向，如果一个有序对位于边缘内（如图3.1 中所示第j个有序对），或者被错误的划分在分类超平面错误的一方（如图3.1 中所示第i个有序对），则目标函数加上强度为ξ的惩罚，其中ξ的值为数据点到正确分类边缘的距离。

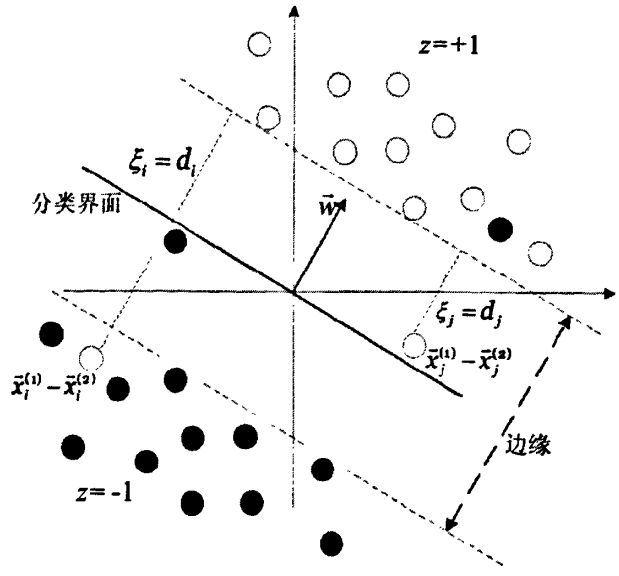

图 3.1 排序支持向量机的几何解释

3.3 基于夹角差异的批量主动排序算法

3.3.1 选择单个样本的主动学习

在特征空间线性可分的情况下，我们假设存在一个线性函数 $f(\vec{x}) = \langle \vec{w}, \vec{x} \rangle$，对每一个训练样本 $\vec{x_i}$，都能满足等式 $f(\vec{x_i}) = y_i$。其中 $\langle \cdot, \cdot \rangle$ 代表作内积，则非空集合：

$$V := \{w \in F \mid y_i \langle \vec{w}, \vec{x_i} \rangle > 0, i = 1, ..., n \text{ 并且 } \| w \| = 1\} \tag{3.9}$$

称之为版本空间（Mitchell，1982）。版本空间 V 由所有可以把训练集正确划分的线性决策器的权值矢量（归一化的）组成。版本空间 V 中权值矢量的数量称之为版本空间的规模。

通过研究表明如果每次选择的未标注样本加入训练集后，得到的新训练集的版本空间的规模能降低一半，那么这个未标注样本就是最值得标注的样本，也是主动学习需要选择的样本。同时每次选择离排序支持向量机决策面距离最近的未标注样本就能够每次近似的将版本空间的规模降低一半。

因此我们可以得到选择单个样本的主动学习的算法，就是每次选择离排序支持向量机决策面距离最近的点。这种算法我们称版本空间主动学习，或者称为基于距离的主动学习。图 3.2 是对基于距离的主动学习的几何解释：

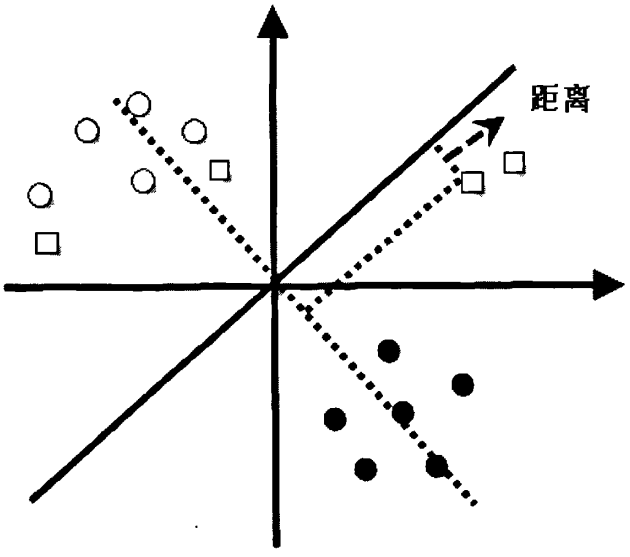

图 3.2 基于距离的主动学习

图 3.2 中空心的圆圈表示正例，实心的圆圈标注负例，方块表示未标注样本，黑色的直线表示决策面。离决策面较远的那些未标注样本，决策函数本身就能够正确划分它们，所以标注它们对决策函数的性能提升没有任何影响，所以它们是不值得标注，而如果选择它们标注则会浪费人的标注时间，提高标注代价。离决策面近的那些点，当前决策函数并不能完全正确划分它们，所以选择它们标注就可以提高决策函数的正确率，它们就是最值得标注的样本，是基于距离的主动学习需要选择的未标注样本。

3.3.2　加入差异度量批量选择样本

用基于距离的主动学习可以解决每次选择一个未标样本的主动学习，当需要一次选择多个未标注样本的时候，我们也可以对基于距离的主动学习做一个简单的扩展。即选择 h 个未标注样本，这些未标注样本离决策面的距离和是最

小的，其中 h 指每次需批量选择的未标注样本的数量。虽然，这 h 个样本每个样本都可以近似的使版本空间的规模降低一半，但整个 h 个样本对决策函数的贡献并不大，因为这 h 个样本中的样本之间肯定存在信息冗余，可能有些样本对决策函数的贡献是一样的，标注 1 个这样的样本和标注 n 个这样的样本对决策函数性能的提升也是一样的。如下图 3.3：

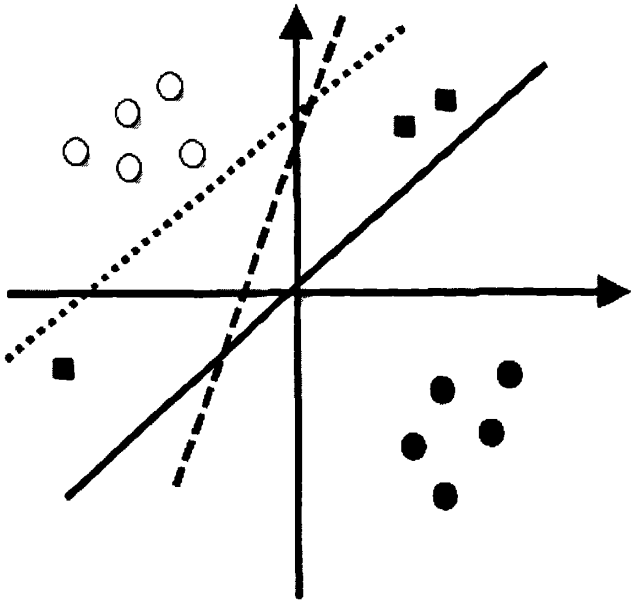

图 3.3 基于距离的批量主动学习的不足

图 3.3 中空心的圆圈表示正例，实心的圆圈标注负例，方块表示未标注样本，黑色的直线表示决策面。例如 $h=2$，依照基于距离的批量主动学习算法，则选择离决策面距离和最小的两个样本，即右上方两个方块。可以看出，这两个样本是离决策面近的样本，同时决策面不能正确划分它们，选择它们进行标注的话对决策函数的性能提升是有好处的。但同时我们也可以看出即使我们只选择这两个样本中的一个样本进行标注，对新的训练集重新训练和得到的新的决策面（图中那个长虚线），也可以对这两个样本正确划分，所以标注这两个样本中的一个还是标注两个对决策函数的贡献是一样的，因此基于距离的批量主动学习算法有其不足，还存在这浪费人工标注的情况。

解决以上的问题有一种 Tong 和 Koller 在 2000 年提出的最小化版本空间的算法[42]，但这种算法时间复杂度太高。本文将使用一种加入夹角差异度量的方

法来解决这个问题。

夹角差异是指两个样本之间的余弦值，余弦值越小，夹角越大，表明这两个样本的差异越大，这样就解决了距离决策面近的样本类似，而造成标注浪费的情况产生。夹角差异用数学公式可定义为：

$$\left|\cos(\angle(\overline{x_i},\overline{x_j}))\right| = \frac{\left|\left\langle \overline{x_i},\overline{x_j}\right\rangle\right|}{\left\|\overline{x_i}\right\|\left\|\overline{x_j}\right\|} \tag{3.10}$$

其中$\left\langle \overline{x_i},\overline{x_j}\right\rangle$表示两个矢量做点积，$\left\|\overline{x_i}\right\|$表示矢量的模。夹角的几何解释如图 3.4：

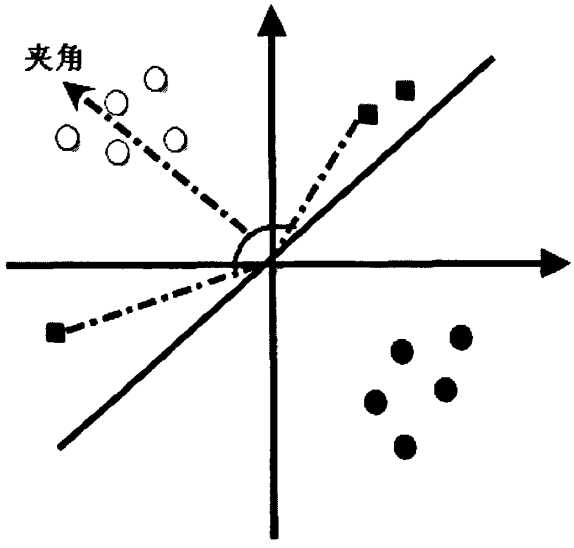

图 3.4 样本之间的夹角

夹角差异的思路就是批量选择的样本，选择与已选择的未标注样本集中夹角差异大的样本。假定U表示未标注样本集，S标注已选择的未标注样本集，那么从U选择的未标注样本$\overline{x_j}$必须满足（其中，选择第一个样本的时候，因为已选择样本集中没有样本，所以还是只使用距离度量，选择离决策面距离最近的样本作为第一个样本）

$$\max_{x_i \in S} \frac{\left|\left\langle \overline{x_i},\overline{x_j}\right\rangle\right|}{\left\|\overline{x_i}\right\|\left\|\overline{x_j}\right\|} \tag{3.11}$$

最后，为了把距离度量和夹角度量整合在一起，本文建立了一种凸函数组

合两种度量，可以通过以下方法批量选择未标注样本，假定U表示未标注样本集，S标注已选择的未标注样本集。初始化S为空集，未标注样本U中每个样本$\overline{x_j}$与决策面的距离可以算出为$g(\overline{x_j})$，则选择的未标注样本需满足

$$\lambda g(\overline{x_j})+(1-\lambda)\max_{x_i\in S}\frac{\left|\langle \overline{x_i},\overline{x_j}\rangle\right|}{\left\|\overline{x_i}\right\|\left\|\overline{x_j}\right\|} \tag{3.12}$$

然后把样本$\overline{x_j}$加入到已选择集S中，重复以上步骤，直到S集合中的样本数量达到批量选择的要求。

λ参数能够权衡距离度量和夹角度量。当λ等于 1 时，则就是单纯基于距离的批量主动算法。当λ等于零时，则是单纯基于夹角的批量主动算法。

3.3.3　时间复杂度

前人研究表明排序支持向量机算法需要$O(m^2)$的时间训练样本得到排序模型[56]，其中m表示训练序对样本的个数（Platt，1999）。因此，不算主动学习选择未标注样本的时间消耗，主动学习通过每次选择h（$1\le h\le m$）个未标注样本总共学习m个样本的时间代价是$O(\frac{m^3}{h})$。

为了选择新的值得标注的未标注样本，每次迭代选择h个未标注样本，都需要计算所有的未标注样本到排序支持向量机模型决策超平面的距离。假定初始的未标注样本数量是n，我们可以得到计算排序支持向量机模型决策超平面的距离需要的总的时间代价是$O(n\frac{m^2}{h})$，其中考虑到单个未标注样本到超平面的距离的计算是与已标注样本的数量相关的。另外，对每个未标注样本计算夹角度量需要$O(nm)$的时间代价。

把训练标注样本和选择未标注样本的时间代价加起来，则从n个未标注样本中每次选择h个样本标注总共学习m个样本的基于夹角差异的批量主动学习算法的时间复杂度是：

$$O(\underbrace{\frac{m^3}{h}}_{\text{训练}}+\underbrace{n\frac{m^2}{h}}_{\text{距离计算}}+\underbrace{nm}_{\text{夹角计算}}) \tag{3.13}$$

对比单纯基于距离的批量主动排序算法，基于夹角差异的批量主动排序算法仅仅需要额外的$O(nm)$的计算代价。

3.3.4　批量选择算法描述

基于夹角差异的批量选择未标注样本算法

给定：

N：需批量选择的未标注样本个数

λ：距离度量和夹角差异度量之间的权衡

L：训练样本集合

U：未标注样本集合

w：上一次训练得到的排序函数权值矢量

初始化：

$S=\{\varnothing\}$：已选择的未标注样本集合，初始为空

MaxCos[]={0}：未标注样本的夹角度量，初始为零

Foreach（样本 $\vec{x}_i$ in U）

Distance[i] = $\vec{x}_i \cdot \vec{w}$;

//算出未标注样本集中每个样本与决策面的距离

End

Repeat

Foreach（样本 $\vec{x}_i$ in U）

$$MaxCos[i]=\max_{y_j \in S}\frac{\left|\vec{x}_i \cdot \vec{y}_j\right|}{\left\|\vec{x}_i\right\|\left\|\vec{y}_j\right\|}$$

//计算每个未标注样本与已选择样本的夹角

//差异度量

End

Foreach（样本 $\vec{x}_i$ in U）

$f(\overline{x_i})$ = distance[i] + λ*maxCos[i];

//计算未标注样本的选择度量

End

$S = S \cup \{\overline{x_i} = \min_{\overline{x_i} \in U} f(\overline{x_i})\}$

//把选择度量最小的未标注样本加入到选择集中

If （size of S == N）

Break; //批量选择的未标注样本达到预定数量

End

End

$L = L \cup S$

//把批量选择的未标注样本标注后加入到训练集中

3.3.5　夹角差异批量主动学习算法流程

基于夹角差异的批量主动排序算法的流程是：首先给定少量已标注训练样本集 L 和大量未标注训练样本集 U，每次迭代过程中交由人工标注的样本个数 N，以及距离度量和夹角度量的权衡 λ 和算法的结束条件。使用排序感知机算法在少量已标注训练样本集 L 上建立初始排序模型，同时，使用 3.3.4 小节列出的选择算法从大量未标注样本 U 中批量选择出 N 个“最值得标注”的样本交由人工标注，加入已标注样本集 L 中，使用排序感知机算法得到新的排序模型。直到达到结束条件，一般是主动学习的次数。

基于夹角差异的批量主动排序算法描述

给定：

N：需批量选择的未标注样本个数

λ：距离度量和夹角差异度量之间的权衡

L：初始训练样本集合

U：初始未标注样本集合

Repeat:

1. 用排序支持向量机学习训练样本集 L 得到排序模型，从而得到权值矢量 w。

2. 用3.3.4小节列出的选择算法从未标注批量选择N个未标注样本集S进行标注。

3. 把选择的N个未标注样本集加入到训练集中，$L=L\cup S$并且把它们从未标注样本几中删除，$U=U-S$

4. 如果满足结束条件，则退出循环。

End

输出：批量主动排序学习所得的排序模型 $f_{\vec{w}^*}(\vec{x})=\left\langle \vec{w}^*,\vec{x}\right\rangle$

3.4 实验及分析

3.4.1 数据集

本文采用两个权威的大规模真实数据集合验证本文提出的两种主动排序学习算法的性能，分别是 OHSUMED 数据集合与 TREC .Gov 数据集合。

3.4.1.1 OHSUMED 数据集

OHSUMED 数据集合曾在国际文档检索竞赛 TREC-9 中使用。该数据集中的文档来源于美国医药信息数据库，内容是医药类杂志的标题和/或摘要。数据集包含了 348566 个文档和 106 个查询。基于这些文档集合和查询集合，OHSUMED 一共标注了 16140 个查询—文档对，每一个查询—文档对都被标注成相关，部分相关或者不相关，最终的标注结果中一共包含了 2557 个相关、2932 个部分相关以及 12498 个不相关的查询－文档对。

每一个 OHSUMED 文档，由 8 个域组成，含义如下：

- I 文章的 OHSUMED 序列号，从 1 到 348566
- U MEDLINE 标识
- S 文章来源
- M MeSH 索引词
- T 文章标题
- P 文章类型
- W 文章摘要
- A 文章作者
- 每一个 OHSUMED 查询，由如下不同域组成：
- I 文章的 OHSUMED 序列号，从 1 到 106
- B 患者信息
- W 信息需求

这些查询来源于医生在给病人看病的过程中所提交的查询字符串，每一个查询由两部分组成：病人情况的简单描述和所需信息的描述。

3.4.1.2 TREC .gov 数据集

TREC .Gov 数据集合[10]是一个网页文档集合，抓取自 2002 年.gov 域名下网站的网页。TREC .Gov 数据集合自 2002 年以来，一直作为 TREC 竞赛中的 Web Track 任务的标准数据集。TREC .Gov 数据集包含了 1,053,110 个网页和 11,164,829 个超链接。本文使用 TREC-2003 中的主题选择任务中给出的 50 个查询，基于这些网页集合和查询集合，TREC 竞赛的组织者针对每一个查询都标注了大量的网页，每一个查询—网页对都被标注成相关或者不相关。对于不同的查询，与其相关的网页数量也是不同的，最少的只有 1 个，最多的有 86 个。

3.4.2 批量主动实验流程

进行主动排序学习算法实验的主要步骤包括：数据集文档和查询通过数据预处理算法和特征提取算法，形成训练数据；使用排序学习算法建立排序模型；同时，使用查询函数从大量未标注样本中选择出那些“最值得标注”的样本交由人工标注，加入已标注样本集中，并反复迭代；在每次迭代的过程中，都使用测试数据检测排序模型，并使用评估算法得到排序结果。如图 3.5 所示。

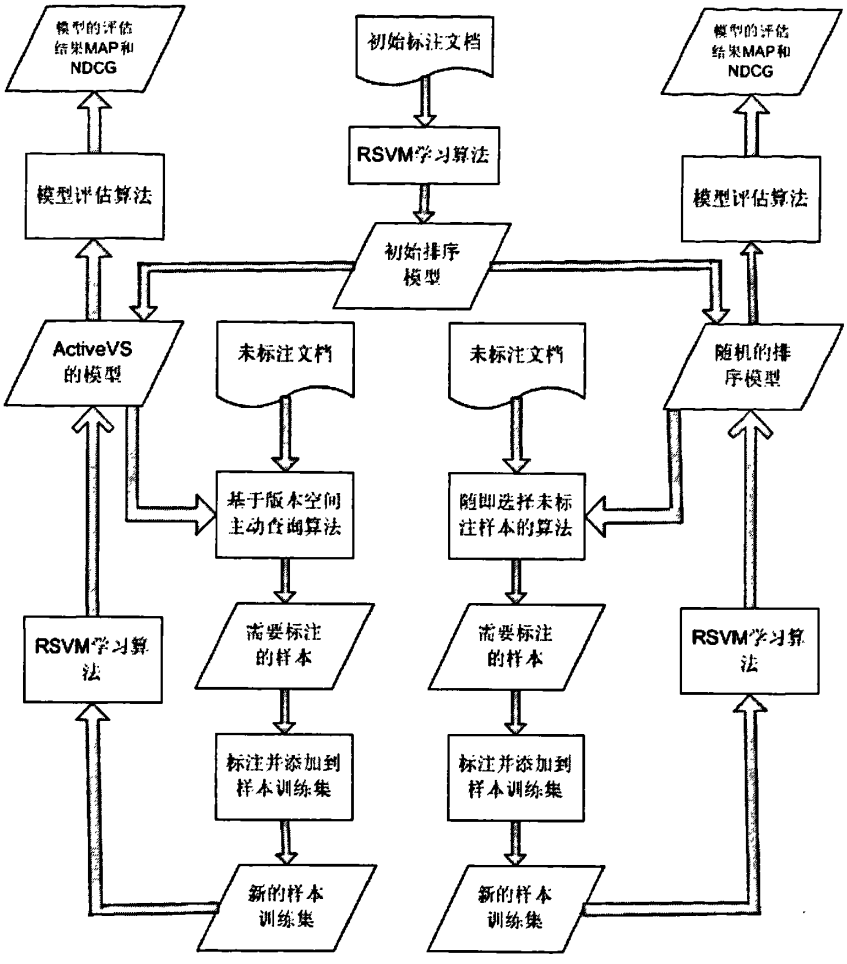

图 3.5 批量主动排序实验流程图

在进行检索前，所有文档和查询都做了相同的预处理，包括抽取词干（stemming）、过滤停用词（stop words）等。

文档数据经过预处理之后由 Lemur 检索系统建立索引。本节的实验索引了文档中的标题域（.T）和摘要域（.W），标题、摘要、标题＋摘要都分别被建立索引进行查询。查询经过同样的预处理之后，由 Lemur 检索系统检索出与查询相关的文档（使用 BM25 算法），进行进一步处理。

每一个提取出来的查询－文档对用一个特征向量来表示，基于特征向量和其对应的序列标号，使用排序感知机算法、排序支持向量机算法和本文所提出

的主动排序感知机算法、主动排序学习支持向量机算法分别建立了排序模型，它们可以用来对测试数据集合中的查询进行排序。

有监督学习的方法需要把“查询－文档对”表示成特征向量，特征向量综合利用查询和文档的一些统计信息，例如词频、倒转文档频率（inversed document frequency）、文档长度以及它们的组合作为特征，这些特征在很多文献中使用过。对于 OHSUMED 数据集合，下表列举出了在学习过程中所使用的所有的特征。其中：函数 C（w, d）计算单词 w 在文档 d 中的出现频率；C 代表整个文档集合；n 是查询中的单词个数；函数|.|表示文档的长度或者文档集合的大小；idf（.）表示文档频率的倒数。在构造特征的过程中，使用了对数函数 log 来消除大数对学习的不良影响。

表 3.1 OHSUMED 数据集实验使用特征列表

$\sum_{q_i \in q \cap d} c(q_i, d)$	$\sum_{q_i \in q \cap d} \log(c(q_i, d)+1)$
$\sum_{q_i \in q \cap d} \log \frac{c(q_i, d)}{\lvert d \rvert}$	$\sum_{q_i \in q \cap d} \log(\frac{c(q_i, d)}{\lvert d \rvert}+1)$
$\sum_{q_i \in q \cap d} \log(idf(q_i))$	$\sum_{q_i \in q \cap d} \log(\log(idf(q_i)))$
$\sum_{q_i \in q \cap d} \log(\frac{\lvert C \rvert}{c(q_i, C)}+1)$	$\sum_{q_i \in q \cap d} \log(\frac{c(q_i, d)}{\lvert d \rvert} \cdot idf(q_i)+1)$
$\sum_{q_i \in q \cap d} c(q_i, d) \log(idf(q_i))$	$\sum_{q_i \in q \cap d} \log(\frac{c(q_i, d)}{\lvert d \rvert} \cdot \frac{\lvert C \rvert}{c(q_i, C)}+1)$
BM25 score	log(*BM25 score*)
LMIR with DIR smoothing	*LMIR with JM smoothing*
LMIR with ABS smoothing	

由于 TREC .gov 数据集合的内容是网页，因此选用如下 44 个特征，包括：词频（tf）、倒转文档频率（idf）、链接信息以及一些经典排序算法结果等作为特征，如下表所示：

表 3.2 TREC .gov 数据集实验使用特征列表

1. *BM25*	2. *dl of body*
3. *dl of anchor*	4. *dl of title*
5. *dl of URL*	6. *HITS authority*
7. *HITS hub*	8. *HostRank*
9. *idf of body*	10. *idf of anchor*
11. *idf of title*	12. *idf of URL*
13. *Sitemap based feature propagation*	14. *PageRank*
15. *LMIR.ABS of anchor*	16. *BM25 of anchor*
17. *LMIR.DIR of anchor*	18. *LMIR.JM of anchor*
19. *LMIR.ABS of extracted title*	20. *BM25 of extracted title*
21. *LMIR.DIR of extracted title*	22. *LMIR.JM of extracted title*
23. *LMIR.ABS of title*	24. *BM25 of title*
25. *LMIR.DIR of title*	26. *LMIR.JM of title*
27. *Sitemap based feature propagation*	28. *tf of body*
29. *tf of anchor*	30. *tf of title*
31. *tf of URL*	32. *tfidf of body*
33. *tfidf of anchor*	34. *tfidf of title*
35. *tfidf of URL*	36. *Topical PageRank*
37. *Topical HITS authority*	38. *Topical HITS hub*
39. *Hyperlink base score propagation: weighted in-link*	40. *Hyperlink base score propagation: weighted out-link*
41. *Hyperlink base score propagation: uniform out-link*	42. *Hyperlink base feature propagation: weighted in-link*
43. *Hyperlink base feature propagation: weighted out-link*	44. *Hyperlink base feature propagation: uniform out-link*

本文所有实验均使用 5 折交叉验证方法验证排序模型的性能，具体计算过程如表 3.3 所示。

表 3.3 实验数据分块表

	训练集	调参集	测试集
Fold1	{S1, S2, S3}	S4	S5
Fold2	{S2, S3, S4}	S5	S1
Fold3	{S3, S4, S5}	S1	S2
Fold4	{S4, S5, S1}	S2	S3
Fold5	{S5, S1, S2}	S3	S4

1. 把查询随机地分为 5 等份，依照对查询的划分，将标注好的查询—文档对也被划分成为 5 份；

2. 对每一个子实验，3 份的数据被合并作为训练集合，另外一份数据作为调参集合，一份作为测试集合。基于训练集合训练出的排序模型在测试集合上进行测试，得到了本次子实验的性能；

3. 重复步骤 2 五次，每次都使用不同组合的训练集合和测试集合，最终的排序性能是五个子实验测试性能的平均值。

3.4.3 实验结果及分析

本小节将对本文提出的基于夹角差异的批量主动学习算法进行实验，主要分为 3 个方面，1，在相同标注量下本文提出的批量主动排序学习算法和非批量主动排序学习算法的性能和时间比较；2，在每轮标注相同数目的未标注样本下，本文提出的夹角差异批量算法和其他的原始的批量算法的性能改变率比较；3，在相同标注量的情况下，夹角差异批量算法和其他的原始的批量算法的性能比较。分别在两个权威的大规模真实数据集合，OHSUMED 数据集合与 TREC .Gov 数据集合上做实验进行验证。

因为实验数据一开始都是已经标注好了，所以可以在上面做主动学习的模拟实验。即假设训练集的一部分为标注样本，剩下的即为未标注样本，主动学习算法从未标注样本里选择的样本也不需要人工标注，只要用它原来的标签就可以。对比实验的把训练集的前 5 个查询设置为初始的训练样本集，其他查询都设为未标注样本集，每次从未标注样本集中批量选择 50 条未标注样本集进行标注，主动学习 9 轮，一共标注 450 条未标注样本。并使用 3.4.2 小节所述的 5

折交叉验证。

3.4.3.1　与单个主动的比较

首先进行基于夹角差异的批量主动与一次只选择一个样本标注的主动排序的比较实验，两种算法在两个数据集 OHSUMED 和 TREC.Gov 上实验，图 3.6 是 OHSUMED 数据集下相同标注量下单主动和批量主动的性能比较，图 3.7 是 TREC.Gov 数据集下相同标注量下单主动和批量主动的性能比较，表 3.4 是 OHSUMED 数据集下相同标注量下单主动和批量主动的时间比较，表 3.5 是 TREC.Gov 数据集下相同标注量下单主动和批量主动的时间比较：

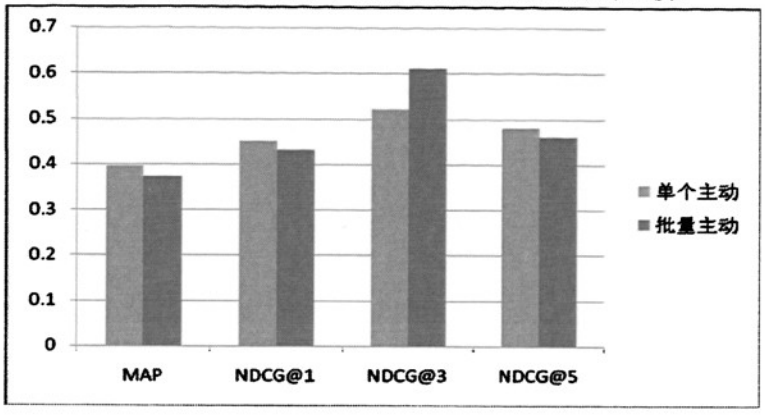

图 3.6 OHSUMED 数据集下相同标注量下单主动和批量主动的性能比较

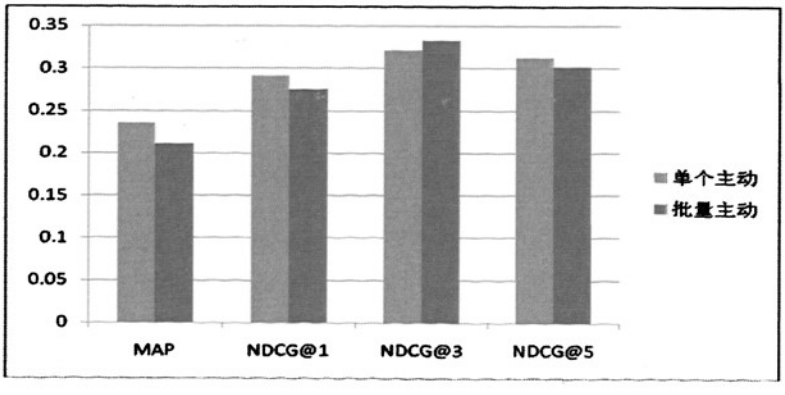

图 3.7 TREC.Gov 数据集下相同标注量下单主动和批量主动的性能比较

表 3.4 和 3.5 比较执行两种算法所需的时间代价，因为两种算法的标注量一样，标注时间一样，时间只计算了迭代训练排序模型和选择样本的时间。

表 3.4 OHSUMED 数据集下相同标注量下单主动和批量主动的时间比较

主动算法	时间
单个主动	约 16 个小时
批量主动	38 分钟

表 3.5 TREC.Gov 数据集下相同标注量下单主动和批量主动的时间比较

主动算法	时间
单个主动	约 18 个小时
批量主动	45 分钟

从上面列出的图和表格数据可以看出在相同标注量的情况下，批量主动和单一主动算法的性能相差不大，但所消耗的时间则相差很大，因此批量主动排序可以在保证主动排序性能的情况下减少主动排序得到排序模型的时间，降低主动排序的学习代价。

3.4.3.2　与原始的批量主动算法的改变率比较

基于夹角差异的批量主动排序算法与单纯的原始的批量排序算法的比较实验，用来比较基于夹角差异的批量主动算法和原始的批量算法的性能差异。原始批量主动排序算法采用单纯基于距离的批量算法，这也是现阶段大部分人使用的批量主动排序算法。两种批量算法初始训练集都为前 5 个查询，然后每次选择 50 个未标注样本进行标注，总共学习 9 轮，比较两个算法在把选择的 50 个未标注样本加到训练集后对排序模型的准确率提高程度，两种算法同样是在 OHSUMED 和 TREC.Gov 两个数据集上实验，评价指标是平均查准率 MAP。图 3.8 是 OHSUMED 数据集上夹角批量和距离批量每次学习的 MAP 比较，图 3.9 是 TREC.Gov 数据集上夹角批量和距离批量每次学习的 MAP 比较。其中图中起始点的数据是在初始的训练集上学习得到的排序模型的性能数据。

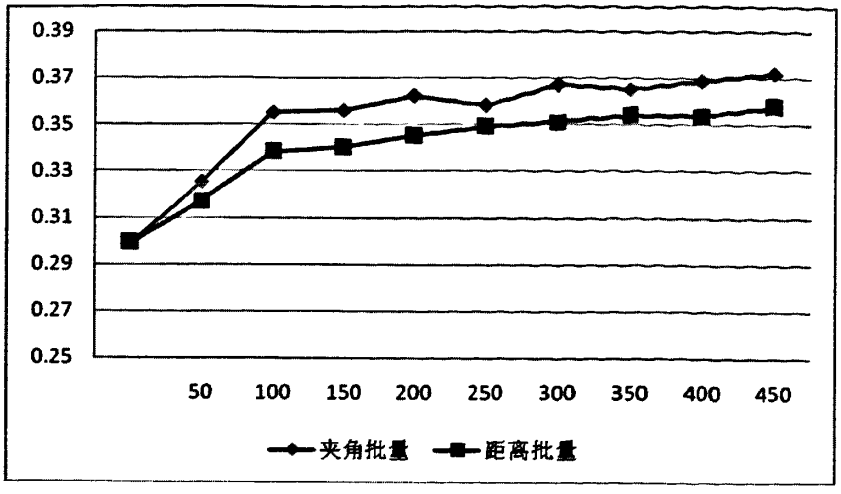

图 3.8 OHSUMED 数据集上夹角批量和距离批量每次学习的 MAP 比较

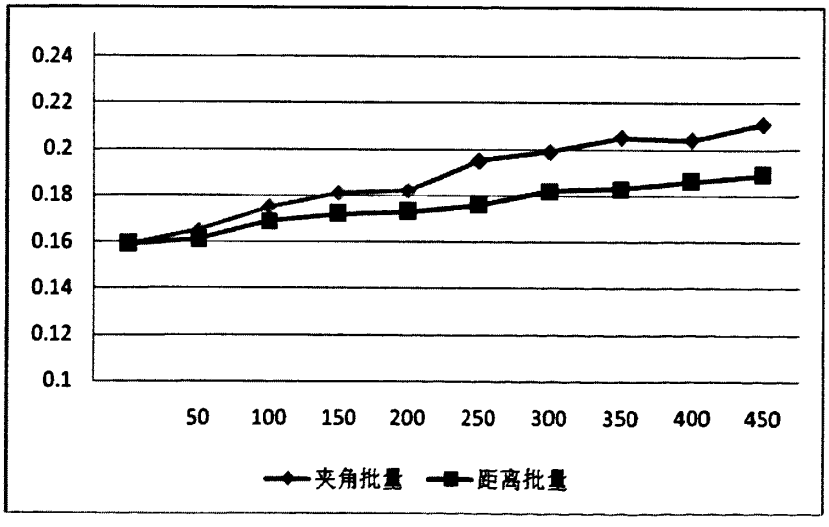

图 3.9 TREC.Gov 数据集上夹角批量和距离批量每次学习的 MAP 比较

从图 3.8 和图 3.9 可以看出，本文提出的基于夹角差异的批量主动排序学习算法选择的样本比原始的批量主动排序算法选择的样本，加入训练集重新训练后得到的排序模型性能更高，它们更能够提高排序模型的排序性能。因此在达到相同的排序性能的时候，与原始的批量主动排序学习算法相比，基于夹角差异的批量主动排序学习算法只需要更少的标注未标注样本，减少批量主动排序学习的标注量，降低批量主动排序学习的标注代价。

3.4.3.3　与原始批量主动算法的相同标注量性能比较

进行基于夹角差异的批量主动学习算法和原始的批量主动排序算法在相同的标注量下的最终性能比较实验，原始的批量主动学习算法同样使用单纯基于距离的批量主动排序算法。两种批量算法初始训练集都为前 5 个查询，然后每次选择 50 个未标注样本进行标注，总共学习 9 轮，比较 9 轮学习之后，批量主动排序学习最终得到的排序模型的排序性能，两种算法同样是在 OHSUMED 和 TREC.Gov 两个数据集上实验，评价指标是 MAP 和 NDCG。图 3.10 是 OHSUMED 数据集上夹角批量和距离批量的最终性能比较，图 3.11 是 TREC.Gov 数据集上夹角批量和距离批量的最终性能比较。

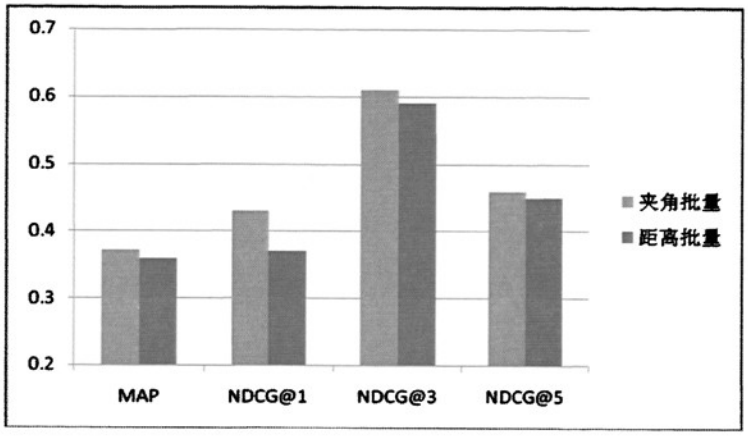

图 3.10 OHSUMED 数据集上夹角批量和距离批量的最终性能比较

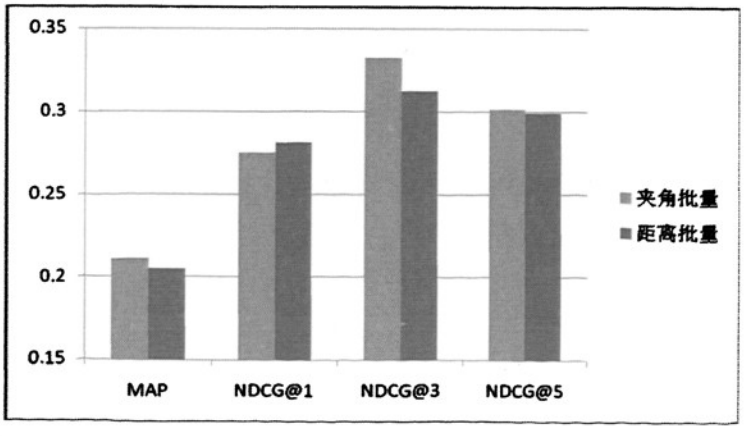

图 3.11 TREC.Gov 数据集上夹角批量和距离批量的最终性能比较

从 OHSUMED 数据集和 TREC.Gov 数据集的实验结果图 3.10 和图 3.11 可以看出，在相同标注量的情况下，不管是从评价指标 MAP，还是评价指标 NDCG，基于夹角差异的批量主动排序算法比原始的批量主动排序算法能够学习得到更好的排序模型。因此，基于夹角差异的批量主动排序算法在批量主动排序算法中有其优越性，提升了批量主动排序的性能和效率。

第四章　基于损失函数的批量主动排序学习

4.1　引言

基于夹角差异的批量主动学习算法是一种启发式的批量算法，先选择一个最好的，然后根据以前已经选择的样本依次选择以后的样本，从而达到批量选择的目的。这样做可以实现批量主动学习，但是选择的第一个未标注样本对后面的影响很大，启发式选择也有可能造成错误累积等一些问题，为了解决这些问题，因此思考能不能一下子批量选择所有未标注的样本，而不是通过启发式的，一个接一个的选择。

批量主动学习的目的就是使选择的样本标注后加入训练集能训练出一个好的决策器，提高决策函数的性能。监督学习的方法都是从使训练集样本的准确概率最大化的角度进行优化决策函数。如果存在未标注样本，半监督学习方法则是从同时最大化标注样本的准确概率和最小化未标注样本的不确定性角度优化决策函数。因此，本章打算直接从主动学习的损失函数入手，用损失函数来控制批量主动选择样本的度量。

4.2　基于损失函数的批量主动排序学习算法

4.2.1　批量主动排序的损失函数

批量主动学习的目的就是使选择的样本标注后加入训练集能训练出一个好的决策器，提高决策函数的性能，同时降低未标注样本集的不确定性。为了使批量选择的样本价值最大，批量选择的样本标注后加入训练集，能够使重新训练的排序模型的准确率最高，同时，能够使未标注样本的信息量价值降低的最大，即不确定性，也就是熵，这样批量选择的样本，就是最值得标注的样本。我们可以最大化排序模型的准确率的对数和最小化未标注样本的熵，即下列公式：

$$\sum_{i\in L}\log P(y_i\mid x_i,w)+\alpha\sum_{j\in U}\sum_{y=\pm 1}p(y\mid x_j,w)\log P(y\mid x_j,w) \tag{4.1}$$

其中α是权衡参数，为了调整标本样本和未标注样本的影响。w是决策函数的权值矢量，L是指已标注样本集合，U指未标注样本集合。

受半监督学习思想的启发，可以得出本文新的批量排序学习的思想。我们假设批量选择m个未标注样本，这个选择的未标注样本集合记为S，S在每次迭代中都要从未标注样本集U中选择，选择的依据就是最大化公式4.1。因此，本文定义了在第$t+1$迭代时批量选择未标注样本集合S的选择函数：

$$f(S)=\sum_{i\in L^t\cup S}\log P(y_i\mid x_i,w^{t+1})-\alpha\sum_{j\in U^t\setminus S}H(y\mid x_j,w^{t+1}) \tag{4.2}$$

其中w^{t+1}表示用新的训练集样本$L^{t+1}=L^t\cup S$重新学习后得到新的决策函数的权值矢量，$H(y\mid x_j,w^{t+1})$表示条件分布$p(y\mid x_j,w^{t+1})$的熵，即：

$$H(y\mid x_j,w^{t+1})=-\sum_{y=\pm 1}p(y\mid x_j,w^{t+1})\log P(y\mid x_j,w^{t+1}) \tag{4.3}$$

因此批量主动排序算法的策略就是选择使公式4.2达到最大值的未标注样本集合S。

但是在实际的求解中，因为$f(S)$直接决定着未标注样本的选择，但是在选择未样本集合的时候，未标注样本集合S的标签是未知的。这个问题的一个典型的解决方案就是用当前决策函数的权值矢量计算$f(S)$的期望，即：

$$\mathrm{E}[f(S)]=\sum_{y_S}\mathrm{P}(y_S\mid x_S,w^t)f(S) \tag{4.4}$$

然而，用$P(y_S\mid x_s,w^t)$表示权重，这样可能会加重已经存在当前的权重矢量w^t中的错误，因为w^t是在一个比较小的标注样本L^t学习得到的。本文使用一个乐观的策略，选择$f(S)$在未标注样本集合S所有可能的标签情况下所能得到的最大值作为$f(S)$的函数值，乐观的函数公式可写为：

$$f(S)=\max_{y_S}\sum_{i\in L^t\cup S}\log P(y_i\mid x_i,w^{t+1})-\alpha\sum_{j\in U^t\setminus S}H(y\mid x_j,w^{t+1}) \tag{4.5}$$

因此，批量主动排序的问题可以转化为怎么选择未标注样本集合S使公式4.5定义的乐观$f(S)$函数达到最大值。虽然这个问题可以通过穷举搜索未标注样本集合U中所有S集合大小的样本子集来解决，但实际情况中却不能这样做，因为搜索空间很大，是指数复杂度的。也不能用启发式搜索的办法解决这个问题，因为不能定义出一个有效的运算集能在搜索空间中使从一个位置传递到另一个位置，同时保证优化函数的增长。

4.2.1.1 熵

熵的概念最先在1864年由鲁道夫·克劳修斯提出，并应用在热力学中。后来在1948年由克劳德·艾尔伍德·香农第一次引入到信息论中来。时间和空间唯独不同的是，它总是向一个方向流动，从过去流向未来，这种不可逆的次序的边界上，时间的弹性软体里包裹着神秘的因果律。科学家已经发明了测量无序的量，它称作熵，熵也是混沌度，是内部无序结构的总量，可以理解成国。未知的信息越多，熵越大，也就是国越大。

熵在消息理论的定义如下：如果有一个系统 S 内存在多个事件 $S=\{E_1,...,E_n\}$,每个事件的概率分布 $P=\{p_1,...,p_n\}$，则每个事情本身的信息量为：

$$I_e=-\log p_i$$

如英语有26个字母，假如每个字母在文章中出现次数平均的话，每个字母的信息量为：

$$I_e=-\log\frac{1}{26}=4.7$$

而汉字常用的有2500个，假如每个汉字在文章中出现次数平均的话，每个汉字的信息量为

$$I_e=-\log\frac{1}{2500}=11.3$$

整个系统的平均信息量为：

$$H_s=\sum_{i=1}^{n}p_iI_e=-\sum_{i=1}^{n}p_i\log p_i$$

这个平均消息量就是信息熵。因为和热力学中描述热力学熵的玻尔兹曼公式形式一样，所以也称为“熵”。

如果两个系统具有同样大的消息量，如一篇用不同文字写的同一文章，由于是所有元素消息量的加和，那么中文文章应用的汉字就比英文文章使用的字母要少。所以汉字印刷的文章要比其他应用总体数量少的字母印刷的文章要短。即使一个汉字占用两个字母的空间，汉字印刷的文章也要比英文字母印刷的用纸少。

实际上每个字母和每个汉字在文章中出现的次数并不平均，因此实际数值并不如同上述，但上述计算是一个总体概念。使用书写单元越多的文字，每个

单元所包含的信息量越大。

4.2.2 损失函数最优化求解

本文打算把这个批量主动排序的问题转化成一个纯粹的数学优化问题。批量主动排序的问题是，在t次迭代后给定标注样本集合L^t，未标注样本集合U^t，我们的任务是在$t+1$次迭代中能够从集合U^t中选择大小为m的子集合S能够使公式4.5所列的函数达到最大值。为了解决这问题，我们首先定义一个$|U^t|\times 2$大小的选择矩阵μ，选择矩阵μ的行数为未标注样本集合U^t的样本数量，列数为2。选择矩阵μ中元素的值不是0则是1，0表示未选择，1表示选择。选择矩阵μ中每行的行矢量μ_j表示未标注样本集合U^t中第j个样本是否被选择，同时表示选择第j个样本是正例还是负例。因此，$t+1$次迭代的选择函数可以描述为以下最优化公式：

$$\begin{aligned}
&\max_{\mu} \sum_{i\in L^t} \log P(y_i \mid x_i, w^{t+1}) + \beta \sum_{j\in U^t} v_j^{t+1}\mu_j^{\mathrm{T}} - \alpha \sum_{j\in U^t} (1-\mu_j e) H(y \mid x_j, w^{t+1}) \\
&\text{约束于：}\ \mu \in \{0,1\}^{|U^t|\times 2} \\
&\qquad \mu \bullet E = m \\
&\qquad \mu_j e \le 1, \forall j \\
&\qquad I^{\mathrm{T}}\mu \le (\frac{1}{2}+\varepsilon) m e^{\mathrm{T}}
\end{aligned} \tag{4.6}$$

其中，v_j^{t+1}是一个行矢量表示$[\log(P(y=1\mid x_j, w^{t+1}), \log(P(y=-1\mid x_j, w^{t+1})]$，e表示2行的列矢量，矢量中元素值都是1，$I$表示$|U^t|$行大小的列矢量，矢量中元素值都为1，$E$表示$|U^t|\times 2$大小的矩阵，矩阵中的元素值都是1。$\bullet$表示矩阵作内积，$\varepsilon$是用户输入的参数，用以控制选择的正例和负例的个数平衡，β是一个自调节参数，用来适应我们选择的未标注样本的标签的准确程度。从公式4.6可以注意到，选择矩阵μ不仅表示从未标注样本集合U^t选择样本，同时也指定了选择的未标注样本的标签。因此，要解决这个最优化问题，就要在$t+1$次迭代选择最优的选择矩阵μ。

公式4.6所列的最优化问题是一个整数规划问题，公式4.6等同于公式4.5，它们的解是一样的。整数规划问题是一个NP复杂问题，因此，要解决这个问题，首先要放松约束把这个问题变成一个连续的最优化问题，即把整数约束$\mu \in \{0,1\}^{|U^t|\times 2}$替换成连续约束$0 \le \mu \le 1$，得到放松的公式描述：

$$\max_{\mu} \sum_{i \in L^t} \log P(y_i \mid x_i, w^{t+1}) + \beta \sum_{j \in U^t} v_j^{t+1} \mu_j^{\mathrm{T}} - \alpha \sum_{j \in U^t} (1 - \mu_j e) H(y \mid x_j, w^{t+1})$$

约束于：$0 \le \mu \le 1$

$$\mu \bullet E = m$$

$$\mu_j e \le 1, \forall j \tag{4.7}$$

$$I^{\mathrm{T}} \mu \le (\frac{1}{2} + \varepsilon) m e^{\mathrm{T}}$$

对于公式 4.7，可以用标注的连续函数优化技术来解决这个问题，找到最大值。可以把公式 4.7 当作是一个关于选择矩阵 μ 的函数

$$f(\mu) = \sum_{i \in L^t} \log P(y_i \mid x_i, w^{t+1}) + \beta \sum_{j \in U^t} v_j^{t+1} \mu_j^{\mathrm{T}} - \alpha \sum_{j \in U^t} (1 - \mu_j e) H(y \mid x_j, w^{t+1}) \tag{4.8}$$

可以看出公式 4.8 是非凹函数，因此，不能很方便使用凸函数优化技术找到一个全局最优解。但是，局部优化技术如拟牛顿法可以很快的得到局部最优解 μ^*，拟牛顿法具体算法可以参考 4.4 节。拟牛顿法求解最大值，需要先求出函数 $f(\mu)$ 的梯度和二次偏导矩阵。函数 $f(\mu)$ 的局部梯度可以近似表示为：

$$\nabla f(\mu_{j(k)}) = \beta v_j^{t+1} + \alpha [H(y \mid x_j, w^{t+1}), H(y \mid x_j, w^{t+1})] \tag{4.9}$$

因此局部梯度 $\nabla f(\mu_{(k)})$ 可以由独立的各个 $\nabla f(\mu_{j(k)})$ 构建组成，本文使用拟牛顿法中的 BFGS 公式里计算二次偏导矩阵，二次偏导矩阵初始为所有元素都为 1，然后每次迭代用以下公式更新：

$$H_{k+1} = H_k - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

其中 $y_k = \nabla f_{k+1} - \nabla f_k$，$s_k = \mu_{k+1} - \mu_k$。

4.2.3　批量选择算法描述

基于损失的批量选择未标注样本算法

给定：

N：需批量选择的未标注样本个数

L：训练样本集合

U：未标注样本集合

ε：拟牛顿算法所需的控制误差

初始化：

初始化选择矩阵 μ_0

初始化 Hessian 矩阵 H_0 为单位矩阵

初始化迭代次数变量 $k=0$

计算排序支持向量机的概率输出模型参数

用公式 $\nabla f(\mu_{j(k)}) = \beta v_j^{t+1} + \alpha[H(y|x_j, w^{t+1}), H(y|x_j, w^{t+1})]$

计算 $\nabla f(\mu_{(0)})$

Repeat

令 $p_k = -H_k \nabla f(\mu_k)$

由精确一维搜索确定步长 α_k，

$f(\mu_k + \alpha_k p_k) = \min_{\alpha \geq 0} f(\mu_k + \alpha_k p_k)$

计算 $\mu_{k+1} = \mu_k + \alpha_k p_k$

计算 $\nabla f(\mu_{k+1})$，

If $\|\nabla f(\mu_{k+1})\| \leq \varepsilon$

$\mu^* = \mu_{k+1}$；

Break；

Else

$y_k = \nabla f_{k+1} - \nabla f_k$，$s_k = \mu_{k+1} - \mu_k$

$$H_{k+1} = H_k - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

End

End

对 μ^* 进行取整化处理

把 μ^* 选择矩阵选择的样本加入到训练集中

4.2.4　损失函数批量主动学习算法流程

基于损失函数的批量主动排序算法的流程是：首先给定少量已标注训练样本集 L 和大量未标注训练样本集 U，每次迭代过程中交由人工标注的样本个数 N，以及拟牛顿算法所需的控制误差 ε 和算法的结束条件。使用排序感知机算法在少量已标注训练样本集 L 上建立初始排序模型，同时，使用 4.2.3 小节列出的选择函数从大量未标注样本 U 中批量选择出 N 个“最值得标注”的样本交由人工标注，加入已标注样本集 L 中，使用排序感知机算法得到新的排序模型。直到达到结束条件，一般是主动学习的次数。

基于损失函数的批量主动排序算法描述

给定：

N：需批量选择的未标注样本个数

ε：拟牛顿算法所需的控制误差

L：初始训练样本集合

U：初始未标注样本集合

Repeat:

1. 用排序支持向量机学习训练样本集 L 得到排序模型，从而得到权值矢量 W。

2. 用4.2.3 小节列出的选择函数从未标注批量选择N个未标注样本集 S 进行标注。

3. 把选择的N个未标注样本集加入到训练集中，$L=L\cup S$ 并且把它们从未标注样本几中删除，$U=U-S$

4. 如果满足结束条件，则退出循环。

End

输出：批量主动排序学习所得的排序模型 $f_{\vec{w}^*}(\vec{x}) = \left\langle \vec{w}^*, \vec{x} \right\rangle$

4.3　排序支持向量机的概率输出模型

损失函数需要计算样本的后验概率 P（正负例类别|样本特征集），但是排序支持向量机算法没有输出后验概率的功能，排序支持向量机算法的输出是没有标准的数，而不是概率。Wahba 曾提出一种由排序算法输出概率的方法，他使用 logistic link 函数输出概率：

$$P(y \mid \vec{x}) = \frac{1}{1+\exp(-yf(\vec{x}))}$$

其中 y 表示正负例类别，$\vec{x}$ 表示样本特征向量，$f(\vec{x})$ 表示排序函数的输出值。此算法通过最大化标注样本的概率训练样本，比如最小化标注样本的错误的对数。

$$\min_{w} \sum_{i \in L} \log(1+\exp(-y_i w^T x_i)) + \frac{\lambda}{2} w^T w$$

因此需要改进排序支持向量机算法，从而能够得到样本的后验概率。

Hastie 和 Tibshirani 在 1996 曾经提出过用高斯来拟合条件类密度 $p(f \mid y=1)$ 和 $p(f \mid y=-1)$，从而求得排序支持向量机输出的概率，但两个高斯估计的只是一个方差。因此后验概率 $p(y=1 \mid f)$ 是一个 sigmoid 函数，函数的坡度由方差所决定。Hastie 和 Tibshirani 调整 sigmoid 的偏差，即当 $f=0$ 时 $p(y=1 \mid f)=0.5$。这个 sigmoid 函数是单调的，但是从方差导出的单一的参数并不能精确的模拟正确的后验概率。

另外也可以更灵活的用高斯来拟合 $p(f \mid y=\pm 1)$。每个高斯的均值和方差可以由数据集来决定。贝叶斯规则可以用来计算后验概率:

$$p(y=1 \mid f) = \frac{p(f \mid y=1)p(y=1)}{\sum_{i=1,-1} p(f \mid y=i)p(y=i)}$$

其中 $p(y=i)$ 是先验概率，它可以从训练集计算获得，后验概率是一个关于 f 的解析式：

$$p(y=1|f)=\frac{1}{1+\exp(af^2+bf+c)}$$

这样模拟 SVM 的输出有两个确定。第一，从两个高斯导出的后验概率的上述公式违反了单调性原则，上述公式是非单调的。第二，在很多数据集上的实验表明，把条件类密度用高斯来拟合的假设是不成立的。

我们可以用参数模型来直接拟合后验概率 $p(y=1|f)$,而不是估计类条件密度 $p(f|y)$。模型的参数需要调节以适应最好的概率输出。参数模型的形式可以从经验数据中观察得到，因此可以用带参数的 sigmoid 形式：

$$p(y=1|f)=\frac{1}{1+\exp(Af+B)}$$

参数A和B可以用对训练集(f_i,y_i)的最大似然估计的方法进行拟合。首先，定义一个新的训练集(f_i,t_i)，其中t_i就是本文所要求的概率，定义如下：

$$t_i=\frac{y_i+1}{2}$$

参数 A 和 B 可以通过最小化训练集概率的对数的负数而得到,即是一个交叉熵错误函数：

$$\min-\sum_i t_i\log(p_i)+(1-t_i)\log(1-p_i)$$

其中p_i表示

$$p_i=\frac{1}{1+\exp(Af_i+B)}$$

上述公式的最小化是一个两个参数的最小化问题。因此，这个问题可以通过很多最优化算法求解。出于鲁棒性的考虑，本文使用了文献[54]介绍的模型依赖的最优化算法。

4.4　拟牛顿法

4.4.1　拟牛顿法介绍

在最优化问题中,拟牛顿法是著名的寻找函数最大值和最小值的一种方法。

拟牛顿法是基于牛顿法的思路，寻找梯度为零的固定点。牛顿法先假设任何函数都可以用二次方程式很好的逼近，然后用一阶导数和二阶导数（梯度和 Hessian 矩阵）来寻找固定点。牛顿法的一个很困难的问题就是二次偏导矩阵（Hessian 矩阵）计算相当复杂困难，这也影响了牛顿法的应用。一个自然的想法是采用梯度的差商近似 Hessian 矩阵来克服这个问题。

在拟牛顿法算法中，二次偏导矩阵 Hessian 矩阵的计算代价将变的相当底。二次偏导矩阵 Hessian 矩阵可以仅用在每次迭代中得到的梯度信息来近似。第一个拟牛顿法由在氩国家实验室工作的物理学家 W.C. Davidon 所提出，他在 1959 年开发了第一个拟牛顿算法：DFP 更新公式。DFP 更新公式随后由 Feltcher 和 Powell 在 1963 推广而流行，但是 DFP 更新公式现在已很少使用。现在最普遍的拟牛顿算法是 SR1 公式和 BFGS 方法。

在牛顿法中，我们可以用二次导数近似的找到函数 $f(x)$ 的最小值。函数 $f(x)$ 可以用 Taylor 级数迭代逼近：

$$f(x_k+\Delta x)\approx f(x_k)+\nabla f(x_k)^T\Delta x+\frac{1}{2}\Delta x^T B\Delta x$$

其中，∇f 表示梯度，B 表示二次偏导矩阵。这个约等式关于 Δx 的梯度是：

$$\nabla f(x_k+\Delta x)\approx \nabla f(x_k)+B\Delta x$$

设梯度为零，即 $f(x_{k+1})$ 得最小值，从而得到等式：

$$\Delta x=-B^{-1}\nabla f(x_k)$$

通过以上公式就可以找到 x_{k+1}，从而找到函数 $f(x)$ 的最小值，以上公式的难点在函数 $f(x)$ 二次偏导矩阵计算困难，还有求其的逆矩阵则更加复杂。而通过拟牛顿法不用计算函数 $f(x)$ 二次偏导矩阵，而是通过一些公式就可以更新函数 $f(x)$ 二次偏导矩阵。

4.4.2　拟牛顿法基本思想及更新公式

梯度下降法和牛顿法的迭代公式可以统一表示为：

$$x_{k+1}=x_{k+1}-\alpha_k H_k g_k$$

其中 α_k 为步长，$g_k=\nabla f(x_k)$，H_k 为 n 阶对称矩阵。

在上述公式中，若令 $H_k=I$，则是梯度下降法；若令 $H_k=G_k^{-1}$，就是牛顿法。前者具有较好的整体收敛性，但收敛速度太慢；后者虽收敛很快，但整体

收敛性差，且需要计算二阶导数，计算量大。因此，如果能做到H_k的选取既能逐步逼近G_k^{-1}，又不需要计算二阶导数，那么由上述公式确定的算法就有可能比梯度下降法快，又比牛顿法计算简单，且整体收敛性好。为了使H_k确实能有上述特点，必须对H_k附加一些条件。(1) H_k是对称正定矩阵，(2) H_{k+1}可以由H_k经简单修正而得到，(3) H_k满足所谓的拟牛顿方程。

$$H_{k+1}y_k = s_k$$

其中$y_k = g_{k+1} - g_k$，$s_k = x_{k+1} - x_k$，这样H_{k+1}就可以较好的逼近G_{k+1}^{-1}。

在满足拟牛顿法三个条件的基础上，产生了很多拟牛顿算法，各个拟牛顿算法的更新公式如下：

表 4.1 拟牛顿法更新公式

方法	$B_{k+1}=$	$H_{k+1}=B_{k+1}^{-1}=$
DFP	$\left(1-\frac{y_k\Delta x_k^T}{y_k^T\Delta x_k}\right)B_k\left(1-\frac{\Delta x_k y_k^T}{y_k^T\Delta x_k}\right)+\frac{y_k y_k^T}{y_k^T\Delta x_k}$	$H_k+\frac{\Delta x_k\Delta x_k^T}{y_k^T\Delta x_k}-\frac{H_k y_k y_k^T H_k^T}{y_k^T H_k y_k}$
BFGS	$B_k+\frac{y_k y_k^T}{y_k^T\Delta x_k}-\frac{B_k\Delta x_k(B_k\Delta x_k)^T}{\Delta x_k^T B_k\Delta x_k}$	$\left(1-\frac{y_k\Delta x_k^T}{y_k^T\Delta x_k}\right)^T H_k\left(1-\frac{y_k\Delta x_k^T}{y_k^T\Delta x_k}\right)+\frac{\Delta x_k y_k^T}{y_k^T\Delta x_k}$
Broyden	$B_k+\frac{y_k-B_k\Delta x_k}{\Delta x_k^T\Delta x_k}\Delta x_k^T$	$H_k+\frac{(\Delta x_k-H_k y_k)y_k^T H_k}{y_k^T H_k\Delta x_k}$
SR1	$B_k+\frac{(y_k-B_k\Delta x_k)(y_k-B_k\Delta x_k)^T}{(y_k-B_k\Delta x_k)^T\Delta x_k}$	$H_k+\frac{(\Delta x_k-H_k y_k)(\Delta x_k-H_k y_k)^T}{(\Delta x_k-H_k y_k)^T y_k}$

其中$y_k = g_{k+1} - g_k$。

4.4.3　拟牛顿法算法描述

给定控制误差ε

1. 给定初始点x_0，初始矩阵H_0（通常取单位阵），计算梯度g_0，令$k=0$。
2. 令$p_k = -H_k g_k$

3. 由精确一维搜索确定步长 α_k，

$$f(x_k+\alpha_k p_k)=\min_{\alpha\geq 0} f(x_k+\alpha_k p_k)$$

4. 令 $x_{k+1}=x_k+\alpha_k p_k$

5. 若 $\|g_{k+1}\|\leq\varepsilon$，则 $x^*=x_{k+1}$，停止算法。否则

$s_k=x_{k+1}-x_k$，$y_k=g_{k+1}-g_k$。

6. 由拟牛顿法的更新公式得 H_{k+1}，令 $k=k+1$，转到第 2 步。

在实际的计算中，由于舍入误差的存在以及一维搜索的不精确，DFP 算法的效率会受到很大影响，但 BFGS 算法所受影响要小得多。特别是采用非精确一维搜索时，DFP 算法效率很低，然而 BFGS 算法却仍然十分有效。目前 BFGS 算法被公认为最好的拟牛顿算法。

4.5 实验及分析

4.5.1 实验设置

基于损失函数批量主动排序算法的实验设置与基于夹角差异的批量主动排序实验设置一样，数据集使用 OHSUMED 和 TREC.Gov 数据集，批量主动实验流程设置也是先从小部分标注样本训练得到一个初始的排序模型，然后根据批量选择算法从未标注样本集中批量选择样本加入到训练集中重新训练得到下一个排序模型，依次类推，直到学习得到好的排序模型。同时实验程序跑出对比算法的实验数据，方便进行算法比较分析。

4.5.2 实验结果及分析

本小节将对本文提出的基于损失函数的批量主动学习算法进行实验，主要分为 3 个方面，1，在相同标注量下本文提出的批量主动排序学习算法和非批量主动排序学习算法的性能和时间比较；2，在每轮标注相同数目的未标注样本下，本文提出的损失函数批量算法和本文前章提出的基于夹角差异的批量算法的性能改变率比较；3，在相同标注量的情况下，损失函数批量算法和基于夹角差异的批量算法的性能比较。分别在两个权威的大规模真实数据集合，OHSUMED

数据集合与 TREC .Gov 数据集合上做实验进行验证。

因为实验数据一开始都是已经标注好了，所以可以在上面做主动学习的模拟实验。即假设训练集的一部分为标注样本，剩下的即为未标注样本，主动学习算法从未标注样本里选择的样本也不需要人工标注，只要用它原来的标签就可以。对比实验的把训练集的前 5 个查询设置为初始的训练样本集，其他查询都设为未标注样本集，每次从未标注样本集中批量选择 50 条未标注样本集进行标注，主动学习 9 轮，一共标注 450 条未标注样本。并使用 3.4.2 小节所述的 5 折交叉验证。

4.5.2.1　与单个主动的比较

首先进行基于损失函数的批量主动与一次只选择一个样本标注的主动排序的比较实验，两种算法在两个数据集 OHSUMED 和 TREC.Gov 上实验，图 4.1 是 OHSUMED 数据集下相同标注量下单主动和批量主动的性能比较，图 4.2 是 TREC.Gov 数据集下相同标注量下单主动和批量主动的性能比较，表 4.1 是 OHSUMED 数据集下相同标注量下单主动和批量主动的时间比较，表 4.2 是 TREC.Gov 数据集下相同标注量下单主动和批量主动的时间比较：

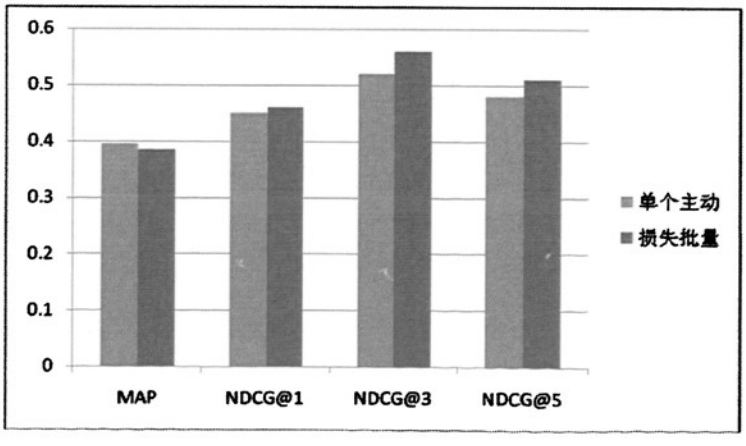

图 4.1 OHSUMED 数据集下相同标注量下单主动和批量主动的性能比较

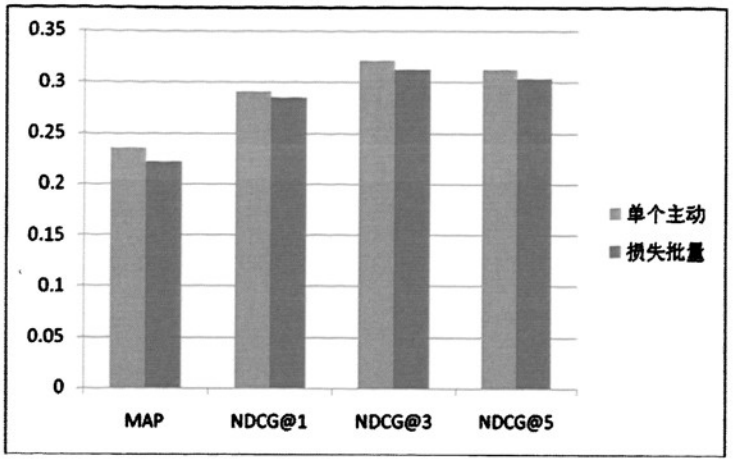

图 4.2 TREC.Gov 数据集下相同标注量下单主动和批量主动的性能比较

表 4.1 和 4.2 比较执行两种算法所需的时间代价，因为两种算法的标注量一样，标注时间一样，时间只计算了迭代训练排序模型和选择样本的时间。

表 4.1 OHSUMED 数据集下相同标注量下单主动和批量主动的时间比较

主动算法	时间
单个主动	约 16 个小时
函数批量	1 小时 32 分钟

表 4.2 TREC.Gov 数据集下相同标注量下单主动和批量主动的时间比较

主动算法	时间
单个主动	约 18 个小时
函数批量	约 2 个小时

从上列的图表可以看出在相同标注量的情况下，批量主动和单一主动算法的性能相差不大，但所消耗的时间则相差很大，因此批量主动排序可以在保证主动排序性能的情况下减少主动排序得到排序模型的时间，降低主动排序的学习代价。

4.5.2.2　与基于夹角差异的批量主动算法的改变率比较

基于损失函数的批量主动排序算法与基于夹角差异的批量排序算法的比较实验，用来比较基于损失函数的批量主动算法和基于夹角差异的批量算法的性

能差异。夹角差异批量主动排序算法采用第三章介绍的批量算法。两种批量算法初始训练集都为前 5 个查询，然后每次选择 50 个未标注样本进行标注，总共学习 9 轮，比较两个算法在把选择的 50 个未标注样本加到训练集后对排序模型的准确率提高程度，两种算法同样是在 OHSUMED 和 TREC.Gov 两个数据集上实验，评价指标是平均查准率 MAP。图 4.3 是 OHSUMED 数据集上损失批量和夹角批量每次学习的 MAP 比较，图 4.4 是 TREC.Gov 数据集上损失批量和夹角批量每次学习的MAP比较。其中初始点是初始训练集上模型的性能数据。

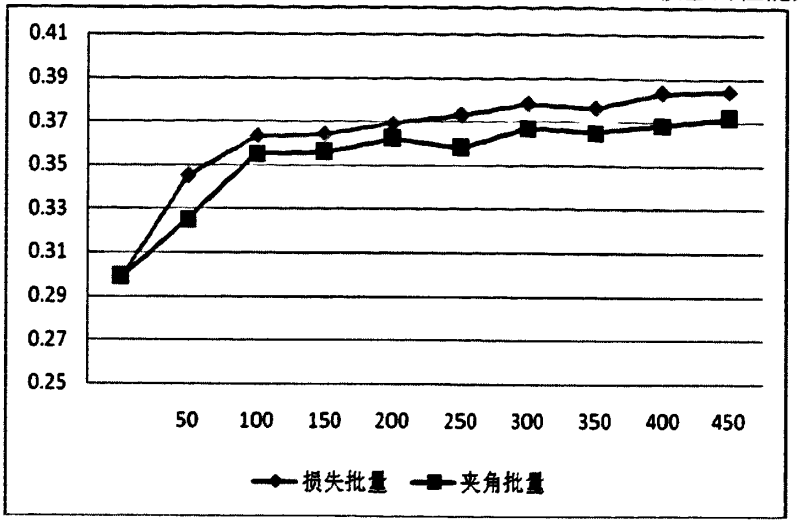

图 4.3 OHSUMED 数据集上损失批量和夹角批量每次学习的 MAP 比较

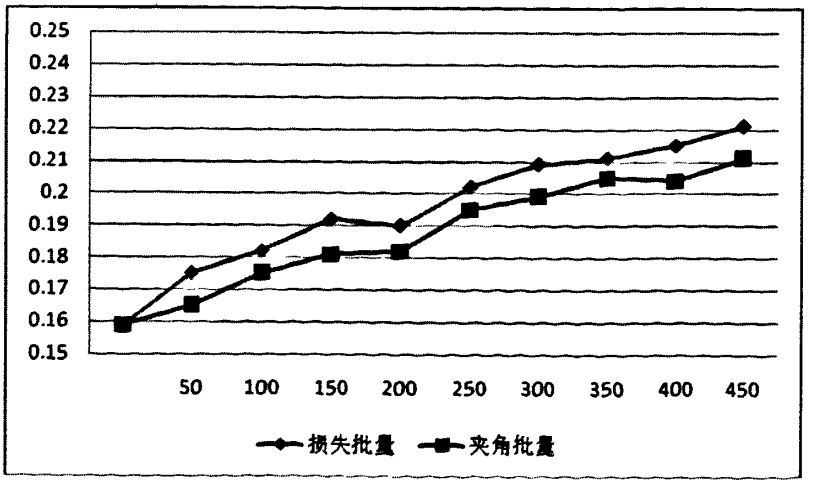

图 4.4 TREC.Gov 数据集上损失批量和夹角批量每次学习的 MAP 比较

从图 4.3 和图 4.4 可以看出，本文提出的基于损失函数的批量主动排序学习算法选择的样本比基于夹角差异的批量主动排序算法选择的样本，加入训练集重新训练后得到的排序模型性能更高，它们更能够提高排序模型的排序性能。因此在达到相同的排序性能的时候，与基于夹角差异的批量主动排序学习算法相比，基于损失函数的批量主动排序学习算法只需要更少的标注未标注样本，减少批量主动排序学习的标注量，降低批量主动排序学习的标注代价。

4.5.2.3　与基于夹角差异的批量主动算法的相同标注量性能比较

进行基于损失函数的批量主动学习算法和基于夹角差异的批量主动排序算法在相同的标注量下的最终性能比较实验，基于夹角差异的批量主动学习算法同样采用第三章介绍的批量算法。两种批量算法初始训练集都为前 5 个查询，然后每次选择 50 个未标注样本进行标注，总共学习 9 轮，比较 9 轮学习之后，批量主动排序学习最终得到的排序模型的排序性能，两种算法同样是在 OHSUMED 和 TREC.Gov 两个数据集上实验，评价指标是 MAP 和 NDCG。图 4.5 是 OHSUMED 数据集上损失批量和夹角批量的最终性能比较，图 4.6 是 TREC.Gov 数据集上损失批量和夹角批量的最终性能比较。

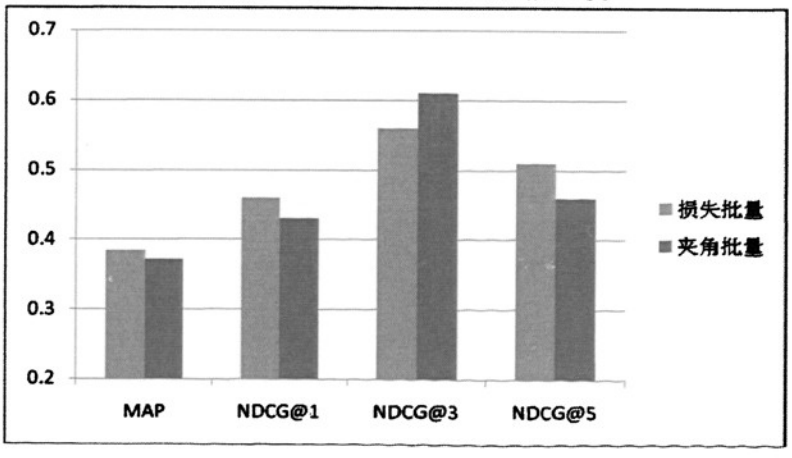

图 4.5 OHSUMED 数据集上损失批量和夹角批量的最终性能比较

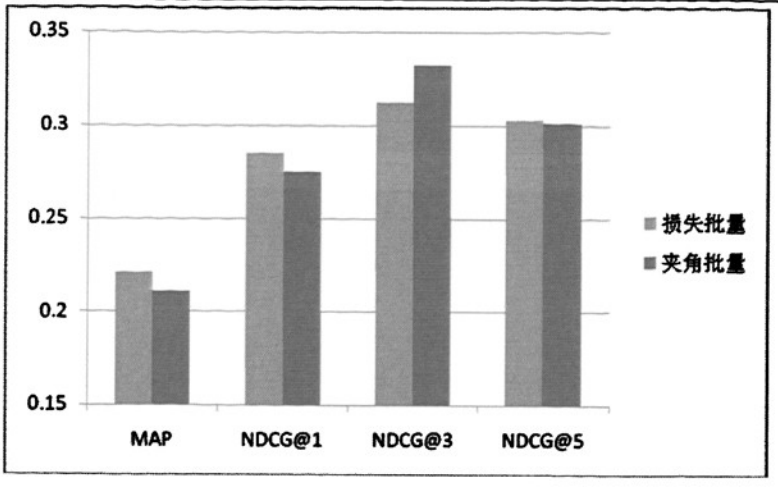

图 4.6 TREC.Gov 数据集上损失批量和夹角批量的最终性能比较

从 OHSUMED 数据集和 TREC.Gov 数据集的实验结果图 4.5 和图 4.6 可以看出，在相同标注量的情况下，不管是从评价指标 MAP，还是评价指标 NDCG，基于损失函数的批量主动排序算法比基于夹角差异的批量主动排序算法能够学习得到更好的排序模型。因此，基于损失函数的批量主动排序算法在批量主动排序算法中有其优越性，提升了批量主动排序的性能和效率。

第五章 结束语

本文的研究领域是信息检索；本文所关注的问题是信息检索中单个主动排序学习训练时间很多、数据标注代价仍然很大和效率不高的问题。针对这些问题，本文提出批量主动排序学习方法，一次能够找到多个值得标注的样本给用户标注，同时这多个标注的样本对排序模型性能的提升都有很大的价值。如此，以期减少主动学习的训练时间，以及在保证排序模型性能的前提下降低标注代价，提高主动排序学习的效率。

5.1 本文工作总结

对现阶段的主动排序学习算法研究分析，我们发现，主动排序学习每次只选择一个未标注样本标注，然后重新训练，需要大量的训练时间，同时标注人员标注下一个样本需要等待很长时间。这使主动学习的标注代价仍然很高，也不能通过多个标注人员并行标注来提高主动学习的效率。

针对上述问题，本文提出批量主动排序学习的思想，主动排序学习的时候，一次能够找到多个值得标注的样本给用户标注，这多个标注的样本对排序模型性能的提升都有很大的价值。

本文提出了两种批量主动排序学习算法，一种是基于夹角差异的批量主动排序学习算法，该算法通过加入批量选择的样本之间的夹角差异度量，来减少批量选择的样本之间的相似度，提高批量主动排序的性能。另一种是基于损失函数的批量主动排序学习算法，该算法直接从提高排序模型性能的损失函数入手，批量选择能够使损失函数达到最小值的那些样本进行标注。

基于夹角差异的批量主动排序研究。基于夹角差异的批量主动排序引入夹角差异度量来减少批量选择的样本之间的相似性，提高批量选择的样本对排序模型的贡献度。夹角差异度量即未选择的样本与已选择的样本之间的余弦差异度，选择差异度最大的未标注样本加入到批量选择样本集中，依次类推，从而选择满足条件数量的样本。

基于损失函数的批量主动排序研究。基于损失函数的批量主动排序算法，

直接从能够提高排序模型性能的损失函数入手，批量选择的依据在于使损失函数取得最小值，即批量选择的未标注样本能使损失函数达到最小值，这样可以一下子选择多个未标注样本。基于损失函数的批量主动排序学习具有很强的理论性。

在不同数据集上进行实验，评价以上两种批量主动排序学习算法，同时跟单样本主动排序学习算法，原始的批量主动排序学习算法等进行比较分析。

通过在两个大规模真实数据集上的实验表明，使用本文提出的算法可在保证排序模型性能的前提下，减少样本的标注量；在同等标注量的条件下，提高排序结果的正确率。

5.2 未来工作展望

在本文的研究工作中，依然有一些问题有待进一步研究。对于下一阶段工作，应从以下三个方面着手，这包括：

1、研究批量主动排序可以防止单个主动排序陷入局部最优的情况

单个主动排序学习算法每次都选择一个最值得标注的样本进行标注，这是一种贪婪算法，会发生陷入局部最优的情况。而批量主动学习算法是选择多个值得标注的样本进行标注，是从宏观上选择多个样本，可以防止陷入局部最优的情况发生，因此，下一步将从理论和实验两个方面验证批量主动排序的寻优能力好于单个主动学习。

2、提出以查询为标注单元（Query-Level）的批量主动排序学习算法。

排序问题不同于传统的分类等问题。在排序过程中，参与训练的基本单元是查询与文档组成的“查询-文档对”，不同查询之间对应的“查询-文档对”差异是很大的。因此，下一步提出以查询为标注单元（Query-Level）的基于列表（List-wise）的主动排序学习算法，考虑查询在主动学习过程中起到的重要作用。

3、使用更多的排序学习算法作为主动排序学习算法中的基本排序算法。

现有的主流排序学习方法多达十余种，究竟选择哪种或哪几种排序学习方法作为主动排序学习中的基本排序算法（Base Ranker），对于最终主动排序学习的结果和性能都有着很大的影响。下一步，考虑使用更多的排序学习算法作为主动排序学习算法中的基本排序算法。

参考文献

[1] Baeza-Yates, R., and Ribeiro-Neto, B. Modern Information Retrieval. New York, NY, USA: Addition Wesley, 1999.

[2] Blum A. and Mitchell T. Combining labeled and unlabeled data with co-training. In: Proceedings of the 11th Annual Conference on Computational Learning Theory, 1998. 92~100.

[3] Burges, C. J. C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. Learning to Rank using Gradient Descent. In: Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 2005.

[4] Cao Y., Xu J., Liu T., Li H., Huang Y., and Hon H. Adapting ranking SVM to document retrieval. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, Seattle, Washington, USA, pp. 186-193, 2006.

[5] Cao Z., Qin T., Liu T., Tsai M. and Li H. Learning to Rank: From Pairwise Approach to Listwise Approach. In: Proceedings of the 24th International Conference on Machine Learning. 2007.

[6] Chu W. and Ghahramani Z. Extensions of Gaussian Processes for Ranking: Semi-Supervised and Active Learning. Proceedings of NIPS Workshop, 2005.

[7] Chu, W. and Keerthi, S. New approaches to support vector ordinal regression. In: Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 145~152. 2005.

[8] Cohn D., Ghahramani Z., and Jordan M.I. Active Learning with Statistical Models. Artificial Intelligence Research, 1996, 4: 129-145.

[9] Crammer, K. and Singer, Y. Pranking with ranking. In: Proceedings of the conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, 2001.

[10] Craswell N., Hawking D., Wilkinson R., and Wu M. Overview of the TREC 2003 web track. In TREC, pages 78–92, 2003.

[11] 邓乃扬, 田英杰. 数据挖掘中的新方法——支持向量机. 北京: 科学出版社, 2004.

[12] Frank, E. and Hall, M. A Simple Approach to Ordinal Classification. In: Proceedings of the 12th European Conference on Machine Learning, Freiburg, Germany, 2001. 145~156.

[13] Freund Y., Seung H.S., Shamir E., and Tishby N. Selective Sampling Using the Query by Committee Algorithm. Machine Learning, 1997, 28: 133-168.

[14] reund, Y., Iyer, R., Schapire, R., and Singer, Y. An efficient boosting algorithm for combining preferences. Journal of Machine Learning. Research 4, 2003, 933–969.

[15] Harrington, E. Online ranking/collaborative filtering using the Perceptron algorithm. In:

Proceedings of the 20th International Conference on Machine Learning. Washington DC, USA, 2003, 250~257.

[16] Hastie T., Tibshirani R. and Friedman J. The Elements of Statistical Learning: Data mining, inference and prediction. Springer-Verlag, 2001.

[17] Herbrich, R., Graepel, T. and Obermayer, K. Large Margin Rank Boundaries for Ordinal Regression. Smola, A., Bartlett, P., Scholkopf, B., and Schuurmans, D., eds., Advances in Large Margin Classifiers. MIT Press, 2000, 115~132.

[18] Hersh W. R., Buckley C., Leone T. J., Hickam D. H. OHSUMED: An interactive retrieval evaluation and new large test collection for research. In: Proceedings of the 17th Annual ACM SIGIR International Conference on Research and Development in Information Retrieval, Dublin, Ireland 1994, 192~201.

[19] Jarvelin, K. and Kekalainen, J. Cumulated Gain-based Evaluation of IR Techniques. ACM Transactions on Information Systems, 2002. 20（4）:422~446.

[20] Joachims T. Transductive Inference for Text Classification using Support Vector Machines. In: Proceedings of International Conference on Machine Learning.1999.

[21] Joachims, T. Optimizing Search Engines Using Click-through Data. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA, 2002, 133~142.

[22] Kramer, S., Widmer, G., Pfahringer, B., and Degroeve, M. Prediction of ordinal classes using regression trees. Fundamenta Informaticae, 2001, 47:1~13.

[23] Lafferty, J. and Zhai, C. Document language models, query models, and risk minimization for information retrieval. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, Louisiana, USA, 2001, 111~119.

[24] Lancaster, F. W. Information retrieval systems: characteristics, testing and evaluation. 2nd Ed., New York: John Wiely and Sons, 1979.

[25] Lewis D., and Gale W. A sequential algorithm for training text classifiers. In: Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 1994, 148~156.

[26] Liu T., Xu J., Qin T., Xiong W, and Li H. LETOR: Benchmarking "Learning to Rank for Information Retrieval". In: Proceedings of SIGIR 2007 Workshop on Learning to Rank for Information Retrieval, Amsterdam, The Netherlands, 2007.

[27] Mitchell T. Mechine Learning. McGraw Hill, 1997.

[28] Muslea I., Minton S., and Knoblock A. Active + Semi-Supervised Learning = Robust Multi-View Learning. In Proceedings of the 19th International Conference on Machine Learning, Sydney, Australia, 2002. 435~442.

[29] Usunier N., Truong V., Massih R. A., and Gallinari P. Ranking with Unlabeled Data: A First Study. Proceedings of NIPS Workshop, 2005.

[30] Nallapati, R. Discriminative Models for Information Retrieval. In: Proceedings of the27th

Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, United Kingdom, 2004. 64~71.

[31] Ponte J. M. and Croft W. B. A language modeling approach to information retrieval. In: Proceedings of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, 1998. 275~281.

[32] Qin T., Liu T., Lai W., Zhang X., Wang D. and Li H. Ranking with Multiple Hyperplanes, In Proceedings of the 30th Annual International ACM SIGIR Conference, Amsterdam, The Netherlands, 2007.

[33] Qin T., Zhang X., Tsai M., Wang D., Liu T. and Li H. Query-level Loss Functions for Information Retrieval. Information Processing and Management, 2007.

[34] Robertson, S. and Hull, D. A. The TREC-9 Filtering Track Final Report. In: Proceedings of Text REtrieval Conference TREC-9, National Institute of Standards and Technology, NIST Special Publication 500-249, 2000, 25~40.

[35] Robertson, S. E., Walker S., Hancock-Beaulieu M. and Gatford M. Okapi in TREC3. In Proceedings of Text REtrieval Conference, Gaithersburg, USA. U.S. National Institute of Standards and Technology, NIST Special Publication 500-225: 1994. 109~126.

[36] Robertson, S., Zaragoza, H., and Taylor M. Simple BM25 extension to multiple weighted fields. In: Proceedings of the 2004 ACM CIKM International Conference on Information and Knowledge Management, Washington, DC, USA, 2004. 42~49.

[37] Salton, G. A Comparison between Manual and Automatic Indexing Methods. Journal of American Documentation, 1969, 20（1）:61~71.

[38] Salton, G., Buckley C., and Fox, E. Automatic query formulations in information retrieval. Journal of the American Society for Information Science, 1983, 34（4）: 262~280.

[39] Salton, G., Fox, E. A., and Wu, H. Extended Boolean information retrieval. Communications of the ACM, ACM Press, 1983, 26（11）: 1022~1036.

[40] Schohn, G. and D. Cohn, Less is more: Active learning with support vector machines. Porc.17th Annual International Conference on Machine Learning, 2000, 839~846.

[41] Shashua, A. and Levin, A. Ranking with large margin principle: two approaches. In: Thrun, S., Becker S., and Obermayer K. eds., Advances in Neural Information Processing Systems 15, Cambridge, MA: MIT Press, 2003, 937~944.

[42] Tong S. and Koller D. Support vector machine active learning with applications to text classification. Journal of Machine Learning Research. 2000, 999~1006.

[43] Vapnik, V. N. Statistical learning theory, John Wiley and Sons, New York, 1998.

[44] Vapnik, V. N. The Nature of Statistical Learning Theory Second Edition. Springer-Verlag New York, Inc., 2000.

[45] Vapnik, V. N. The Nature of Statistical Learning Theory. Springer, 1995.Vapnik, V. N. The Nature of Statistical Learning Theory Second Edition. Springer-Verlag New York, Inc., 2000.

[46] Wang Y., Kuai Y., Huang Y., Li D. and Ni W. Confidence-based Active Ranking for

Document Retrieval. Accepted by the 7th International Conference on Machine Learning and Cybernetics. Kunming, China. July, 2008.

[47] 王扬, 刘杰, 黄亚楼, 李栋, 蒯宇豪. 一种基于排序感知机的主动排序学习算法. 计算机工程, Vol. 24, 2008.

[48] Xu J. and Li H. AdaRank: A Boosting Algorithm for Information Retrieval. In Proceedings of the 30th Annual International ACM SIGIR Conference, Amsterdam, The Netherlands, 2007.

[49] 徐君. 用于信息检索的代价敏感排序学习算法研究: [博士学位论文]. 天津: 南开大学, 2005.

[50] Yarowsky D. Unsupervised word sense disambiguation rivaling supervised methods. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, 1994. 189~196.

[51] 易星. 半监督学习综述: [硕士学位论文]. 北京：清华大学，2004.

[52] 张莹. 基于自主学习的中文文本分类算法研究: [硕士学位论文]. 黑龙江：哈尔滨工业大学，2006.

[53] Zhu X. Semi-Supervised Learning Literature Survey. （Technical Report 1530）, Computer Sciences, University of Wisconsin. 2005.

[54] P. E. Gill, W. Murray, and M. H. Wright. Practical Optimization. Academic Press, 1981

[55] Yang Wang, Yuhao Kuai, Yalou Huang, Dong Li and Weijian Ni. Confidence-based Active Ranking for Document Retrieval. Proceedings of the 7th International Conference on Machine Learning and Cybernetics (ICMLC 2008), China. July, 2008.

[56] Platt, Fast training of support vector machines using sequential minimal optimization. Advances in kernel methods-support vector learning, Cambridge, MA:MIT Press, 1999, 185-208

[57] S. Hoi, R. Jin, J. Zhu, and M. Lyu. Batch mode active learning and its application to medical image classification. In Proceedings of the 23rd International Conference on Machine Learning, 2006

[58] S. Hoi, R. Jin, and M. Lyu. Large-scale text categorization by batch mode active learning. In proceeding of the International World Wide Web Conference, 2006

[59] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In Proceedings of the 17th International Conference on Machine Learning, 2000.

[60] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang. Representative sampling for text classification using support vector machines. In Proceeding of the 25th European Conference on Information Retrieval Research, 2003

致 谢

在本文工作的进行过程中，我得到了很多人的帮助，我要向他们表示感谢。

感谢我的导师黄亚楼教授，在黄老师的指导和帮助下我才得以完成学业。三年来黄老师让我学到了很多知识，提供了很多实践的机会，并给予精心的指导。特别是在本文的进展过程中，黄老师多次指导、督促，保证了我的论文工作的进度和工作质量。可以说，没有黄老师对我的教导就不会有我今天的进步。

感谢实验室的谢茂强老师，在进行信息检索项目时，不断地给予我谆谆教诲和悉心指导，让我在高水平的项目中锻炼自己，提升自己的研究能力和水平。同时，谢老师在他在我做研究遇到问题的时候督促我的学习并给予我做人处事方法的指导。谢老师他精湛的技术和踏实的研究态度是我今后学习的榜样。

感谢实验室的博士师兄王扬。王扬师兄在我的毕业论文工作中给予了大量的帮助，提供了丰富的意见和建议，对很多细节之处都提出了建设性的意见。在王扬师兄的帮助下我才得以完成本文。同时感谢信息检索项目组其他成员，李栋、倪维健、刘杰博士，刘金莉、郑楠等同学。非常感谢大家在我撰写论文工作过程中给与的帮助和鼓励。

感谢武涛、李超、邹剑、刘洋、任莉媛、刘金莉，在三年的同窗生活中，他们给与我莫大的帮助，给我在生活、学习中带来快乐，这段欢乐时光是我永远不会忘记的。

感谢杨波，郑楠，刘辛，杨俊丽，这些我可爱师弟师妹们。

衷心感谢我的父母。感谢你们含辛茹苦地哺育我长大、教会我做人并给了我一个宽松自由的成长环境。祝愿我的父母身体健康、生活快乐。

最后，祝愿培养我的南开大学软件学院和南开大学智能信息处理实验室蒸蒸日上，更创辉煌！

蒯宇豪

2009年4月

南开大学智能信息处理实验室

个人简历

一、个人信息：

姓名：蒯宇豪　　性别：男　　出生日期：1984-12-11　　籍贯：江苏镇江

二、学习经历：

2006-09 至 2009-06　南开大学软件学院　模式识别与智能系统专业　工学硕士

2002-09 至 2006-06　南开大学软件学院　软件工程专业　　工学学士

三、在学期间参与的项目：

1. 信息检索中基于损失函数优化的排序学习研究，国家自然科学基金项目，项目编号:60673009，执行期限：2007.01~2009.12。
2. Entity Search based on Text Mining，微软亚洲研究院高校合作项目，执行年限：2006.04~2008.06
3. 智能太阳能电池板系统，微软嵌入式挑战杯，进入全球前 30 强并去美国微软总部参加决赛，执行年限:2006.02-2006.07

四、在学期间完成的论文：

1. Yang Wang, Yuhao Kuai, Yalou Huang, Dong Li and Weijian Ni. Confidence-based Active Ranking for Document Retrieval. Proceedings of the 7th International Conference on Machine Learning and Cybernetics (ICMLC 2008). Kunming, China. July, 2008.
2. 王扬, 黄亚楼, 刘杰, 李栋, 蒯宇豪. 一种基于排序感知机的主动排序学习算法. 计算机工程, Vol. 24, 2008.

排序学习中的批量主动学习问题研究

作者： 蒯宇豪
学位授予单位： 南开大学