硕士学位论文

信息检索中的
主动排序学习问题研究

论文作者：王　扬

学　　号：2120050450

培养院系：软件学院

专　　业：计算机软件与理论

研究方向：信息检索

指导教师：黄亚楼 教授

南开大学研究生院

2008 年 4 月

MASTER'S DISSERTATION

Research on Active Learning to Rank in Information Retrieval

By: Yang Wang

Supervisor: Prof. Yalou Huang

Nankai University

April 2008

本文研究工作得到以下项目资助

国家自然科学基金项目

项目名称：信息检索中基于损失函数优化的排序学习研究

项目编号：60673009

执行期限：2007.1 ~ 2009.12

微软亚洲研究院资助研究项目

项目名称：Entity Search Based on Text Mining

执行期限：2006.4 ~ 2008.6

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的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

摘 要

随着网络技术的迅速发展和互联网规模的不断扩大，互联网已成为全球最大、最广泛使用的信息库。如何有效检索这些海量信息成为当前重要的研究课题，因而信息检索技术越来越受到人们的重视。在目前绝大多数的信息检索系统中，其检索出来的信息（如文档等）都以排序的方式返回给用户。因此，如何高效地为信息进行排序成为信息检索模型研究的核心问题之一。

近年来，利用监督学习的方法构造排序模型是信息检索领域中对排序方法研究的热点。排序感知机算法与排序支持向量机算法是基于监督学习的排序算法中的代表。然而，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。标注样本是一项耗时长、难度大且代价昂贵的工作。因此，找到一种能够降低标注代价的排序学习方法是十分必要的。

本文针对信息检索中排序学习样本标注代价过大的问题，提出把主动学习方法融入到排序学习中去，在查询函数的设计与构造、主动排序学习算法的研究与实现、实验设计与实验验证分析等方面开展研究。

本文提出了基于样本不确定程度的查询函数。使用本文提出的查询函数，排序模型可以通过计算每个样本对应不同序标号的确定程度，自动找出最不确定的样本，作为“最值得标注”的样本，减少了样本标注量，从而降低了标注代价。提出并实现了基于数据点的主动排序感知机（Active PRank）算法和基于有序对的主动排序支持向量机（Active RSVM）算法，并应用于文档检索和网页检索。

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，可以提高排序结果的正确率。

关键词：信息检索 排序学习 主动学习 感知机 支持向量机

Abstract

As the World Wide Web grows rapidly to become the largest and the most popular source of readily available information, it is increasingly important to be aware of the ways to access the large volume of information. At present, most information retrieval systems rank the retrieved information before presenting them to users. Thus, one of the key problems in information retrieval is ranking information effectively and efficiently.

In recent years, supervised learning approach for ranking is one of the hottest research topics in information retrieval, of which PRank and Ranking SVM are two representatives. However, like many other supervised approaches, one of the main problems with supervised learning to rank is the lack of labeled data, since labeling instances to create a rank model is time-consuming and costly. Thus, it is beneficial to minimize the number of labeled instances.

In this paper, we bring the idea of active learning into learning to rank for information retrieval, propose and realize two active ranking algorithms, to which referring as Active PRank and Active RSVM. Then, we apply the proposed algorithms to document retrieval and web retrieval.

Specifically, we present an uncertainty-based query function to estimate the uncertainty of each instance, and find out the instances providing more information for the ranker, reducing the labeling cost.

Experimental results on two real-world datasets show that, compared with the methods of passive learning to rank, PRank and RSVM, our proposed active ranking algorithms offer several advantages: first, given the same number of training instances, our approach provides higher ranking accuracy; second, to achieve the same ranking accuracy, our approach needs less number of labeled instances. In other words, our methods are capable of reducing the labeling cost greatly without decreasing the ranking accuracy.

Key words: Information Retrieval, Learning to Rank, Active Learning, Perceptron, Support Vector Machines.

目 录

第一章　绪论

1.1　引言

随着网络技术的迅速发展和互联网规模的不断扩大，互联网已经成为了全球最大、最广泛使用的信息库。如何有效检索这些海量信息成为当前重要的研究课题。因而信息检索技术越来越受到人们的重视。

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。自从上世纪五十年代起，信息检索技术逐渐对人类的科学研究和日常生活产生积极而又重要的影响。

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，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。排序学习就是一种基于有监督学习的排序方法，近年来逐渐成为信息检索领域的研究热点，很多研究者提出了大量的相关排序学习算法。

作为有监督的学习，排序学习对样本标注有着较高的要求。本文以信息检索为研究背景，关注信息检索中排序学习样本标注代价过大的问题，提出能够降低标注代价的排序学习方法，并应用于文档检索和网页检索中。

1.2　研究背景

本文的研究工作基于众多研究者们在信息检索、排序学习以及主动学习等领域取得的丰硕研究成果。在本节中，本文将对它们分别做简要的介绍，引出本文的研究问题。

1.2.1　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的查询相关的文档子集，是处理海量文本的重要手段。信息检索的基本模型可以用图 1.1 描述：用户根据其信息需求，总结出一个查询字符串提交给信息检索系统，信息检索

系统根据该查询在文档集中检索出与其相关的文档子集，经排序后返回给用户。

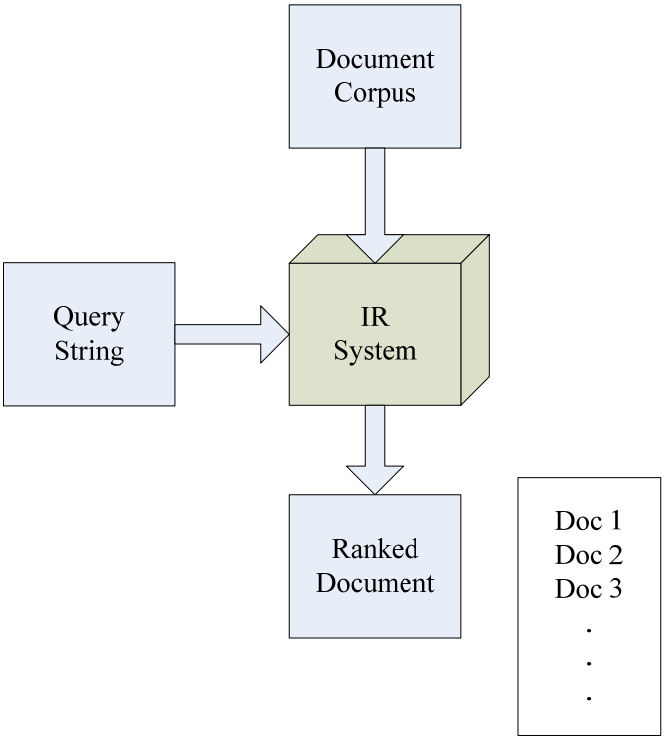

图 1.1　信息检索基本模型

信息检索研究的对象——信息，可以有多种表现形式，包括数字、图形、图像、语音和文本等，其中文本是最普遍的信息表达方式。本文研究的对象是文本。

影响一个信息检索系统性能的最关键因素是其检索模型，包括文档和查询条件的表示方法、文档评价和查询相关性的匹配策略、查询结果的排序方法和用户进行相关反馈的机制等。信息检索的模型研究起源于上世纪六十年代，经过相关科研人员近半个世纪的努力，一些有效的信息检索模型陆续提出并逐渐应用到相关的系统中。其中影响比较大的检索模型包括：布尔逻辑模型[24][39]、向量空间模型[37][38]、概率检索模型[35][36]、统计语言模型[31]、以及基于有监督学习的检索模型[4]。

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按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。

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。

1.3　本文动因

目前，所有的主流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。然而，标注样本是一项耗时长、难度大，而且代价昂贵的工作。因此，找到一种能够降低标注代价的排序学习方法是十分必要的。

以网页检索为例：现在从互联网上获取大量的文本是很方便的，据估计互联网上有超过30亿的网页，并且每年以150万的速度增长[52]，这些信息用人脑来检索是一个不现实的问题，这也是搜索引擎日趋火爆的原因。所以当进行网页分析或检索的过程中，训练集的数量是非常大的，就非常有必要采用主动学习的方法，以便在保持精度的情况下精简训练集，有效减少训练时间。

另外，在实际的应用当中，有些数据是很珍贵或者是非常难获得的，还有一些获取数据的环境非常复杂和危险。在这种情况下，寻求有限的、最有价值的数据就变得极为重要，主动学习于此有着重要作用。比如在一些工业事务中，一个训练样本可能要花费好几天的时间和成千上万的金钱，所以，若能最优的选择这些样本，无疑能节约大量的时间和金钱。

本文将围绕如何解决信息检索中排序学习数据标注代价过大的问题，展开相关的研究工作。本文将提出主动排序学习方法，在保证排序模型性能的前提下降低标注代价。

1.4　本文主要工作及目标

本文的研究领域是信息检索领域；本文关注的问题是信息检索中排序学习数据标注代价过大的问题；本文的目标是提出主动排序学习方法，在保证排序模型性能的前提下降低标注代价。

本文将以基于顺序回归的排序算法为例，对排序学习算法做出深入的研究探讨，针对目前主流的基于数据点和基于有序对的排序学习方法，提出并实现基于数据点的主动排序感知机（Active PRank）算法和基于有序对的主动排序支持向量机（Active RSVM）算法，并应用于文档检索和网页检索。具体讲，本文将在以下三个方面开展研究工作：

1、主动排序感知机（Active PRank）算法研究。

排序感知机[9]是基于数据点的排序学习的代表算法，其学习目标是在特征空

间中找到一个排序方向和 *k*-1 个阈值，这 *k*-1 个阈值把空间划分成 *k* 个连续的子空间，每一个子空间对应着一个序标号。排序感知机算法具有运算复杂度低，算法实现简单等优点。

本文将主动学习的思想引入到排序学习研究中，提出一种基于排序感知机的主动排序学习算法—Active PRank 算法。定义查询函数，选择出排序模型“最不确定”的样本作为“最值得标注”的样本，交由人工标注。使用主动排序感知机算法，可显著降低标注代价。

2、主动排序支持向量机（Active RSVM）算法研究。

排序支持向量机（Ranking SVM，RSVM）[17]算法是基于有序对的排序学习的代表算法。排序支持向量机算法的核心思想是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，并通过支持向量机（Support Vector Machine，SVM）求解。排序支持向量机算法具有模型稳定，排序结果较好等优点。

本文将提出一种基于不确定性的查询函数，找出排序支持向量机算法中“最不确定”的样本作为“最值得标注”的样本，完成主动排序支持向量机算法的研究与实现。

3、实验验证与结果分析。

信息检索是一个应用性很强的研究领域。因此，对本文所提出的算法进行实验验证、分析是非常必要的。将本文提出的两种主动排序学习算法应用于文本检索与网页检索，通过基于大规模真实数据集合的实验，验证算法在现实大规模信息检索应用上的有效性。

用于信息检索的排序模型需要大量的训练数据，而标注这些数据代价非常昂贵，限制了排序学习在信息检索中的有效应用。因此，研究如何在保证排序模型性能的前提下减少所需的标注样本量有着重大的意义：

1、本文把主动学习的概念引入到排序学习方法中，在保证排序性能的前提下减少所需的标注样本量，降低标注代价；

2、推动排序学习方法在信息检索领域的研究与应用；

3、研究成果除用于信息检索外，在其他需要排序学习的相关领域，如经济学、社会科学等领域都可有广泛应用。

1.5　本文组织结构

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。

第三章详细介绍基于数据点的排序学习方法，并在此基础上提出基于数据点的主动排序学习方法——主动排序感知机算法。描述主动排序感知机算法的排序决策函数，查询函数，更新过程以及算法流程。

第四章详细介绍基于有序对的排序学习方法，并在此基础上提出基于有序对的主动排序学习方法——主动排序支持向量机算法。描述主动排序支持向量机算法的排序决策函数，查询函数，更新过程以及算法流程。

第五章将把本文提出的有关算法应用于信息检索，介绍本文实验使用的数据集和评价指标，描述本文实验设计、实验流程和实验设置，给出相关实验结果并进行实验分析。

第六章是总结与展望，总结本文的工作，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[24][39]、向量空间模型[37][38]、概率统计模型[35][36]、统计语言模型[31]、基于有监督学习的检索模型[4]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[39]（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（公式 2.2）。

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

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来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：

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

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。

BM25[35]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　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。

2.2.1　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 $\vec{x}$ 的集合 X

$$X=\{\vec{x}_1,\cdots,\vec{x}_m\}\subseteq\mathbb{R}^n \tag{2.9}$$

和其对应的标号

$$Y=\{y_1,\cdots,y_m\} \tag{2.10}$$

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 $p(\vec{x},y)$ 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 $(\vec{x}_i,y_i)$ 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 $\vec{x}$ 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 $f(\vec{x})\neq y$ 的概率。

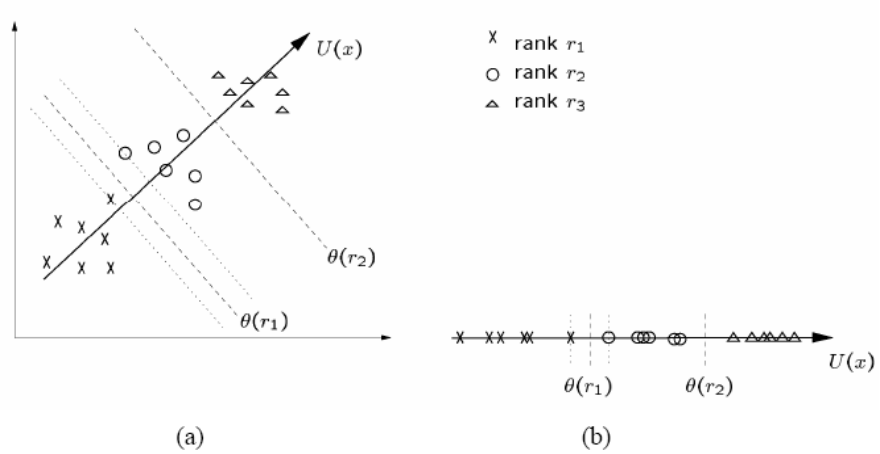

图2.1　（a） 用于排序的效应函数U（x），通过把数据点投影到U（x）上可以得到对数据的排序，$\theta(r_i)$是各个序类别之间的边界。（b） 数据在U（x）上的投影

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。

2.2.2　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[9]（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（$b_1 \le b_2 \le \cdots \le b_{k-1}$）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。

2.2.3　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 $\vec{x}_1$ 和 $\vec{x}_2$，且它们各自的等级数为 i 和 i+1，那么 $\vec{x}_1 - \vec{x}_2$ 就是正例，$\vec{x}_2 - \vec{x}_1$ 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 SVM^{light} 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]。

2.2.4　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[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。

2.3　机器学习中降低标注代价方法

机器学习的目标就是创建一个系统，该系统能够通过获取经验和数据改进自身的性能。对于很多自然学习的活动来说，这种经验和数据是通过与外界的交互而获得，比如采取行动，相互询问或通过实验。但是在大量机器学习的研究当中，很多情况下往往把学习者当成一个被动接受待处理数据的容器，这种被动的接受策略忽略了一个事实，就是在很多情况下学习者它自身有能力采取行动、收集数据，对它努力理解的这个世界作出自己的判断。目前，很多学者探讨怎样能够更有效地发挥学习者这个最有力工具的作用，并提出了很多方法，如自学习[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，以下将分别介绍这些学习方法。

2.3.1　自学习

自学习（Self Learning）是一种较早提出的降低标注代价方法。其中，Bootstrapping 算法[50]是自学习的代表算法。Bootstrapping 方法采取“步步为营”的思想，它从很小量的标注样本开始，通过度量已标注的数据与未标注的数据间相似关系，找到与已标注样本最相似的未标注样本，并把它也作为当前的这一类数据，视同已标注样本一样，并由此重新建立模型，进入下一轮迭代。经过多次迭代后，未标注样本逐渐被分别标注为各类别。

这种思路充分利用同一类数据间自身的相似性，使得在即使只有很少的数据被标注的情况下也能训练出代表这一类数据的模型。Bootstrapping 算法大大降低了人的工作量，并且在词义消歧等领域取得了令人满意的结果。

2.3.2　多视图学习

多视图学习[28]（Multi-view Learning）建立在如下假设的基础上：样本的特征存在着天然分割，表示的对象是能被多个特征子集描述，并且在每个特征子集都可以完全独立地学习目标概念。在现实世界中有很多这样的例子。例如：语音可以通过口形与声音识别，Web 页面可以通过内容与链接描述文字识别。对这种具有多个独立特征集对象的学习可看作多视图学习问题，是当前机器学习的热点问题之一。对这种多视图表示对象进行聚类是多视图学习的其中一个问题。

在多视图学习问题中，一个样本 x 是被不同视图中的特征描述。例如：一个有两个视图表示的领域中，一个已标注样本用三元组<x_1，x_2，1>表示，其中 1 是“标签”，x_1 与 x_2 是它在两个视图中的描述。相似的，一个未标注样本表示为<x_1，x_2, ?>。对于任意一个样本 x，$V_1(x)$ 表示 x 在视图 V_1 中的描述。

多视图学习算法具有两个重要性质，一个是视图的条件独立性，另一个是视图间的兼容性。

所谓视图的条件独立性，就是指进行多视图学习在一个有两个视图表示的领域中，对任何样本 x=<x_1，x_2>，给定其的标签 1，如果 x_1 与 x_2 是相互独立的，即：

$$\Pr[V_1(x)=x_1 \mid V_2(x)=x_2, x=1]=\Pr[V_1(x)=x_1, x=1] \tag{2.11}$$

$$\Pr[V_2(x)=x_2 \mid V_1(x)=x_1, x=1]=\Pr[V_2(x)=x_2, x=1] \tag{2.12}$$

那么，称这两个视图各自是条件独立的。

所谓视图间的兼容性，就是指在一个有两个视图表示的领域中，假定 f_1 是在视图 V_1 的目标概念，f_2 是在视图 V_2 的目标概念，f 是在全局的目标概念，如果对大多数样本 $x=<x_1, x_2>$，有：

$$f(x_1, x_2)=f_1(x_1)=f_2(x_2) \tag{2.13}$$

那么，称视图 V_1 与 V_2 是兼容的。

2.3.3　半监督学习

半监督学习[51][53]（Semi-Supervised Learning）是另一种运用未标注样本的学习方式，结合了有监督学习和无监督学习的特点。

半监督学习有多种具体的实现方式。其中，最为经典的是 Co-training 算法[2]，它是一种多视图半监督学习算法。它假设所有训练数据的特征都可以被划分为互相独立的两部分，每一部分都对分类的结果提供有用的信息，并且它们是一致的。Co-training 方法分别针对这两个部分的特征进行学习，两部分即相互独立，又互相监督。开始时仅针对已标注的数据，由两部分特征分别训练出两个模型。然后使用这两个模型处理同一条未标注样本，若二者得到的结果一致，那么就可以把这条数据标记为由模型预测得到的结果，并且把这条数据也作为一条已标记数据一样对待，并更新模型，一直迭代下去，即得到了最终的模型。

2.3.4　直推式学习

在传统的归纳式学习中，分类算法的目标是从有限的训练样本集中训练出一个对整个样本空间而言期望判别误差尽可能小的分类器。然而，这样的高标准在许多实际问题中没有必要，因为用户仅仅是对一些特定的样本进行识别和分类，希望能够对这一特定测试集获得误差尽可能小的分类。如果把这一特定测试集有机地加入到分类器的设计和训练过程中，则不但可以对这一特定测试集获得良好的分类效果，而且可以在很大程度上提高原有归纳式学习算法的推广性能。这就是直推式学习[20]（Transductive Learning）的基本思想。

Joachims 提出了直推式支持向量机（Transductive SVM，TSVM）算法。TSVM 首先按照某个规则估计无标签样本中的正例数 N。然后使用 SVM 学习算法对有标签样本训练出一个初始分类器。用初始分类器对无标签样本进行分类，对判别函数输出值最大的 N 个无标签样本暂时赋为正标签，其余的赋为负标签。对所有样本重新训练，对新得到的分类器，按一定的规则交换一对标签值不同的测试样本的标签符号。这一步骤反复执行，直到收敛。实验表明，该方法取得了较好的分类精度。

主动学习也是机器学习中降低标注代价的一种重要方法，本文在下一节进行详细的介绍。

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。

[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]、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。

[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。

[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。

第三章　主动排序感知机算法

主动排序感知机算法模型由五部分组成（*R, L, U, Q, S*）。其中：*R* 是一个基本排序模型。主动排序感知机算法使用排序感知机算法作为基本排序模型；*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，使用查询函数 *Q*，排序模型可以从 *U* 中找出那些“最不确定的样本”作为“最值得标注”的样本交由人工标注；*S* 是可以为通过 *Q* 找出的“最值得标注”未标注样本提供正确标签的指导者。

本章第一节在详细描述排序感知机算法的基础上，定义主动排序感知机算法排序决策函数，并给出公式描述；第二节进一步定义主动排序感知机算法查询函数；第三节描述主动排序感知机模型更新排序模型过程；第四节给出主动排序感知机算法。

3.1　排序感知机决策函数

3.1.1　排序感知机模型

从概念上讲，感知机（Perceptron）在神经生理学领域已被讨论了很多年，但作为一种学习机器的模型，是由 Frank Rosenblatt 在 20 世纪 60 年代初提出的，通常称作 Rosenblatt 感知机。感知机最初被用来解决模式识别问题，最简单的情况就是用于构造线性可分情况下的两类分类规则。

为了有效地构造分类规则，感知机利用了最简单的神经元模型的自适应特性，每个神经元都是有 n 个输入 $x=(x_1,x_2,...,x_n)$，和一个输出 $y\in\{-1,1\}$，输入与输出的函数依赖关系为：

$$y=\mathrm{sgn}\{\vec{w}\cdot\vec{x}-b\} \tag{3.1}$$

其中：$\mathrm{sgn}(t)$ 是符号函数，如果，$t>0$，$\mathrm{sgn}(t)=1$；如果，$t<0$，$\mathrm{sgn}(t)=-1$；b 是阈值，$\vec{w}$ 是权重向量。

空间被超平面 $\vec{w}\cdot\vec{x}-b=0$ 分开，向量 $\vec{w}$ 和阈值 b 决定了超平面的位置，学习

机器总共包含 n+1 个可调参数。在学习过程中，感知机为每一个神经元选择适当的系数。

感知机算法是一个简单的迭代过程，主要包括以下 3 个主要步骤：

1、初始化权重向量 $\vec{w}=0$，阈值 b=0；

2、如果下一个训练样本 $(\vec{x}_{k+1}, y_{k+1})$ 被正确分类，即 $y_{k+1}(\vec{w}_{k+1}\cdot\vec{x}_{k+1}-b)>0$，则所有系数维持原值不变，其中：$1\le k\le n-1$；

3、如果下一个训练样本 $(\vec{x}_{k+1}, y_{k+1})$ 被错误分类，即 $y_{k+1}(\vec{w}_{k+1}\cdot\vec{x}_{k+1}-b)<0$，则按下述规则修正系数：

$$\begin{aligned}\vec{w}_{k+1}&=\vec{w}_k+\lambda y_{k+1}\vec{x}_{k+1}\\ b_{k+1}&=b_k+\lambda y_{k+1}R^2\end{aligned}\qquad(3.2)$$

其中：$R=\max(\|\vec{x}\|)$，λ 是学习率。结果的返回值为最后一次更新的 $(\vec{w},b)$。

传统的感知机模型是一个在线的、错误驱动的学习机器，在简单的问题上表现出较强的推广能力，但难以解决非线性的分类和回归问题。

2001 年 Crammer 和 Singer 提出了用改进的感知机算法[9]（Perceptron Rank, PRank）进行排序，简称为 PRank 算法。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。

3.1.2　排序决策函数描述

定义主动排序感知机算法的排序决策函数

$$H(\vec{x})=\min_{r\in\{1,2,\cdots,k\}}\{r:\vec{w}\cdot\vec{x}-b_r<0\}\qquad(3.3)$$

其中：$\vec{w}$ 为排序模型对输入样本 $\vec{x}$ 每一维特征赋予的权重。b_r 为一组阈值（ $b_1\le b_2\le\ldots\le b_k$，通常设置 $b_k=\infty$ ），这 k 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，即满足所有 $b_{r-1}<\vec{w}\cdot\vec{x}<b_r$ 的样本 $\vec{x}$ 都有相同的排序结果。

对于每一个样本 $\vec{x}_i$，首先计算权重向量 $\vec{w}$ 与 $\vec{x}_i$ 的内积 $\vec{w}\cdot\vec{x}_i$，找出所有满足

$\vec{w}\cdot\vec{x}_i < b_r$ 中最小的 b_r，并将此 b_r 对应的序标号 $\vec{x}_i$ 作为排序模型对样本的预测排序结果。

排序决策函数 $H(\vec{x}) = \min_{r\in\{1,2,...,k\}}\{r : \vec{w}\cdot\vec{x} - b_r < 0\}$，阈值 $b_1, b_2, ..., b_k$ 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。因此，任何一个样本-排序结果对（$\vec{x}$，y），对所有 r=1, 2, … , y-1，应满足 $\vec{w}\cdot\vec{x}_i > b_r$；对所有 r=y, … , k-1，应满足 $\vec{w}\cdot\vec{x}_i < b_r$。设置所有满足 $\vec{w}\cdot\vec{x}_i > b_r$ 的 $y_r = +1$；所有满足 $\vec{w}\cdot\vec{x}_i < b_r$ 的 $y_r = -1$，即 y 对应一组向量 $(y_1, ..., y_{k-1}) = (+1, ..., +1, -1, ... -1)$，其中所有 y_r $+1$ 中最大的 $r = y - 1$。

若对所有的 r 都有 $y_r(\vec{w}\cdot\vec{x} - b_r) > 0$，那么预测结果与标注结果是一致的；

若至少存在一个 r 对应的 $y_r(\vec{w}\cdot\vec{x} - b_r) \le 0$，那么预测结果是存在错误的；需要更新 $\vec{w}$ 与 b。对所有 $y_r(\vec{w}\cdot\vec{x} - b_r) \le 0$ 的 b_r，将 b_r 替换为 $b_r - y_r$；同时，将 $\vec{w}$ 替换为 $\vec{w} + (\sum_{r:yr(\vec{w}\cdot\vec{x}-br)\le 0} y_r)\vec{x}$。整个更新过程可用图 3.1 表示[9]：

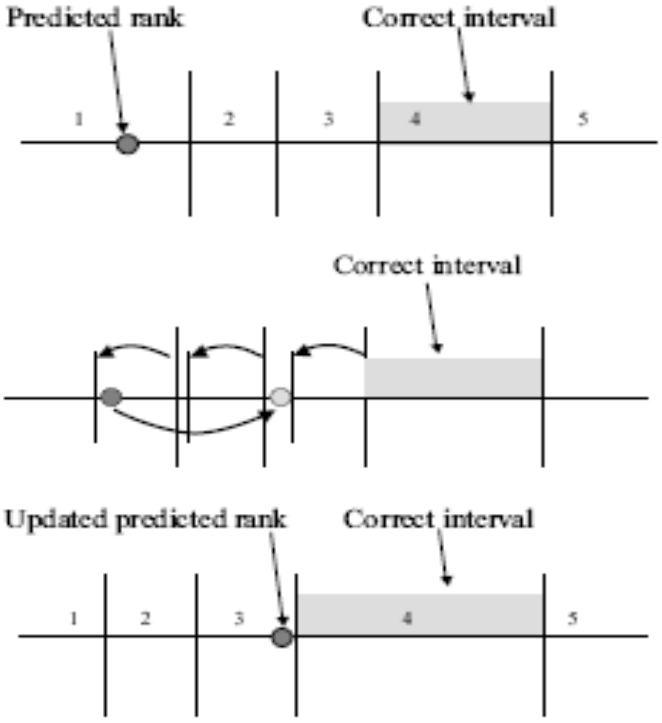

图 3.1　人工标注后排序模型的更新过程

3.2　查询函数

对于主动学习而言，初始的训练集中只有少量已标注样本，存在着大量的未标注样本。查询函数的目的是从大量的未标注样本中找到“最值得标注”的样本交由人工标注。“最值得标注”的样本是指在现有排序模型条件下，模型返回的排序结果置信度最低的那部分样本。即通过排序模型计算出的排序结果与两个序标号都非常接近，难以确定属于哪一个序标号的样本。

排序函数 $H(\vec{x})=\min_{r\in\{1,2,\ldots,k\}}\{r:\vec{w}\cdot\vec{x}-b_r<0\}$ 是根据样本 $\vec{x}$ 与权重 $\vec{w}$ 的内积 $\vec{w}\cdot\vec{x}$ 与 b_r 的差来确定其排序结果。对训练集中的未标注训练样本 $\vec{x}_i$，如果 $\vec{x}_i$ 的 $\left|\vec{w}\cdot\vec{x}-b_r\right|$ 越小，则 $\vec{x}_i$ 与排在其相邻类别样本的差异越不明显，即越难以确定 $\vec{x}_i$ 属于哪一个序标号。

因此，定义主动排序感知机算法的查询函数：

$$Q(\vec{x}_i)=\arg\min_{\substack{i=1,2,\ldots,m\\ j=1,2,\ldots,r-1}}\left|\vec{w}\cdot\vec{x}_i-b_j\right| \tag{3.4}$$

使用训练样本集中的已标注样本学习得到排序模型，并用此模型对训练集中所有未标注样本进行预测。对于训练集中的所有未标注样本 $\vec{x}_i$，首先通过计算 $f(\vec{x}_i)$，找出所有 b_r 中，离 $\vec{x}_i$ 最近的一个 b_r 设置为 b_j；再通过计算 $Q(\vec{x}_i)$，找出离 b_j 最近的一批 $\vec{x}_i$ 交由人工标注。

反复进行如上操作。在每次迭代过程中，使用 $Q(\vec{x}_i)$ 选择 T 个这样“最值得标注”的样本由人工标注，并将交由人工标注的 T 个样本从未标注样本集 U 移入到已标注样本集 L 中。

3.3　更新排序模型

在每轮迭代后，当原有排序模型对样本 $\vec{x}_i$ 给出的预测排序结果 $\hat{y}_i$ 与人工标注的排序结果 y_i 存在差异时，需要对排序模型进行更新。使用交由人工标注的那部分样本调整 $\vec{w}$ 和 b_r，帮助指导排序模型的更新。更新的过程借鉴排序感知机算法的思想。

排序决策函数 $H(\vec{x})=\min_{r\in\{1,2,\ldots,k\}}\{r:\vec{w}\cdot\vec{x}-b_r<0\}$；阈值 $b_1,b_2,\ldots,b_k$ 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。因此，任何一个样本-排序结果对（$\vec{x}$，y），对所有 r=1, 2, … , y-1，应满足 $\vec{w}\cdot\vec{x}>b_r$；对所有 r=y, … ,

k-1，应满足 $\vec{w}\cdot\vec{x}<b_r$。设置所有满足 $\vec{w}\cdot\vec{x}>b_r$ 的 $y_r=+1$；所有满足 $\vec{w}\cdot\vec{x}<b_r$ 的 $y_r=-1$，即 y 对应一组向量 $(y_1,...,y_{k-1})=(+1,...,+1,-1,...-1)$，其中所有 $y_r=+1$ 中最大的 $r=y-1$。

若对所有的 r 都有 $y_r(\vec{w}\cdot\vec{x}-b_r)>0$，那么预测结果与标注结果是一致的；

若至少存在一个 r 对应的 $y_r(\vec{w}\cdot\vec{x}-b_r)\le 0$，那么预测结果是存在错误的；需要更新 $\vec{w}$ 与 b。对所有 $y_r(\vec{w}\cdot\vec{x}-b_r)\le 0$ 的 b_r，将 b_r 替换为 b_r-y_r；同时，将 $\vec{w}$ 替换为 $\vec{w}+(\sum_{r:yr(\vec{w}\cdot\vec{x}-br)\le 0}y_r)\vec{x}$。

对于所有的 T 个人工标注样本进行如上处理，更新排序模型，完成一次迭代。反复迭代 N 次，建立排序模型，并用此模型对测试样本排序。对 U 中所有样本，从中选择出 $\left|\vec{w}\cdot\vec{x}^{n}-\vec{w}\cdot\vec{x}^{n-1}\right|$ 和 $\left|\vec{w}\cdot\vec{x}^{n}-\vec{w}\cdot\vec{x}^{n+1}\right|$ 最小的 T 个样本交由人工标注；反复循环如下操作 T 次；每次读入一个样本 $\vec{x}^t$ 的预测标记结果 $\hat{y}^t$ 和标注结果 y^t；

若 $\hat{y}^t=y^t$；则 $\vec{w}^{t+1}=\vec{w}^t$，$b_r^{t+1}=b_r^t$；

若 $\hat{y}^t\ne y^t$；则更新 $\vec{w}^t$，b_r^t

1．对 r=1, 2, …, k-1；如果 $y^t\le r$，则 $y_r^t=-1$；

否则 $y_r^t=+1$；

2．对 r=1, 2, …, k-1；如果 $y_r(\vec{w}\cdot\vec{x}-b_r)\le 0$，则 $\tau_r^t=y_r^t$；

否则 τ_r^t=0；

3．更新 $\vec{w}^{t+1}\leftarrow\vec{w}^{t}+(\sum_r\tau_r^t)\vec{x}^t$；

对 r=1, 2, …, k-1，更新 $b_r^{t+1}\leftarrow b_r^t-\tau_r^t$；

并将人工标记后的样本从 U 移入 L；最终输出性能较好的排序决策函数：$H(\vec{x})=\min_{r\in\{1,2,...,k\}}\{r:\vec{w}\cdot\vec{x}-b_r<0\}$。

3.4　主动排序感知机算法描述

主动排序感知机算法的流程是：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，以及每次迭代过程中交由人工标注的样本个数 T 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，使用查询函数从大量未标注样本 U 中选择出那些最不确定的样本作为“最值得标注”的样本交由人工标注，加入已标注样本集 L 中，并反复迭代；在每次迭代的过程中，更新排序模型，并使用测试数据检测排序模型。最终输出性能相对较好的排序函数 $H(\vec{x})=\min_{r\in\{1,2,...,k\}}\{r:\vec{w}\cdot\vec{x}-b_r<0\}$。

主动排序感知机算法可用算法 3.1 描述：

算法 3.1　主动排序感知机算法描述
给定： 已标注训练样本集 L； 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 T； 结束条件（算法达到设定精度或完成迭代次数）； 初始化：$\vec{w}_1 = 0$，$b_1 = b_2 = ... = b_{k-1} = 0$，$b_k = \infty$； 1、使用 L 建立排序模型 H； 2、使用 $H(\vec{x})$ 对 U 进行排序； 3、使用 $Q(\vec{x}_i)$ 从 U 中选择 T 个“最值得标注”的样本，交由人工标注，并将这些样本从 U 移入 L； 4、更新排序模型； 5、重复步骤 2，3，4，直到满足结束条件； 输出：$H(\vec{x}) = \min_{r\in\{1,2,...,k\}}\{r : \vec{w}\cdot\vec{x} - b_r < 0\}$。

第四章　主动排序支持向量机算法

主动排序支持向量机算法模型由五部分组成（R, L, U, Q, S）。其中，R 是一个基本排序模型。主动排序支持向量机算法使用排序支持向量机算法作为基本排序模型；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，使用查询函数 Q，排序模型可以从 U 中找出那些“最不确定的样本”作为“最值得标注”的样本交由人工标注；S 是可以为通过 Q 找出的“最值得标注”未标注样本提供正确标签的指导者。

本章第一节在详细描述排序支持向量机算法的基础上，定义主动排序支持向量机算法排序决策函数，并给出公式描述；第二节进一步定义主动排序支持向量机算法查询函数，并详细介绍相似度度量的方法；第三节描述主动排序支持向量机模型更新排序模型过程，最后给出主动排序支持向量机算法。

4.1　排序支持向量机决策函数

4.1.1　排序支持向量机模型

排序支持向量机[11][17][21][43][44][45]（RSVM）是解决排序学习问题的一类典型算法，它的核心思想是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，并且用支持向量机进行求解，寻找能将同一个簇内的最佳候选标记序列和其他候选标记序列分开的超平面。

排序学习需要找到排序函数 f，使得

$$\vec{x}_i \prec_X \vec{x}_j \Leftrightarrow f(\vec{x}_i) \prec_Y f(\vec{x}_j) \tag{4.1}$$

其中：$\prec_X$ 和 $\prec_Y$ 是定义在 X 空间和 Y 空间上的顺序关系。

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 $f(\vec{x}_i) \succ f(\vec{x}_j)$，函数 f 违反了公式（4.1）。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 l_{pref}。

$$l_{pref}(\vec{x}_1,\vec{x}_2,y_1,y_2,f(\vec{x}_1),f(\vec{x}_2))=\begin{cases}1 & (y_1\succ y_2)\wedge\neg(f(\vec{x}_1)\succ f(\vec{x}_2))\\ 1 & (y_1\prec y_2)\wedge\neg(f(\vec{x}_1)\prec f(\vec{x}_2))\\ 0 & else\end{cases} \tag{4.2}$$

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。

效应函数$U(\vec{x})$和$f(\vec{x})$有着如下的对应关系：

$$f(\vec{x})=r_i\Leftrightarrow U(\vec{x})\in[\theta(r_{i-1}),\theta(r_i)] \tag{4.3}$$

其中：$\theta(r_i),i=1,\cdots,k-1$为类别$r_i$和$r_{i+1}$之间的分类界面。

假设$U(\vec{x})$为线性函数，

$$U(\vec{x};\vec{w})=\langle\vec{w},\vec{x}\rangle \tag{4.4}$$

其中：$\vec{x}$是输入的向量，$\vec{w}$是排序模型的参数。则$U(\vec{x})$对于训练数据集合 S'中的第 i 个元素（有序对）没有错误，当且仅当

$$\begin{aligned}&z_i=+1\Rightarrow\langle\vec{w},\vec{x}_i^{(1)}\rangle-\langle\vec{w},\vec{x}_i^{(2)}\rangle>0\\&\Leftrightarrow\langle\vec{w},\vec{x}_i^{(1)}-\vec{x}_i^{(2)}\rangle>0\end{aligned} \tag{4.5}$$

$$\begin{aligned}&z_i=-1\Rightarrow\langle\vec{w},\vec{x}_i^{(1)}\rangle-\langle\vec{w},\vec{x}_i^{(2)}\rangle<0\\&\Leftrightarrow\langle\vec{w},\vec{x}_i^{(1)}-\vec{x}_i^{(2)}\rangle<0\end{aligned} \tag{4.6}$$

其中：$z_i=\Omega(y_i^{(1)},y_i^{(2)})$，$\Omega(y_i^{(1)},y_i^{(2)})$是一个指示函数，如果$y_i^{(1)}\succ y_i^{(2)}$，其值为 1，否则为－1。这样优先关系（preference relations）就被表达成两个向量之差的形式$\vec{x}_i^{(1)}-\vec{x}_i^{(2)}$，可以看成是两个向量的联合。

因此，可以得到：

$$\vec{x}_i\succ\vec{x}_j\Leftrightarrow\langle\vec{\omega},\vec{x}_i-\vec{x}_j\rangle>0 \tag{4.7}$$

在支持向量机的求解过程中，进一步假设在向量空间$\vec{x}_i^{(1)}-\vec{x}_i^{(2)}$中，$\Omega(y_i^{(1)},y_i^{(2)})=+1$和$\Omega(y_i^{(1)},y_i^{(2)})=-1$对应的向量$\vec{x}_i^{(1)}-\vec{x}_i^{(2)}$间存在着有限的边界（margin），这样，可以加强上面公式（4.5）和（4.6）的条件，得到

$$z_i\langle\vec{w},\vec{x}_i^{(1)}-\vec{x}_i^{(2)}\rangle>1-\xi_i,\quad i=1,\cdots,\ell \tag{4.8}$$

其中：ξ_i表示第 i 个限制条件被违反的程度。假设存在一个$\vec{w}$满足所有上述的ℓ个限制，其中最大化边界的$\vec{w}^*$可以由最小化二范数$\|\vec{w}\|^2+C\sum_{i=1}^{\ell}\xi_i$来得到，此时，

它也就是在最大化排序问题中不同类之间的边界，这种方法和传统的支持向量机非常类似，得到的二次规划问题如下：

$$\begin{aligned} &\min \left\|\vec{w}\right\|^2 + C\sum_{i=1}^{\ell}\xi_i \\ &\text{s.t. } \xi_i \geq 0, \quad z_i\left\langle \vec{w}, \vec{x}_i^{(1)} - \vec{x}_i^{(2)} \right\rangle > 1-\xi_i, \quad i=1,\cdots\ell \end{aligned} \tag{4.9}$$

图 4.1 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_i＝+1 的有序对，实心圆点代表了 z_i＝－1 的有序对。公式（4.9）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 4.1 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 4.1 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。

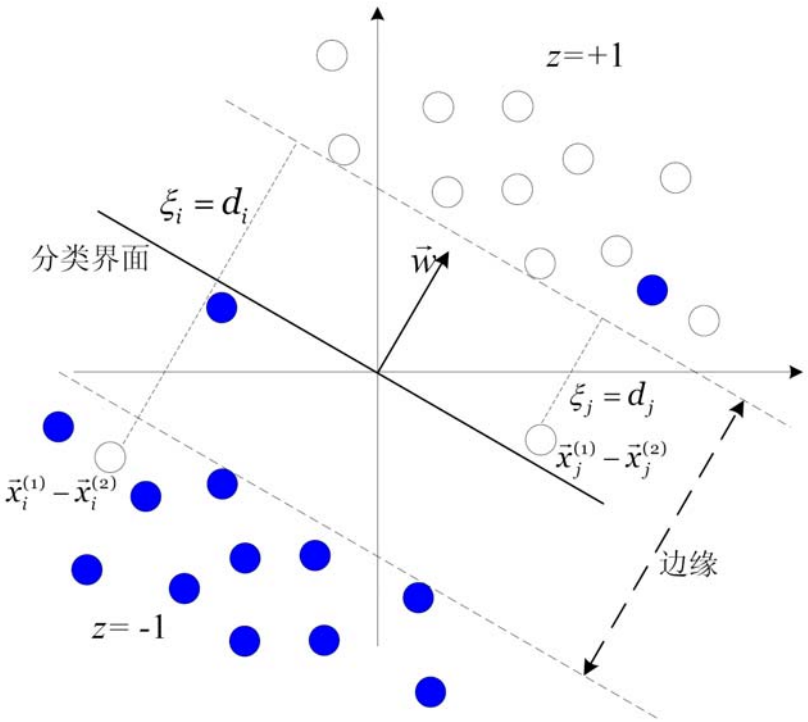

图 4.1　排序支持向量机的几何解释

4.1.2　排序决策函数描述

定义主动排序支持向量机算法的排序决策函数 $f(\vec{x}_i)$：

$$f(\vec{x}_i)=\min\sum_{i=1}^{\ell}\left[1-z_i\left\langle\vec{w},\vec{x}_i^{(1)}-\vec{x}_i^{(2)}\right\rangle\right]_+ +\lambda\left\|\vec{w}\right\|^2 \qquad (4.10)$$

其中：$\lambda=1/(2C)$，下标“+”表示正值部分，$[x]_+=\max(0,x)$。

公式（4.10）是主动排序支持向量机算法的排序决策函数，它由对模型的惩罚项和在训练数据集合上的经验损失项两部分组成。其中惩罚项只与模型的参数有关，越复杂的模型，对其的惩罚也就越高，对于线性模型，其值为模型向量的 2 阶范数；经验损失代表模型对训练数据集的拟合程度，经验损失项由 ℓ 个子项组成，其中 ℓ 为训练样本中有序对的个数，第 i 个子项对应着排序模型对第 i 个有序对 $\vec{x}_i^{(1)}-\vec{x}_i^{(2)}$ 的拟合程度，模型对训练数据拟合的越好，经验损失越小。

假设有序对 $\vec{x}_i^{(1)}-\vec{x}_i^{(2)}$ 所对应的 $z_i=+1$（意味着 $\vec{x}_i^{(1)}\succ\vec{x}_i^{(2)}$），如果 $\left\langle\vec{w},\vec{x}_i^{(1)}-\vec{x}_i^{(2)}\right\rangle\geq 1$（对第 i 个有序对的预测正确且在正边缘之外），则模型在第 i 个有序对上的损失为 0，否则，其在第 i 个有序对上的损失为其预测值和 1 之间的差值 $1-\left\langle\vec{w},\vec{x}_i^{(1)}-\vec{x}_i^{(2)}\right\rangle$；假设有序对 $\vec{x}_i^{(1)}-\vec{x}_i^{(2)}$ 所对应的 $z_i=-1$（意味着 $\vec{x}_i^{(1)}\prec\vec{x}_i^{(2)}$），如果 $\left\langle\vec{w},\vec{x}_i^{(1)}-\vec{x}_i^{(2)}\right\rangle\leq -1$（对第 i 个有序对的预测正确且在负边缘之外），则模型在第 i 个有序对上的损失为 0，否则，其在第 i 个有序对上的损失为其预测值和-1 之间的差 $\left\langle\vec{w},\vec{x}_i^{(1)}-\vec{x}_i^{(2)}\right\rangle-(-1)$。综合以上两种情况，如果对某一个有序对 $\vec{x}_i^{(1)}-\vec{x}_i^{(2)}$ 的预测满足 $z_i\left\langle\vec{w},\vec{x}_i^{(1)}-\vec{x}_i^{(2)}\right\rangle\geq 1$，则模型在第 i 个有序对上的损失为 0，否则其损失为 $1-z_i\left\langle\vec{w},\vec{x}_i^{(1)}-\vec{x}_i^{(2)}\right\rangle$。

经验损失相对于 $z\left\langle\vec{w},\vec{x}^{(1)}-\vec{x}^{(2)}\right\rangle$ 的变化曲线如图 4.2 所示，损失函数在（1，0）点有一个转折，像一个铰链，因此这种损失函数也被成为 Hinge（铰链）损失。

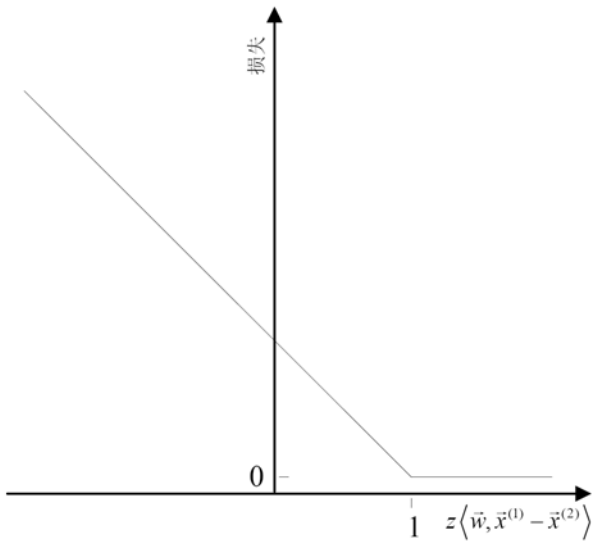

图 4.2　排序支持向量机的 Hinge 损失函数，横轴为 $z\langle \vec{w}, \vec{x}^{(1)} - \vec{x}^{(2)} \rangle$

4.2　查询函数

4.2.1　查询函数描述

查询函数的设计与实现是主动排序支持向量机算法模型中的最主要的环节。使用查询函数，排序模型可以自动的从未标注样本中查找出最值得标注的样本，交由人工标注，降低标注代价。

传统的主动学习算法认为，那些学习模型难以预测的样本比那些学习模型容易预测的样本可以为学习模型提供更多的信息。而学习模型对于样本预测的难易程度可以用这些样本的不确定性来度量。

主动排序支持向量机算法定义了一个基于不确定性的查询函数。查询函数计算每一个未标注训练样本集中的未标注样本的不确定程度。使用查询函数，排序模型可以从大量的未标注样本中找到“最不确定”的样本“最值得标注”的样本交由人工标注。

定义主动排序支持向量机算法的查询函数：

$$Q(\vec{x}_i)=\arg\min\left|\max_{j_1=1,\ldots,r} avg\ sim(\vec{x}_i,\vec{x}_{j_1})-\max_{j_2=1,\ldots,r,j_2\neq j_1} avg\ sim(\vec{x}_i,\vec{x}_{j_2})\right| \tag{4.11}$$

对于所有的未标注训练样本 $\vec{x}_i$，计算 $\vec{x}_i$ 与所有 $\vec{x}_j$ 之间的平均相似度 $avgsim(\vec{x}_i,\vec{x}_j)$。其中 $\vec{x}_j$ 是具有相同序标注的已标注训练样本。这样，对于每一个 $\vec{x}_j$，使用查询函数计算后，排序模型会自动返回 r 个数值（r 为序标号的个数），每个数值代表未标注训练样本 $\vec{x}_i$ 与具有相同序标注的已标注训练样本的一批 $\vec{x}_j$ 之间的平均相似度。使用这些数值中的最大的减去次大的，找出那些差值最小的 $\vec{x}_i$。这些差值小的 $\vec{x}_i$ 即为最不确定的样本，交由指导者进行标注。

4.2.2　相似度度量

主动排序支持向量机算法使用 BM25[35]值作为样本间相似度的度量。BM25 是一种经典的无监督排序算法。

$\vec{x}_m$ 和 $\vec{x}_n$ 分别表示两个未标注训练样本的特征向量，$\vec{x}_m$ 与 $\vec{x}_n$ 的相似度定义为：

$$sim(\vec{x}_m,\vec{x}_n)=\vec{x}_m(BM25\ score)-\vec{x}_n(BM25\ score) \tag{4.12}$$

其中：$\vec{x}_m(BM25\ score)$ 表示 $\vec{x}_m$ 的 BM 25 值。

对查询 Q 中的每一个检索单元 ω_i，一共有三个权值与之相关：

$$U=\frac{(k_2+1)\psi}{k_2+\psi} \tag{4.13}$$

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）。

$$V=\frac{(k+1)\phi}{k(1-b+bL)+\phi} \tag{4.14}$$

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。

$$W=\log\left(\frac{r+0.5}{(R-r)+0.5}\Big/\frac{(n-r)+0.5}{(N-n)-(R-r)+0.5}\right) \tag{4.15}$$

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：

$$SC(Q,D)=\sum_{\omega\in Q}UVW \tag{4.16}$$

使用公式（4.16）可以计算出每一个未标注训练样本 $\vec{x}_i$ 的 BM25 值 $\vec{x}_i(BM25\ score)$，并将此值公式（4.12）得到样本间相似度。

4.3　主动排序支持向量机算法描述

在每轮迭代后，当原有排序模型对样本 $\vec{x}_i$ 给出的预测排序结果 $\hat{y}_i$ 与人工标注的排序结果 y_i 存在差异时，需要对排序模型进行更新。使用交由人工标注的那部分样本调整 $\vec{w}$ 和 b_r，帮助指导排序模型的更新。更新的过程与排序支持向量机算法相同。

主动排序支持向量机算法的流程是：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，以及每次迭代过程中交由人工标注的样本个数 T 和算法的结束条件。使用排序支持向量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，使用查询函数从大量未标注样本 U 中选择出那些最不确定的样本作为“最值得标注”的样本交由人工标注，加入已标注样本集 L 中，并反复迭代；在每次迭代的过程中，更新排序模型，使用测试数据检测排序模型，并使用评估函数对结果进行评价。最终输出性能相对较好的排序函数 $f(\vec{x}_i)=\min\sum_{i=1}^{\ell}\left[1-z_i\left\langle\vec{w},\vec{x}_i^{(1)}-\vec{x}_i^{(2)}\right\rangle\right]_+ +\lambda\left\|\vec{w}\right\|^2$。

主动排序支持向量机算法可用算法 4.1 描述：

算法 4.1　主动排序支持向量机算法描述

给定：

已标注训练样本集 L；

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 T；

结束条件（算法达到设定精度或完成迭代次数）；

初始化：

1、使用 L 建立排序模型 H；

2、使用 $f(\vec{x})$ 对 U 进行排序；

3、使用 $Q(\vec{x}_i)$ 从 U 中选择 T 个“最值得标注”的样本，交由人工标注，并将这些样本从 U 移入 L；

4、更新排序模型；

5、重复步骤 2，3，4，直到满足结束条件；

输出：

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

第五章　主动排序学习在信息检索中的应用

5.1　信息检索实验流程

随着互联网络的飞速发展和信息的日益丰富，人们越来越依赖搜索引擎来寻找所需信息，而在搜索引擎的背后，是信息检索技术在起作用。在本章中，我们把本文所提出的两种主动排序学习算法直接应用于信息检索中，包括文档检索和网页检索。

一个标准的信息检索系统可用图 5.1 描述：信息检索研究从一定规模的文档集合（document collection）中找出满足用户信息需求的信息，其目标是根据用户输入的查询，按照相关度对检索到的信息进行排序返回给用户。

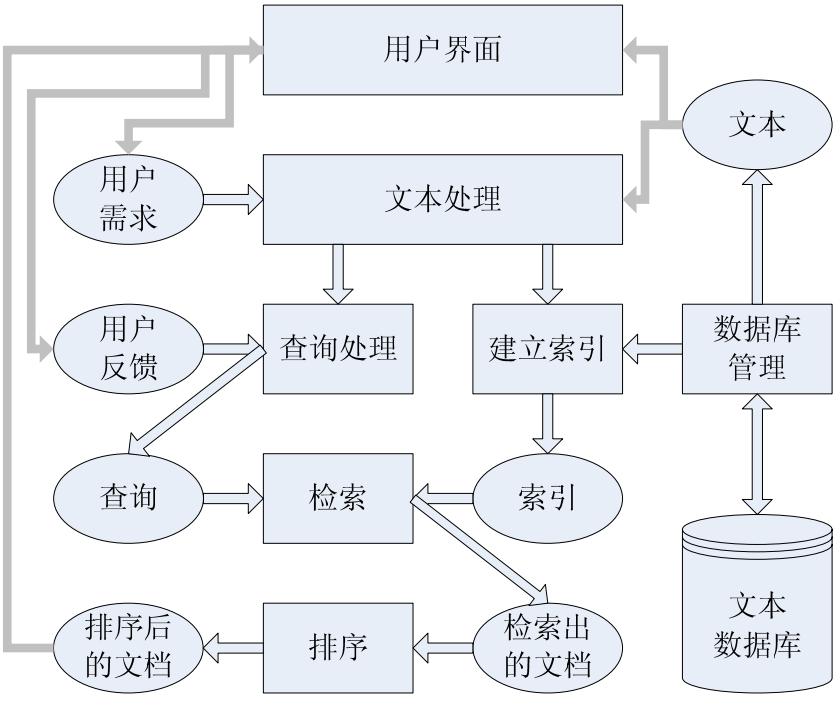

图 5.1　信息检索系统结构图

目前，在绝大多数的信息检索系统中，其检索出来的信息都以排序的方式返回给用户，这种方式使得用户最先得到最相关的文档，同时可以根据不同的需求自由地选择浏览信息的数量，因此，信息检索的核心问题也就归结为如何

高效准确的为文档进行排序。

对于排序学习算法实验，主要包括以下几个步骤：数据预处理、特征提取、训练排序模型、测试排序模型及结果评价等。图 5.2 为进行实验的主要步骤：实验数据与查询经过预处理算法和特征提取算法，形成训练数据；使用排序学习算法对训练数据进行训练，得到排序模型；使用测试数据对排序模型进行测试，并使用评价指标算法得出实验结果。

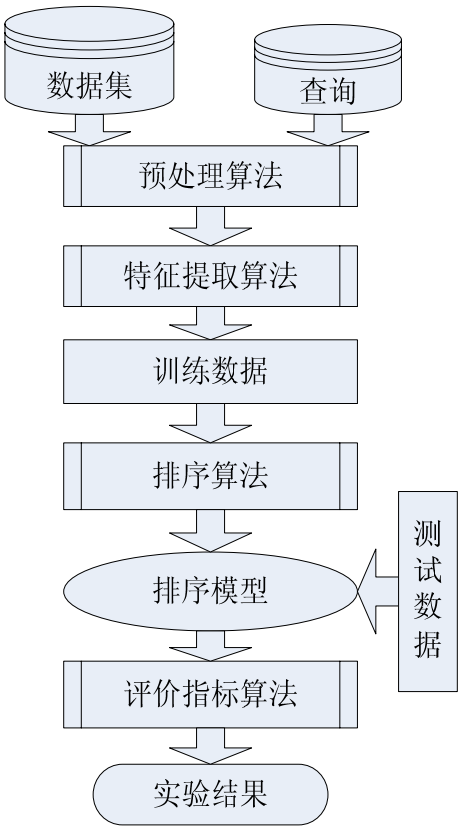

图 5.2　排序学习实验流程图

5.2　实验用数据集

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。

5.2.1　OHSUMED 数据集

OHSUMED 数据集合[18]曾在国际文档检索竞赛 TREC-9[34]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。

5.2.2　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 年早期.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。

5.3　检索性能评价指标

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。

5.3.1　MAP

MAP[1]（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：

$$AvgP_i = \sum_{j=1}^{M} \frac{Precision(j) \times \mathrm{pos}(j)}{\text{number of documents relevant to } Q_i} \tag{5.1}$$

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：

$$MAP = \frac{\sum_i AvgP_i}{\text{number of queries}} \tag{5.2}$$

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。

5.3.2 NDCG

NDCG[19]（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。

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

$$NDCG@r = N_r \bullet \sum_{j=1}^{r} \frac{2^{r(j)}-1}{\log(1+j)} \qquad (5.3)$$

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。

5.4　主动排序学习实验步骤

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 5.3 所示。

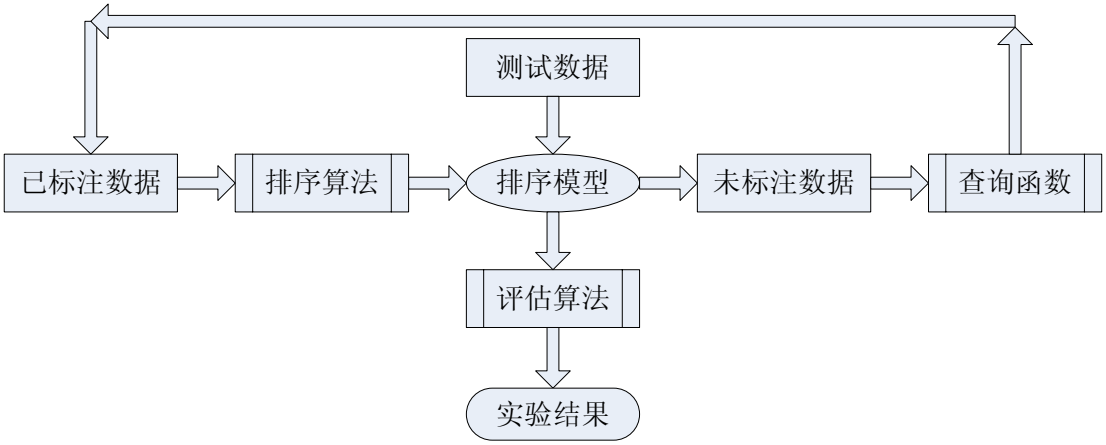

图 5.3　主动排序学习实验流程图

本文使用排序感知机算法作为主动排序感知机算法的对比算法，使用排序支持向量机算法作为主动排序支持向量机算法的对比算法。

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，这些特征在文献[26][30]中使用过。

对于 OHSUMED 数据集合，表 5.1 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 5.1　OHSUMED 数据集实验使用特征列表

1. $\sum_{q_i \in q \cap d} c(q_i, d)$	2. $\sum_{q_i \in q \cap d} \log(c(q_i, d)+1)$
3. $\sum_{q_i \in q \cap d} \log \frac{c(q_i, d)}{\lvert d \rvert}$	4. $\sum_{q_i \in q \cap d} \log(\frac{c(q_i, d)}{\lvert d \rvert}+1)$
5. $\sum_{q_i \in q \cap d} \log(idf(q_i))$	6. $\sum_{q_i \in q \cap d} \log(\log(idf(q_i)))$
7. $\sum_{q_i \in q \cap d} \log(\frac{\lvert C \rvert}{c(q_i, C)}+1)$	8. $\sum_{q_i \in q \cap d} \log(\frac{c(q_i, d)}{\lvert d \rvert} \cdot idf(q_i)+1)$
9. $\sum_{q_i \in q \cap d} c(q_i, d) \log(idf(q_i))$	10. $\sum_{q_i \in q \cap d} \log(\frac{c(q_i, d)}{\lvert d \rvert} \cdot \frac{\lvert C \rvert}{c(q_i, C)}+1)$
11. $BM25\ score$	12. $\log(BM25\ score)$
13. *LMIR with DIR smoothing*	14. *LMIR with JM smoothing*
15. *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[26]等作为特征，如表 5. 2 所示。

表 5.2　Doc .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*

续表 5.2　Doc .gov 数据集实验使用特征列表

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 5. 3 所示。

表 5.3　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。

5.5　实验结果及分析

5.5.1　主动排序感知机算法实验结果及分析

本小节实验的目的是验证本文提出的主动排序感知机（Active Prank）算法，使用排序感知机（PRank）算法作为对比算法，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，验证本文提出的主动排序感知机算法的性能。

本小节的实验共分三组：1．Active PRank 算法与 PRank 算法随标注量增加 MAP 结果变化图；2．相同标注量条件下 Active PRank 算法与 PRank 算法 MAP 与 NDCG 结果比较；3．当排序模型 MAP 值达到同一正确率时，Active PRank 算法与 PRank 算法标注量的比较。

5.5.1.1　OHSUMED 数据集结果及分析

1．本实验使用的训练集中，包括 100 条已标注训练样本和大量的未标注样本。在每次迭代过程中，通过查询函数 Q 从未标注样本中选择 T=50 个样本，交由人工标注，并反复循环 N=10 次，共标注 600 条样本。共进行五组实验，对结果取平均值。图 5.4 是使用 PRank 算法随机标注样本和 Active PRank 算法选择性标注样本，排序结果随标注量增加，MAP 变化的对比图。

从图 5.4 可以看出，使用 PRank 算法和 Active PRank 算法进行排序，排序结果的正确率（MAP 值）随着标注量的增加均逐渐提高，且使用 Active PRank 算法选择性添加样本进行标注的结果正确率要优于使用 PRank 算法随机添加样本进行标注。

2．在同样标注 600 条样本的条件下，分别使用 PRank 算法随机选择样本标

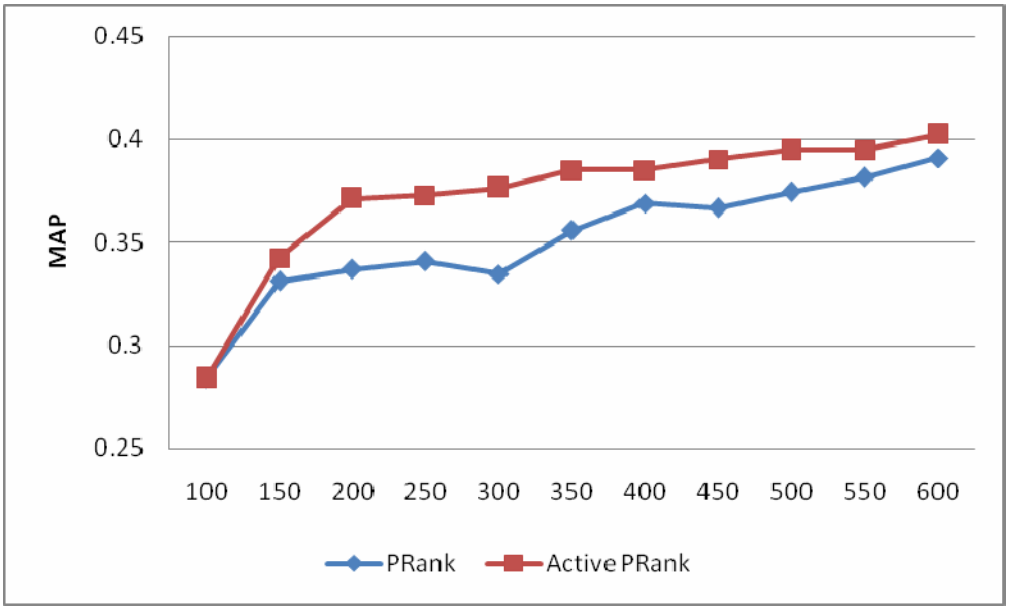

图 5.4　Active PRank 算法、PRank 算法随标注量增加结果变化图

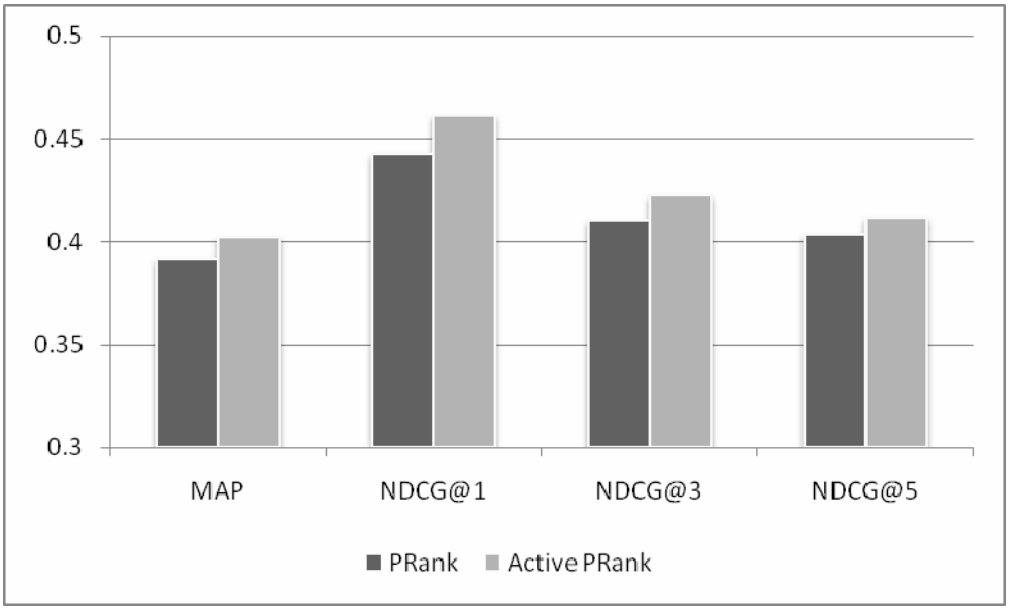

图 5.5　相同标注量条件下 Active PRank 算法与 PRank 算法结果比较图

注与本文提出的 Active PRank 算法选择“最值得标注”的样本进行标注，使用相同的测试集进行对比实验。实验结果比较如图 5.5 所示。

从图 5.5 中可以看出，在同样标注 600 条样本的条件下，Active PRank 算法的 MAP 值和 NDCG 值都高于 PRank 算法。可见，使用本文提出的 Active PRank 算法在同等标注量的条件下，无论是排序的整体效果，还是排序结果中顶部序列的准确性，都比 PRank 算法有提高。

3. 当排序模型 MAP 值达到并稳定同一正确率时，Active PRank 算法与 PRank 算法标注量的比较如表 5.4 所示。

表 5.4　达到同一正确率时，Active PRank 算法与 PRank 算法标注量比较

排序算法	MAP	标注量
PRank	0.35	350
Active PRank	0.35	200

使用 Active PRank 算法，当 MAP 值达到并稳定在 0.35 以上时，只需要标注约 200 条样本；而在同等条件下，使用 PRank 算法，需要标注约 350 条样本。可见，使用 Active PRank 算法可在保证排序模型性能的前提下，减少样本的标注量。

5.5.1.2　TREC .gov 数据集结果及分析

1．本实验使用的训练集中，包括 100 条已标注训练样本和大量的未标注样本。在每次迭代过程中，通过查询函数 Q 从未标注样本中选择 T=50 个样本，交由人工标注，并反复循环 N=10 次，共标注 600 条样本。共进行五组实验，对结果取平均值。图 5.6 是使用 PRank 算法随机标注样本和 Active PRank 算法选择性标注样本，排序结果随标注量增加，MAP 变化的对比图。

从图 5.6 可以看出，使用 PRank 算法和 Active PRank 算法进行排序，排序结果的正确率（MAP 值）随着标注量的增加均逐渐提高，且使用 Active PRank 算法选择性添加样本进行标注的结果正确率要优于使用 PRank 算法随机添加样本进行标注。

2．在同样标注 600 条样本的条件下，分别使用 PRank 算法随机选择样本标注与本文提出的 Active PRank 算法选择“最值得标注”的样本进行标注，使用相同的测试集进行对比实验。实验结果比较如图 5.7 所示。

从图 5.7 中可以看出，在同样标注 600 条样本的条件下，Active PRank 算法

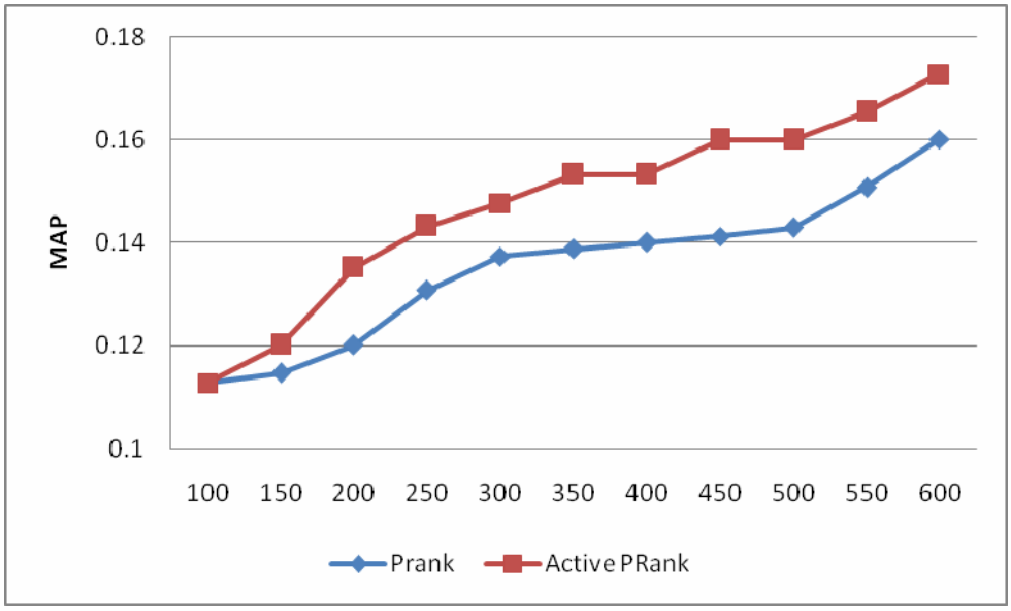

图 5.6　Active PRank 算法、PRank 算法随标注量增加结果变化图

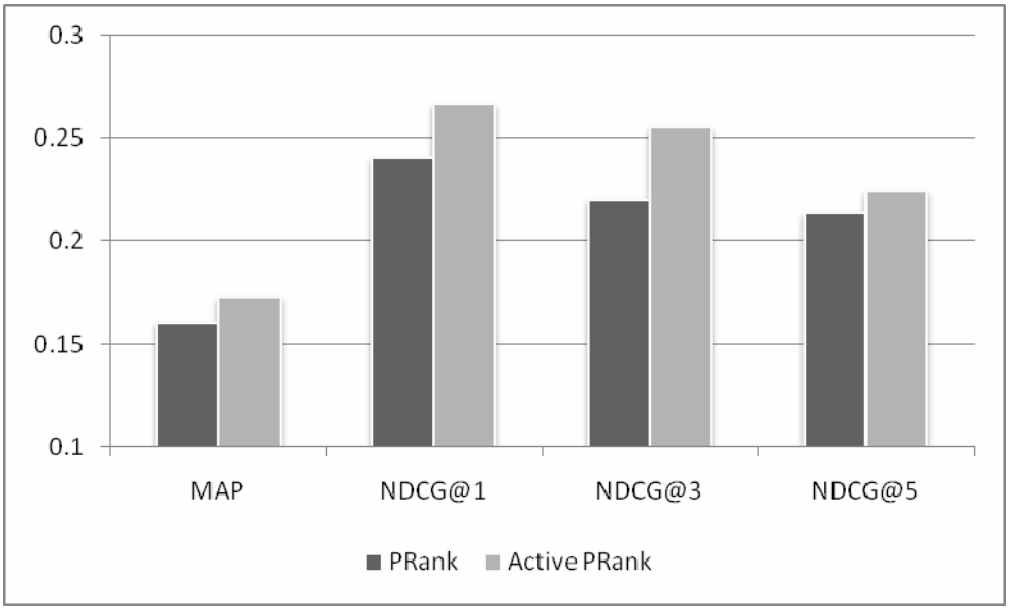

图 5.7　相同标注量条件下 Active PRank 算法与 PRank 算法结果比较图

的 MAP 值和 NDCG 值都高于 PRank 算法。可见，使用本文提出的 Active PRank 算法在同等标注量的条件下，无论是排序的整体效果，还是排序结果中顶部序列的准确性，都比 PRank 算法有提高。

3. 当排序模型 MAP 值达到并稳定同一正确率时，Active PRank 算法与 PRank 算法标注量的比较如表 5.5 所示。

表 5.5　达到同一正确率时，Active PRank 算法与 PRank 算法标注量比较

排序算法	MAP	标注量
PRank	0.14	450
Active PRank	0.14	250

使用 Active PRank 算法，当 MAP 值达到并稳定在 0.14 以上时，只需要标注约 250 条样本；而在同等条件下，使用 PRank 算法，需要标注约 450 条样本。可见，使用 Active PRank 算法可在保证排序模型性能的前提下，减少样本的标注量。

5.5.2　主动排序支持向量机算法实验结果及分析

本小节实验的目的是验证本文提出的主动排序支持向量机（Active RSVM）算法，使用排序支持向量机（RSVM）算法作为对比算法，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，验证本文提出的主动排序支持向量机算法的性能。

本小节的实验设计同 5.2.1 节，共分三组：1．Active RSVM 算法与 RSVM 算法随标注量增加 MAP 结果变化图；2．相同标注量条件下 Active RSVM 算法与 RSVM 算法 MAP 与 NDCG 结果比较；3．当排序模型 MAP 值达到同一正确率时，Active RSVM 算法与 RSVM 算法标注量的比较。

5.5.2.1　OHSUMED 数据集结果及分析

1．本实验使用的训练集中，包括 100 条已标注训练样本和大量的未标注样本。在每次迭代过程中，通过查询函数 Q 从未标注样本中选择 T=50 个样本，交由人工标注，并反复循环 N=10 次，共标注 600 条样本。共进行五组实验，对结果取平均值。图 5.8 是使用 RSVM 算法随机标注样本和 Active RSVM 算法选择性标注样本，排序结果随标注量增加，MAP 变化的对比图。

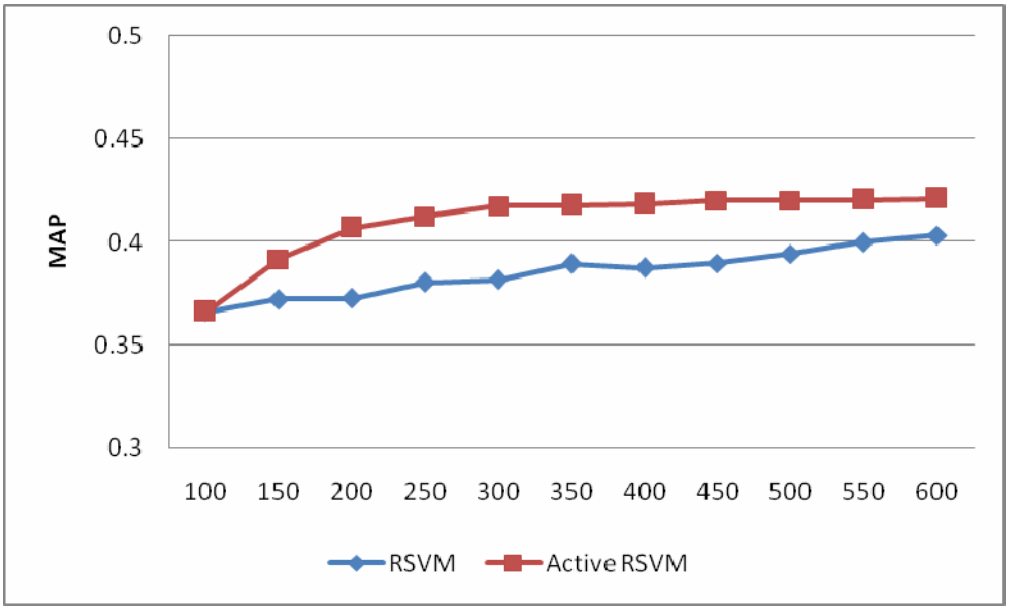

图 5.8　Active RSVM 算法、RSVM 算法随标注量增加结果变化图

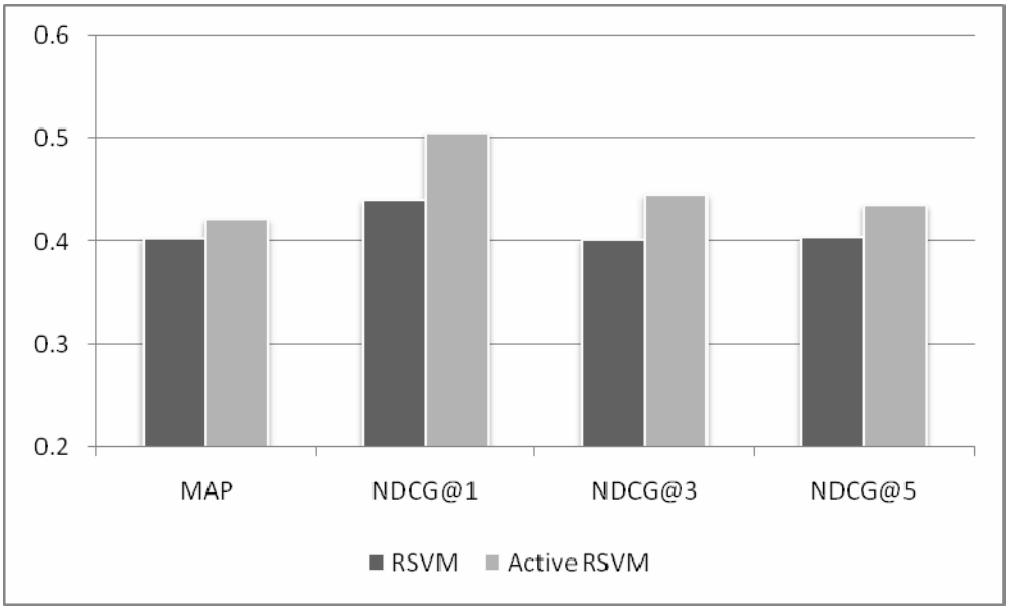

图 5.9　相同标注量条件下 Active RSVM 算法与 RSVM 算法结果比较图

从图 5.8 可以看出，使用 RSVM 算法和 Active RSVM 算法进行排序，排序结果的正确率（MAP 值）随着标注量的增加均逐渐提高，且使用 Active RSVM 算法选择性添加样本进行标注的结果正确率要优于使用 RSVM 算法随机添加样本进行标注。

2．在同样标注 600 条样本的条件下，分别使用 RSVM 算法随机选择样本标注与本文提出的 Active RSVM 算法选择“最值得标注”的样本进行标注，使用相同的测试集进行对比实验。实验结果比较如图 5.9 所示。

从图 5.9 中可以看出，在同样标注 600 条样本的条件下，Active RSVM 算法的 MAP 值和 NDCG 值都高于 RSVM 算法。可见，使用本文提出的 Active RSVM 算法在同等标注量的条件下，无论是排序的整体效果，还是排序结果中顶部序列的准确性，都比 RSVM 算法有提高。

3．当排序模型 MAP 值达到并稳定同一正确率时，Active RSVM 算法与 RSVM 算法标注量的比较如表 5.6 所示。

表 5.6　达到同一正确率时，Active RSVM 算法与 RSVM 算法标注量比较

排序算法	MAP	标注量
RSVM	0.40	550
Active RSVM	0.40	200

使用 Active RSVM 算法，当 MAP 值达到并稳定在 0.40 以上时，只需要标注约 200 条样本；而在同等条件下，使用 RSVM 算法，需要标注约 550 条样本。可见，使用 Active RSVM 算法可在保证排序模型性能的前提下，减少样本的标注量。

5.5.2.2　TREC .gov 数据集结果及分析

1．本实验使用的训练集中，包括 100 条已标注训练样本和大量的未标注样本。在每次迭代过程中，通过查询函数 Q 从未标注样本中选择 T=50 个样本，交由人工标注，并反复循环 N=10 次，共标注 600 条样本。共进行五组实验，对结果取平均值。图 5.10 是使用 RSVM 算法随机标注样本和 Active RSVM 算法选择性标注样本，排序结果随标注量增加，MAP 变化的对比图。

从图 5.10 可以看出，使用 RSVM 算法和 Active RSVM 算法进行排序，排序结果的正确率（MAP 值）随着标注量的增加均逐渐提高，且使用 Active RSVM 算法选择性添加样本进行标注的结果正确率要优于使用 RSVM 算法随机添加样

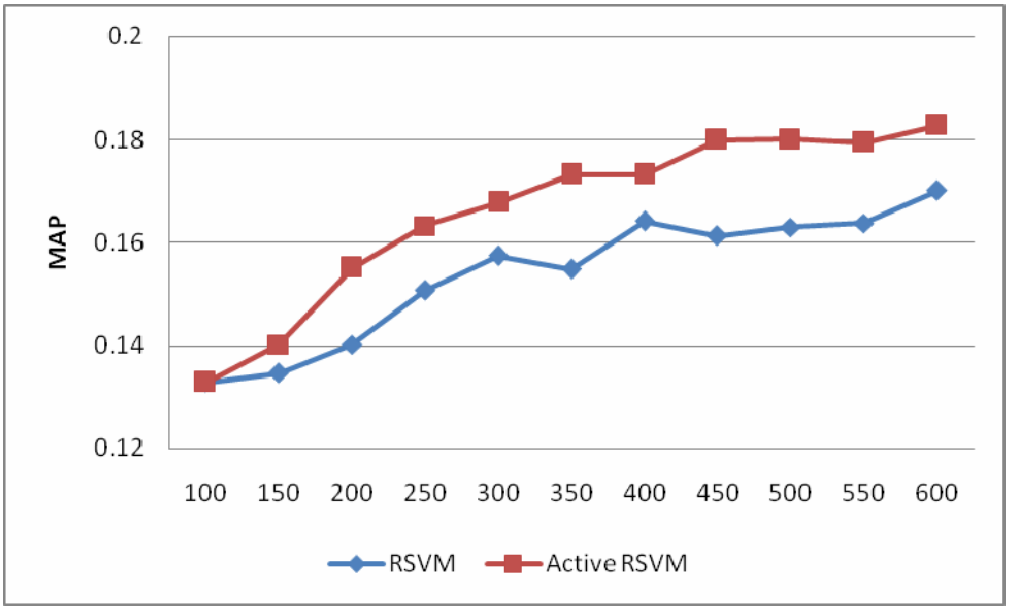

图 5.10　Active RSVM 算法、RSVM 算法随标注量增加结果变化图

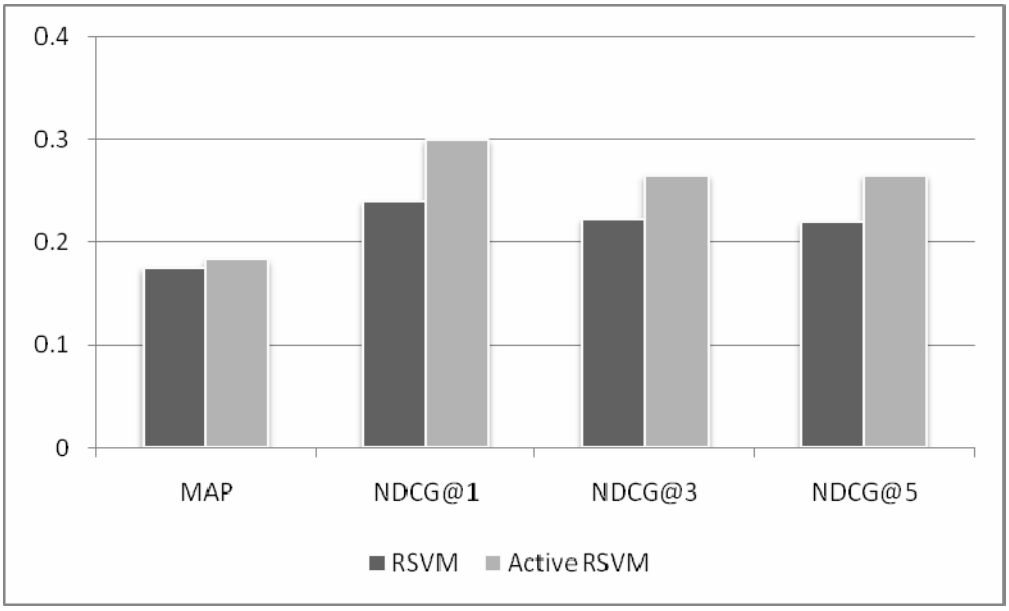

图 5.11　相同标注量条件下 Active RSVM 算法与 RSVM 算法结果比较图

本进行标注。

2．在同样标注 600 条样本的条件下，分别使用 RSVM 算法随机选择样本标注与本文提出的 Active RSVM 算法选择“最值得标注”的样本进行标注，使用相同的测试集进行对比实验。实验结果比较如图 5.11 所示。

从图 5.11 中可以看出，在同样标注 600 条样本的条件下，Active RSVM 算法的 MAP 值和 NDCG 值都高于 RSVM 算法。可见，使用本文提出的 Active RSVM 算法在同等标注量的条件下，无论是排序的整体效果，还是排序结果中顶部序列的准确性，都比 RSVM 算法有提高。

3．当排序模型 MAP 值达到并稳定同一正确率时，Active RSVM 算法与 RSVM 算法标注量的比较如表 5.7 所示。

表 5.7　达到同一正确率时，Active RSVM 算法与 RSVM 算法标注量比较

排序算法	MAP	标注量
RSVM	0.16	400
Active RSVM	0.16	250

使用 Active RSVM 算法，当 MAP 值达到并稳定在 0.16 以上时，只需要标注约 250 条样本；而在同等条件下，使用 RSVM 算法，需要标注约 400 条样本。可见，使用 Active RSVM 算法可在保证排序模型性能的前提下，减少样本的标注量。

通过在两个大规模真实数据集上的实验表明，使用本文提出的主动排序感知机（Active PRank）算法和主动排序支持向量机（Active RSVM）算法选择性标注样本比作为对比实验的排序感知机算法（PRank）和排序支持向量机（RSVM）算法取得更好的效果，即本文提出的主动排序学习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。

第六章　结束语

本文的研究领域是信息检索；本文关注的问题是信息检索中排序学习数据标注代价过大的问题；针对这一问题，本文提出主动排序学习方法，在保证排序模型性能的前提下降低标注代价。

6.1　本文工作总结

对信息检索中的排序学习应用进行分析，我们发现，目前主流的排序学习算法都是基于有监督学习的方法，需要大量的标注样本。然而，在实际的信息检索问题中，获取大量的标注样本是一项耗时长、难度大，而且代价昂贵的工作。

针对上述问题，本文把主动学习方法引入到排序学习中，由机器自动确定最值得标注的样本。本文提出了基于样本不确定程度的查询函数。使用查询函数，排序模型可以通过计算每个样本对应不同序标号的确定程度，自动找出最不确定的样本，作为“最值得标注”的样本，减少了样本标注量，从而降低了标注代价。

本文提出并实现基于数据点的主动排序感知机算法和基于有序对的主动排序支持向量机算法。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。本文取得的研究成果总结如下：

1、主动排序感知机（Active PRank）算法研究。

排序感知机（PRank）算法是基于数据点的排序学习的代表算法，具有运算复杂度低，算法实现简单等优点。本文将主动学习的思想引入到排序学习研究中，提出一种基于排序感知机的主动排序学习算法——Active PRank 算法。基于查询函数，选择出排序模型“最不确定”的样本作为“最值得标注”的样本，交由人工标注。使用主动排序感知机算法，可显著降低标注代价。

2、主动排序支持向量机（Active RSVM）算法研究。

排序支持向量机（Ranking SVM）算法是基于有序对的排序学习的代表算法。

排序支持向量机算法具有模型稳定，排序结果较好等优点。本文提出一种基于排序支持向量机的主动排序学习算法——Active RSVM 算法。基于本文提出的的查询函数，找出排序支持向量机算法中“最不确定”的样本作为“最值得标注”的样本，完成主动排序支持向量机算法的研究与实现。

3、主动排序学习中查询函数的研究。

在排序学习中应用主动学习的方法，由于标注单元比分类中更加复杂，判断那些样本是“最值得标注”变得更为困难。本文初步提出两种查询函数的设计方法——不确定性最大的和基于置信度的查询函数来查找出哪些样本是“最值得标注”的样本。

4、实验验证与结果分析。

信息检索是一个应用性很强的研究领域。因此，对本文提出的算法进行实验验证、分析是非常必要的。将本文提出的两种主动排序学习算法应用于文本检索与网页检索，通过基于大规模真实数据集合的实验，验证算法在现实大规模信息检索应用上的有效性。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。

6.2　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，应从以下两个方面着手，这包括：

1、提出以查询为标注单元（Query-Level）的主动排序学习算法。

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。

2、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] Freund, 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. 2001, 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.

致 谢

感谢我的导师黄亚楼教授，本论文是在黄老师的悉心指导下完成的。从论文的选题、实验设计到论文的构思及撰写无不凝聚着导师的智慧和心血。三年来，黄老师严谨的治学态度、渊博的学术知识、活跃的学术思维以及敏锐的科研洞察力都给我留下了深刻的印象，这将对我今后的工作学习生活产生积极而又深远的影响。

感谢信息检索项目组全体成员，谢茂强老师，李栋、倪维健、刘杰博士，蒯宇豪、刘金莉、郑楠等同学。非常怀念与大家一起讨论时的场景。论文中的很多想法是与大家一起碰撞后得到的；论文中的很多不足是与大家讨论后完善的。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。

在智能信息处理实验室学习生活的时光将是我人生珍贵的记忆。感谢实验室孙凤池、殷爱茹、师文轩、苑晶等老师对我的关心与指导，感谢实验室各位同仁，高远、陶通、孙杨、张展宇、张国恒、周立、史吏、黄佳等同学，以及各位师兄师姐、师弟师妹，非常荣幸能和你们一起走过研究生生活。

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！

王 扬

2008 年 4 月于南开园

个人简历

一、个人信息：

姓名：王　扬　　性别：男　　出生年月：1983 年 5 月

二、学习经历：

2005.09~2008.06 南开大学软件学院计算机软件与理论专业 攻读工学硕士学位

2001.09~2005.06 天津大学信息学院计算机科学与技术专业 工学学士学位

三、研究生期间参与的研究项目：

1、 项目名称：信息检索中基于损失函数优化的排序学习研究

项目来源：国家自然科学基金项目，编号 60673009

执行年限：2007.01~2009.12

参与工作：主动排序学习算法研究，项目实验平台搭建

2、 项目名称：Entity Search based on Text Mining

项目来源：微软亚洲研究院高校合作项目

执行年限：2006.04~2008.06

参与工作：相关算法研究及实验流程设计

3、 项目名称：邯郸钢铁新厂区实验管理系统

项目来源：邯郸钢铁集团横向合作项目

执行年限：2007.09~2008.05

参与工作：项目前期调研，需求分析，系统框架设计等

4、 项目名称：天津海事局海测大队“数字海事”试点单位建设规划

项目来源：天津海事局横向合作项目

执行年限：2007.09~2007.10

参与工作：撰写“数字海事”试点单位建设规划

四、研究生期间完成的论文：

[1]. **Yang Wang**, Yuhao Kuai, Yalou Huang, Dong Li and Weijian Ni. Confidence-based Active Ranking for Document Retrieval. Accepted by the 7th International Conference on Machine Learning and Cybernetics. 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.

[3]. Maoqiang Xie, Jinli Liu, Nan Zheng, Dong Li, Yalou Huang and **Yang Wang**. Semi-Supervised Graph-Ranking for Text Retrieval. In Proceedings of the Fourth Asia Information Retrieval Symposium. Harbin, China. Jan, 2008.

[4]. Weijian Ni, Yalou Huang, Dong Li and **Yang Wang**. Boosting over Groups and Its Application to Acronym-Expansion Extraction. Accepted by the 4th International Conference on Advanced Data Mining and Applications. Chengdu, China. July, 2008.

[5]. Dong Li, Weijian Ni, Maoqiang Xie, Yalou Huang and **Yang Wang**. An Ensemble Approach to Learning to Rank. Accepted by the 5th International Conference on Fuzzy Systems and Knowledge Discovery. Jinan, China. Oct, 2008.

[6]. Dong Li, Weijian Ni, Maoqiang Xie, Yalou Huang and **Yang Wang**. Multiple Ranker Method in Document Retrieval. Submitted to the 2008 International Conference on Intelligent Computing. Shanghai, China. Sep, 2008.

五、研究生期间获得的奖励与荣誉：

1. 2007 年 10 月 2006-2007 年度天津市三好学生
2. 2007 年 12 月 2007 年度南开大学三好学生
3. 2007 年 12 月 2007 年度南开大学研究生奖学金
4. 2008 年 05 月 2008 年度南开大学优秀共青团员