**学习排序用于信息检索和自然语言处理**

**Hang Li**

**摘要**

学习排序是指在排序任务中训练模型的机器学习技术。学习排序对于信息检索、自然语言处理和数据挖掘中的许多应用都很有用。最近，人们对这个问题进行了深入的研究，并取得了重大进展。本讲座将介绍该领域的基本问题、现有方法、理论、应用和未来的工作。

作者首先展示了信息检索和自然语言处理中的各种排名问题可以形式化为两个基本的排名任务，即排名创建(或简单的排名)和排名聚合。在创建排名时，给定一个请求，希望根据从请求和产品中派生的特性生成产品排名列表。在排名聚合中，给定一个请求以及一系列产品的排名列表，需要生成新的产品排名列表。

排名创造(或排名)是学习排名的主要问题。它通常被形式化为一个有监督的学习任务。作者详细说明了排名创建和排名聚合的学习方法，包括训练和测试、评估、特征创建以及主要方法。已经提出了许多创建排名的方法。根据所使用的损失函数，这些方法可以分为逐点方法、两两方法和列表方法。它们也可以根据所采用的技术进行分类，如基于支持向量机的方法、Boosting支持向量机方法、基于神经网络的方法。

并详细介绍了一些流行的学习排序方法。其中包括恶作剧、OC支持向量机、排名支持向量机、IR支持向量机、GBRank、RankNet、LambdaRank、ListNet & ListMLE、AdaRank、SVM MAP、SoftRank、Borda Count、Markov Chain和CRanking。

作者介绍了学习排序的几个应用实例，包括网络搜索、协同过滤、定义搜索、关键字提取、依赖查询的摘要和机器翻译中的重排序。

在统计学习框架中给出了排序创建的学习公式。并对学习排名的未来研究方向进行了探讨。

关键字：学习排名，排名，排名创建，排名聚合，信息检索，自然语言处理，监督学习，网络搜索，协同过滤，机器翻译。

**前言**

这本书介绍了一个关于学习排名的调查，并详细描述了学习排名的方法。这本书的主要重点是监督学习排名创建。

本书针对信息检索、自然语言处理、机器学习、数据挖掘和其他相关领域的研究人员和实践者。它假设读者具备统计学和机器学习的基本知识。

第一章给出了学习排名的正式定义。第2章描述了创建排名的学习，第3章描述了排名聚合的学习。第4章详细解释了最先进的学习排序方法。第五章介绍了学习在排名中的应用。第六章介绍了排名学习理论。第七章介绍了目前和未来关于学习排名的研究。

在此，我谨向我的同事刘铁岩、徐军、秦涛、曹云波、胡云华表示衷心的感谢。我们一直在一起学习排名。非常感谢我们的实习生曹哲、蔡明峰、耿秀波、兰艳艳、夏芬、李明、江鑫、陈伟也参与了相关的研究。

我非常感谢马维英、韩晓文和沈国强对我的鼓励和指导。我还要感谢我们在微软的内部合作者Chris Burges、Stephen Robertson、Michael Taylor、John Guiver、Dmitriy Meyerzon、Victor Poznanski、Rangan Majumder、Steven Yao，以及我们的外部合作者Rong Jin、Zhi-Hua Zhou、Hamed Valizadegan、Cheng Xiang Zhai和Thorsten Joachims。

我非常感谢徐君和陶琴为我的书提供了一些材料。

非常感谢两位匿名评论者，陈旺、魏武和魏晨，他们阅读了本书的草稿，并提出了许多有价值的意见。

我还要感谢Graeme Hirst和Michael Morgan。没有他们的支持，这本书就不会出版。

**学习排名**

**1.1 等级**

在信息检索和自然语言处理中有许多任务，其核心问题是排序。其中包括文档检索、实体搜索、问答、元搜索、个性化搜索、在线广告、协同过滤、文档摘要和机器翻译。

在我们看来，基本上有两种类型的排名问题:排名创建(或简单的排名)和排名聚合。排名创建是使用对象的特性创建对象的排名列表，而排名聚合是使用对象的多个排名列表创建对象的排名列表，这将在本章后面正式描述。

文档检索、协同过滤、机器翻译中的重新排名都是排名创建的例子，而元搜索则是排名聚合的例子。

**文档检索**

文档检索包括网络搜索、企业搜索、桌面搜索等。虽然有局限性，但它仍然是人们访问存在于网络和计算机上的大量信息的最实际的方式。例如，根据IProspect 2的一份报告，56%的互联网用户每天都使用网络搜索，88%的互联网用户每周都使用网络搜索。

文档检索可以描述为以下任务(参见图1.1)，其中排名起着关键作用。检索系统维护一个文档集合。给定来自用户的查询，系统从集合中检索包含查询词的文档，对文档进行排序，并向用户显示排名最高的文档列表(例如，1000个文档)。排序主要是根据文档与查询的相关性来执行的。

**协作费尔增长率**

协同过滤是计算机系统在电子商务、网络广告等领域向用户进行推荐的最基本模型。例如，如果用户对数据库中某些电影的偏好是已知的，那么我们可以使用协作过滤向用户推荐他们可能没有看过且可能感兴趣的电影。

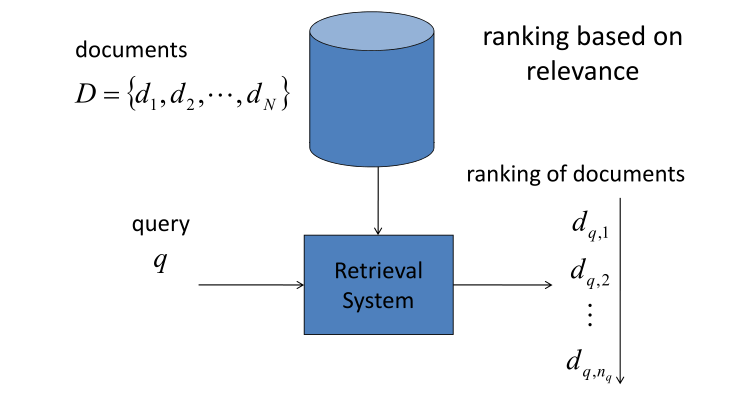


图1.1:文档检索。向下箭头表示文档的等级

协同过滤中的数据以矩阵形式给出，其中行对应于用户，列对应于项目(参见图1.2)。矩阵中的一些元素是已知的，而其他的则是未知的。元素表示用户对具有多个等级(级别)的物品的评级。问题是如何确定矩阵的未知元素。一个常见的假设是，相似的用户可能对相似的项目有相似的评级。当指定用户时，系统会给出一个项目的排序列表，排序的项目越高，排名越靠前。

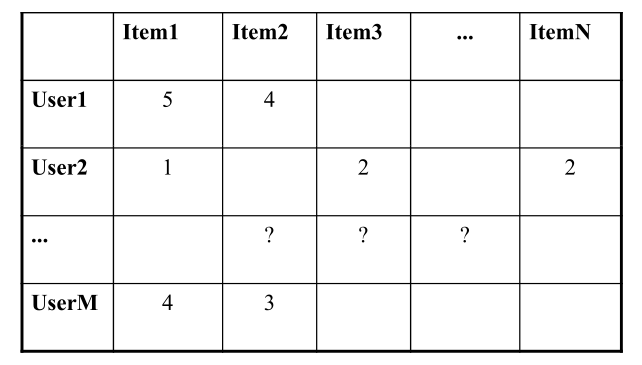


图1.2:协同过滤

**机器翻译**

机器翻译可以帮助人们跨语言获取信息，因此非常重要。给定源语言中的一个句子，通常目标语言中有大量可能的翻译(句子)。然而，翻译的质量可能有所不同。如何选择最合理的翻译是关键问题。

一种流行的机器翻译方法包括两个阶段:候选的生成和重新排序(见图1.3)。给定源语言中的一个句子，系统首先生成并排序。

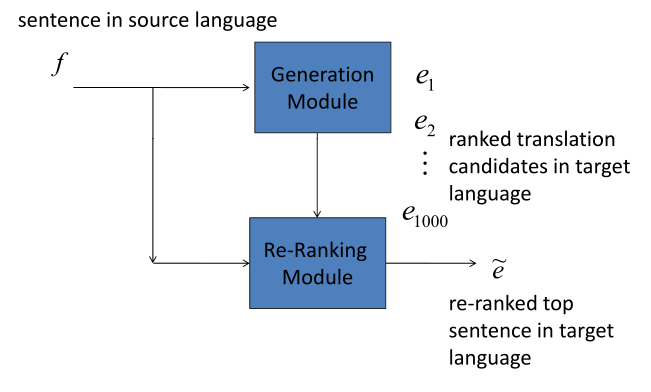


图1.3:机器翻译

使用生成模型对目标语言中所有可能的候选翻译进行生成，然后使用判别模型对排名靠前的候选翻译(比如，1000个候选翻译)进行重新排序，最后选择排名靠前的候选翻译作为输出。重新排序的过程是根据候选对象的翻译质量的可能性进行的，这对机器翻译的性能至关重要。

**元搜索**

元搜索系统是将用户请求发送到多个搜索系统，并从这些搜索系统聚合结果的系统。随着网络的不断发展，越来越多的搜索系统(有时在不同的领域)变得可用，元搜索变得越来越重要。更正式的说法是，在元搜索中，查询被提交给几个搜索系统，然后从这些系统返回文档的排名列表。然后，元搜索系统将所有排名列表组合在一起，生成一个新的排名列表(元排名列表)，这比所有单个排名列表要好。在实践中，不同系统返回的文档集可能不同。

我们可以将这些文档集的并集作为最终的文档集。图1.4说明了元搜索的过程。

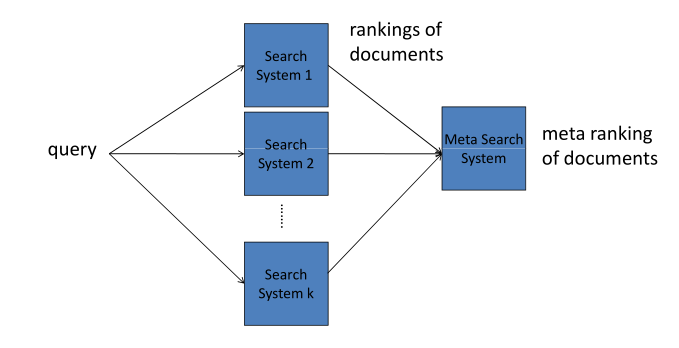


图1.4:元搜索

**1.2学习排名**

最近，在机器学习、信息检索和自然语言处理的交叉领域出现了一个新的领域，称为排名学习。学习排名就是使用机器学习技术进行排名。它是基于之前在机器学习和统计方面对排名的研究，也有自己的特点。

学习排名可能有两种定义。从广义上讲，学习排名是指任何用于排名的机器学习技术。狭义的排名学习是指在上面描述的排名创建和排名聚合中建立排名模型的机器学习技术。本书采用后一种(狭义的)定义。图1.5给出了学习排名过程中问题的分类。

近年来，在学习排名技术的研究和开发方面做出了重大努力。许多强大的方法已经被开发出来，其中一些已经成功地应用到实际应用中，如网络搜索。已发表论文百余篇。基准数据集已发布(如[70])，并对任务3进行了竞赛。举办了讲习班(如[58,64,65])和杂志特刊(如[69])。关于这一主题的一本书也已出版[68]。

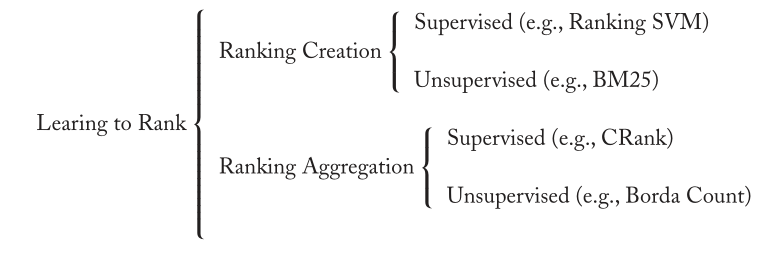


图1.5:学习排序的问题分类

**1.3创建排名**

我们可以将已经描述的排名创建问题概括为一个更一般的任务。假设有两个集合。为简单起见，我们将它们称为请求集合Q = {q1, q2，···，qi，···，qM}和提供(或对象)集合O = {o1, o2，···，oj，···，oN}，分别为4。在文档检索、协同过滤和机器翻译中，Q可以分别是一组查询、一组用户和一组源句子。O可以分别是一组文档、一组项目和一组目标句。注意Q和O可以是无限集。给定q的一个元素q和O的一个子集O (O∈2O)，我们将根据q和O的信息对O中的元素进行排序。排名(rank creation)使用rank (score)函数F (q, O): q × on→$n执行 。



其中，n = |O|， q是q的一个元素，O是O的一个子集，SO是O中元素得分的集合，π是被SO排序的O中元素的排序列表(排列)。注意，即使对于相同的O, F也可以给出两个不同的排名列表和两个不同的q。也就是说,我们关心排名啊,对一个特定的q。

而不是使用F (q, O),我们通常用排名(或得分)函数F (q, O)为便于操作、其中q是Q的一个元素，,O是一个元素的O, O。分排名函数F (q,o)给o中的每个o赋一个分数，然后使用分数对o中的元素进行排序。这意味着排序实际上是按照f (q, o): q × o→$来执行的



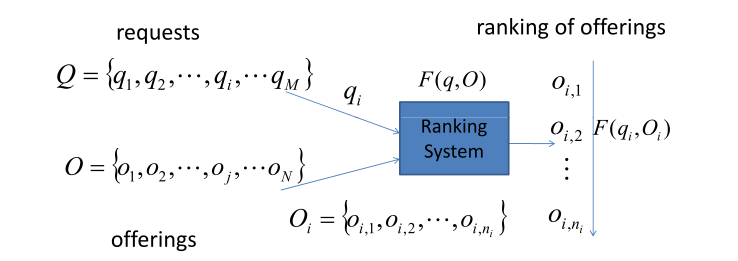


图1.6:排名创建(带有全局排名功能)向下箭头表示对象的排序

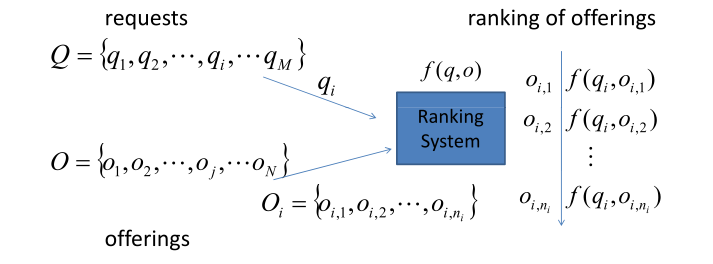


图1.7:排名创建(带有本地排名功能)向下箭头表示对象的排序

我们称F (q, O)为全局排序函数，称F (q, O)为局部排序函数，因为前者用于对象的子集，而后者用于单个对象(参见图1.6和1.7)。

**1.4排名聚合**

我们还可以定义一般的排序聚合任务。再次假设Q = {q1, q2，···，qi，···，qM}和O = {o1, o2，···，oj，···，oN}分别是一组请求和一组提供。阿q q的一个元素和一个子集的O, O有k排名列表:“={我|ππ∈#,i = 1 , · · · , k},其中#是所有o .的榜上排名聚合需要请求q和排名列表的产品”作为输入,并生成一个新的产品排名π与排名作为输出函数F (q,Σ):Q×Πk→$n



我们通常简单地定义



也就是说，我们假设排序函数不依赖于请求。

排名聚合实际上是将多个排名列表组合成一个排名列表的过程，这比任何一个原始排名列表都要好，如图1.8所示。排名模式是一个全球排名模式。

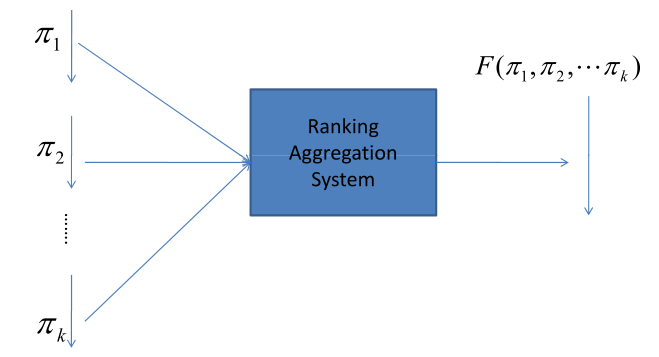


图1.8:排名聚合。向下的箭头表示对象的排名

排名创建根据请求和产品的特性生成排名，而排名聚合则根据产品的排名生成排名。注意，排名创建的输出可以用作排名聚合的输入。

**1.5学习创建排名**

当提到学习排名时，它通常意味着使用监督学习创建排名。这也是本书的主要关注点。学习任务可以这样描述。有两个系统:学习系统和排名系统。

学习系统以训练数据作为输入。训练数据由请求及其相关的产品排名列表组成。为每个请求qi∈{q1、q2 , · · · , qm},有一组相关联的产品Oi∈{O1、O2 , · · · , Om} (Oi = {Oi 1 Oi, 2 , · · · , oi,ni},i= 1 , · · · , m),有一个“真正的”排名的产品π∈{π1,π2π , · · · , πm}。学习系统在训练数据的基础上构建一个排名模型(通常是一个局部排名模型f (q, o))。

然后，排名系统利用学习到的排名模型进行排名预测。给定一个新的请求qm+1，排名系统会收到一个子集的产品Om+1，分配数分使用排序模型对产品进行排序，并对产品的得分进行降序排序，得到排序列表πm+1。见图1.9。

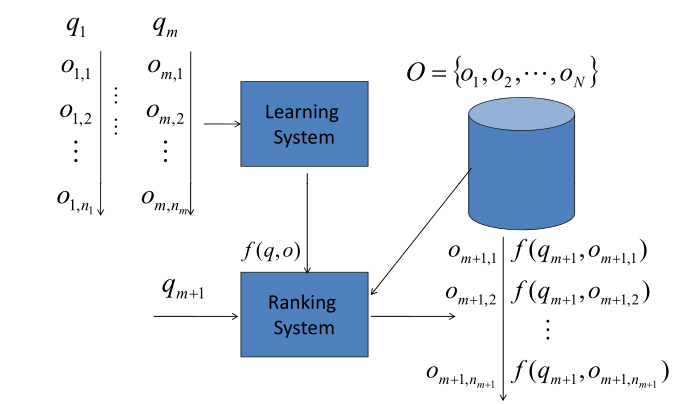


图1.9:学习创造排名向下箭头表示排名

以下是排名创造学习的主要特点。

•创造排名:生成一个排名基于请求和祭的祭

•特点:使用功能上定义的请求和祭

•当地排名模型:当地排名模式f (q, o)是利用

•监督学习:排名模型通常是由监督学习

**1.6学习排名聚合**

排序聚合可以是有监督的，也可以是无监督的。在监督学习设置中，学习系统以训练数据作为输入。训练数据由请求及其相关的产品排名列表组成。为每个请求qi∈{q1、q2 , · · · , qm},有一组相关联的产品Oi∈{O1、O2 , · · · , Om} Oi = {Oi 1 Oi, 2 , · · · , oi,ni},i= 1 , · · · , m。此外,对于每个Oi, k的一组的榜上:Σi={πi,1,π,2 , · · · , π,k},以及一套“真正的排名:π。学习系统利用训练数据构建了一个排名模型F (q，Σ)。

然后，排名系统利用学习到的排名模型进行排名预测。给定一个新的请求qm+1，排序系统接收到相关的供给集Om+1上的k个排序列表:Σm+1 = {πm+1,1， πm+1,2， πm+1,k}，对具有的供给赋分建立排序模型，并按照分数降序排列，得到排序表πm+1。

以下是排名聚合学习的主要特点。

•排名聚合:生成一个排名的产品来自多个排名列表产品

•排名:中使用多个产品的排名列表

•全球排名模型:全球排名模式F (q,Σ)是利用

•监督或无监督学习:排名模型是由监督或无监督学习

**学习创造排名**

本章对排名创建的学习进行了一般性的介绍。排名创建的目的是根据产品和请求的特性创建产品的排名列表，这样请求的“好的”产品就会排在最前面。创建排名的学习涉及到使用机器学习技术自动构建排名模型。

由于排名创建在实践中的重要性，近年来对排名创建的学习进行了深入的研究。人们提出了许多方法，其中一些技术已成功应用于网络搜索等应用。

接下来，我们以文档检索(或搜索)为例进行说明。不失一般性，这里描述的技术可以应用于其他应用程序。

**2.1文档检索为例**

学习排序创建(一般学习排序)在文档检索中起着非常重要的作用。传统上，文档检索的排序模型f (q, d)没有经过训练，其中q表示查询，d表示文档。在BM25[90]中，排名模型f (q, d)表示为条件概率分布P (r|q, d)，其中r取1或0为值，表示相关或不敬，q和d分别表示查询和文档。在语言模型(Language Model for IR, LMIR)[80,113]中，排名模型被定义为一个条件概率分布P (q|d)，其中q表示查询，d表示文档。BM25和LMIR都是根据给定的查询和文档计算的，因此不需要训练(只需要调整几个参数)。

近年来，文献检索领域出现了一种新的趋势，即利用机器学习技术自动构建文献检索的排序模型f (q, d) (cf.，[39])。这是由许多事实所推动的。在文档检索中，特别是在网络搜索中，有许多信号可以表示相关性。将这些信息整合到排名模型中，并自动构建排名模型成为一种自然的选择。在网络搜索引擎中，积累了大量的搜索日志数据，如点击数据。这也带来了一个新的机会，通过从搜索日志中提取训练数据，以低成本自动创建排名模型。这些都促进了对排名学习的研究。事实上，学会排名已经成为现代网络搜索的关键技术之一。

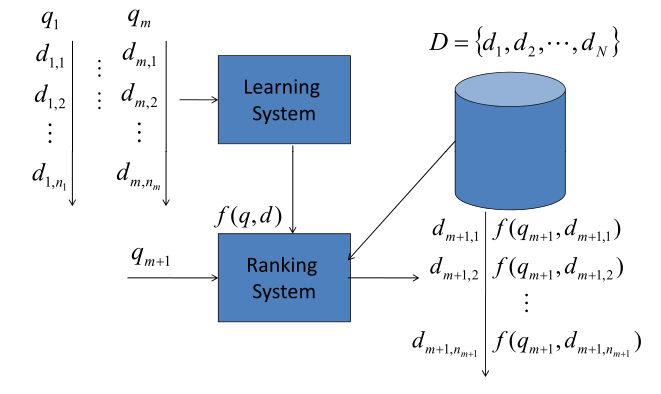


图2.1:学习文档检索的排序

**2.2学习任务**

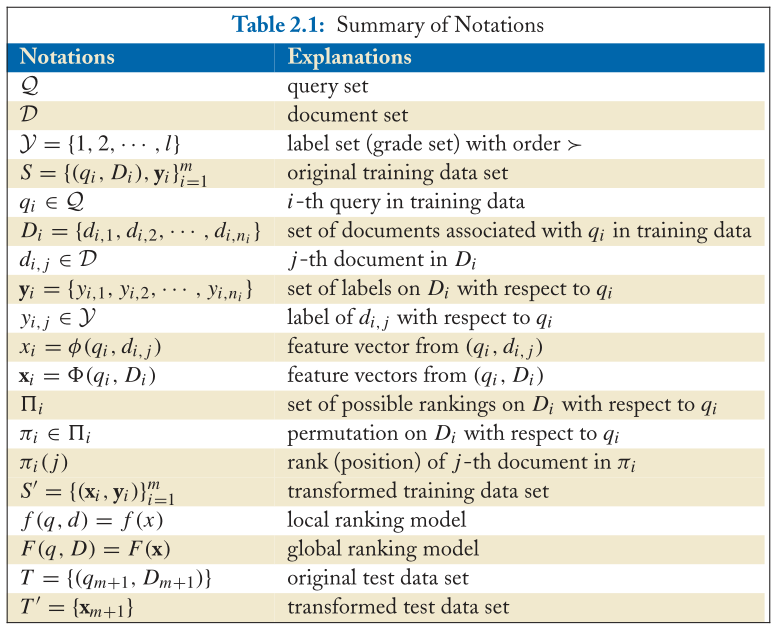
我们以文档检索为例，描述了在学习创建排名时的一些问题。这包括培训和测试过程、培训数据创建、特征构建和评估。我们还讨论了排序与其他任务(如有序分类)之间的关系。

**2.2.1培训和测试**

创建排名的学习包括训练和测试，是一种有监督的学习任务。

训练数据包含查询和文档。每个查询都与许多文档相关联。文中还给出了与查询相关的文档。如后面所述，相关信息可以通过几种方式给出。这里，我们采用最广泛使用的方法，并假设文档与查询的相关性是由标签表示的。这些标签有几个等级(水平)。文档的等级越高，说明文档的相关性越强。

假设Q是查询集，D是文档集。设Y ={1,2，···，l}为标签集，其中标签表示等级。在等级l % 1 ~ 1% %···% 1之间存在一个总顺序，其中%表示顺序关系。进一步假设{q1, q2，···，qm}是训练查询的集合，qi是第i个查询。Di = {Di,1, Di,2，···，Di,ni}是与查询qi相关的文档集合，yi = {yi,1, yi,2，···，yi,ni}是与查询qi相关的标签集合，其中ni表示Di和yi的大小;di,j表示di中的第j个文档;yi,j∈Y表示yi中的第j个等级标号，表示di,j对于qi的关联度。原始训练集记为S = %(qi, Di)， yi&m i=1。



从每个查询文档对(qi, di,j)， i = 1,2，···，m创建一个特征向量xi,j = φ(qi, di,j);J = 1,2，···，ni，其中φ为特征函数。也就是说，特性被定义为查询函数和文档函数。设xi = {xi,1, xi,2，···，xi,ni}，我们将训练数据集表示为S ' = %(xi, yi)&m i=1。我们的目标是培养本地排名模型(q, d) = f (x),可以指定一个分数给定查询文档对q和d,或等价为一个给定的特征向量x。更一般的意义上,我们也可以考虑一个全球排名模型(q, d) = f (x)注意当地排名模型输出一个分数,而全球排名模型输出的是一组分数。

设Di中的文档由整数{1,2，···，ni}标识。我们定义排列(排序表)πi在Di上为从{1,2，···，ni}到它自身的双射。我们用#i表示Di的所有可能排列的集合，用πi(j)表示第j个文档的排列πi的秩(或位置)，用π - 1(j)表示第j个文档的排列πi。排序就是使用排序模型F (qi, Di)(或F (qi, Di))给出的分数，为给定查询qi和相关文档集合Di选择一个排列πi∈#i。

测试数据由一个新的查询qm+1和相关的文档集Dm+1组成。T = {(qm+1, Dm+1)}。我们创建特征向量xm+1，利用训练好的排序模型对Dm+1中的文档进行评分，并根据评分对文档进行排序，得到文档的排序列表作为输出πm+1。

表2.1给出了符号的摘要。图2.2和2.3说明了培训和测试过程。

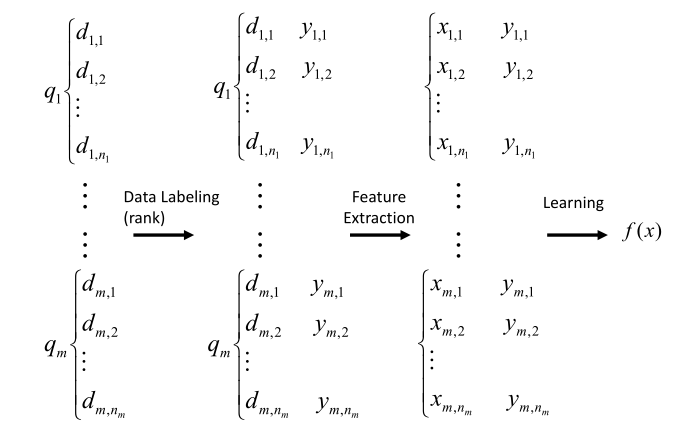


图2.2:培训过程



图2.3:测试过程

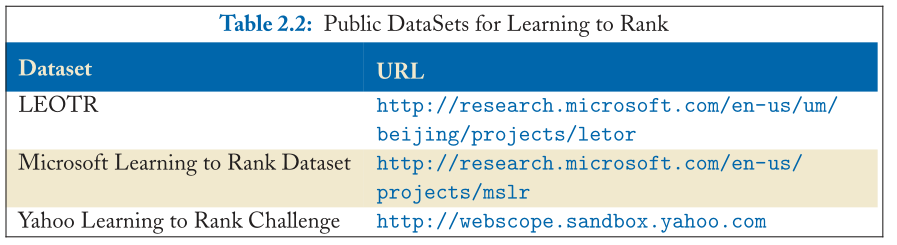
训练和测试数据与分类、回归等传统监督学习中的数据相似，但又有所不同。查询及其关联的文档组成一个组。组是i.i.d.数据，而组中的实例不是i.i.d.数据。(局部)排序模型是查询和文档的函数，或等价于查询和文档的特征向量的函数。

**2.2.2训练数据创建**

创建排名的学习是一项有监督的学习任务，因此如何创建高质量的训练数据是一个关键问题。理想情况下，训练数据应该包含每个查询的完美的文档排名列表。

然而，在实践中，这种数据可能很难获得，因为排名列表必须反映用户对与查询有关的文档相关性的平均判断。目前，有两种常见的方法来创建训练数据。第一个是人类标签，它在红外领域被广泛使用。首先，从搜索系统的查询日志中随机选择一组查询。假设有多个搜索系统。然后将查询提交给搜索系统，并收集所有排名靠前的文档。因此，每个查询都与来自多个搜索系统的文档相关联(这称为池策略)。然后要求人工判断人员对所有查询文档对进行相关性判断。相关性判断一般分为完美(perfect)、优秀(excellent)、良好(good)、公平(fair)、不好(bad)五个层次。人类的判断是从普通用户的角度进行关联判断的。例如，如果查询是“Microsoft”，而网页是microsoft.com，那么标签就是“perfect”。此外，维基百科上关于微软的页面是“优秀的”。把微软作为主要话题的页面会被标记为“好”，只提到微软的页面会被标记为“不错”，与微软无关的页面会被标记为“不好”。然后将表示相关性的标签分配给查询文档对。对查询文档对进行标注可以由多名法官进行，然后再进行多数投票。由于人工标注的代价很高，因此一些查询和文档对通常只由一个判断者进行判断。因此，如何提高人类相关性判断的质量成为学习排序研究的一个重要问题。

另一种生成训练数据的方法是从点击数据派生出来的。网络搜索引擎的点击数据记录了用户提交查询后对文档的点击。Clickthrough数据代表了用户对相关性的隐式反馈，因此对于相关性判断非常有用。一种方法是使用文档单击次数之间的差异来派生文档对[57]的首选项(相对相关性)。假设对于一个查询，在前1、2、3个位置返回了三个文档a、B、C，并且记录了用户对文档的总点击次数。如果文档B比文档A有更多的点击，那么我们可以确定文档B比文档A对于这个查询更相关，因为用户似乎更喜欢文档B而不是文档A，即使文档A的排名低于文档B。给定一个文档的排序列表，用户倾向于点击顶部的文档，甚至这些文档可能并不相关。因此，顶部的文档往往有更多的点击。这种现象被称为“点击偏见”。使用上述方法，我们可以有效地处理点击偏好，因为它通过查看排名较高的文档的“跳转”来获得文档的偏好对。因此，该方法可以生成文档的偏好对作为学习排序的训练数据。在每个文档对中，就查询而言，一个文档被认为比另一个文档更相关。也见[87,88]。



以上两种方法各有利弊。当数据是由人创建时，很难保持数据的质量。人类的判断容易出错，而且他们对相关性的理解也有局限性，因为他们不是查询所有者。此外，手动数据标注的成本也很高。而通过点击数据推导训练数据的成本较低，数据也可以代表真实用户的判断。这种方法的缺点是，点击率数据是有噪声的，并且只能用于头查询(高频查询)。

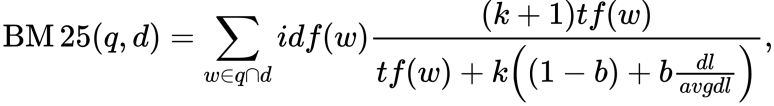
表2.2给出了一个公共数据集的列表，用于学习排名研究。它们都是由第一种方法创建的数据集。

**2.2.3功能建设**

排名模型f (q, d)实际上定义为f (x)，其中x是基于q和d的特征向量，也就是说，排名模型是基于特征的。这就是排名模型具有泛化能力的原因。具体来说，它是从少量查询及其关联文档训练而来的，但适用于任何其他查询及其关联文档。在其他机器学习任务中，学习的性能高度依赖于所使用特征的有效性。因此，如何定义有用的特征是一个极其重要的问题。

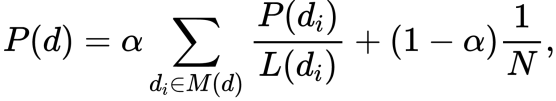
在网络搜索中，BM25和PageRank是广泛使用的排名功能。事实上，两者都可以被视为无监督排名模型。在网络搜索的早期阶段，最终的排名模型通常被简单地定义为BM25和PageRank的线性组合，或者类似的东西。后来，开发出了越来越多的功能。这也是为什么在构建排名模型时，需要一种更普遍、更有原则的学习方法。给出了BM25和PageRank的定义。

BM 25是一个表示文档d与查询q相关性的概率模型[90]。它实际上查看查询词和文档词之间的匹配程度，并利用查询词在文档中的出现次数来表示相关性。具体来说，查询q和文档d的BM25计算为



其中w表示d和q中的一个单词，tf (w)表示w在d中的频率，idf (w)表示w的逆文档频率，dl表示d的长度，avgdl表示文档的平均长度，k和b为参数。

Page Rank表示网页的重要性[78]。它将网络看作是一个有向图，其中页面是顶点，超链接是有向边，在网络图上定义了一个马尔可夫过程，并将马尔可夫过程的平稳分布(页面排名)看作是页面重要性的分数。网页d的网页排名定义为P (d)



P (d)的概率是访问页面d P (di)的概率是访问页面di, M (d)是一组页面链接到d, L (di)的outlinks di, N是图上的节点的总数,和α是一个重量。

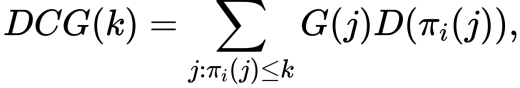
在网络搜索中还有其他的功能。表2.3给出了一些例子，这些例子已经被证实在网络搜索中是有效的。它们包括查询-文档匹配特征和文档特征，分别表示文档与查询的相关性和文档的重要性。

Web页面通常包含许多字段(元数据流)，例如标题、锚文本、URL、提取的标题[50,51]，以及点击数据[3]中的相关查询。可以为web页面的每个字段定义querydocument匹配特性(例如，BM25)，从而利用相同类型的许多特性。

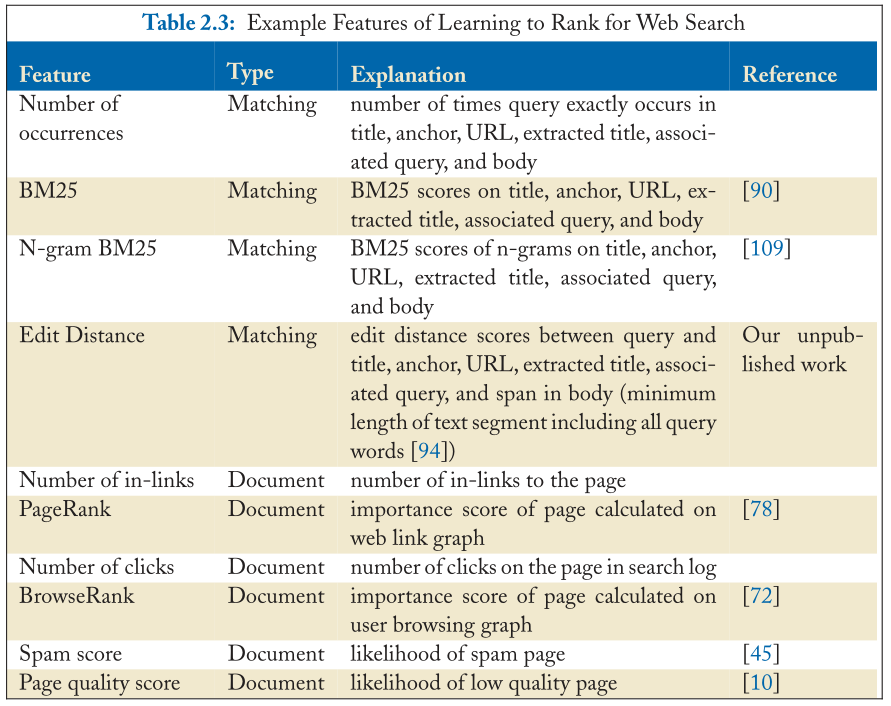
**2.2.4评价**

对排名模型的性能评价是通过比较该模型输出的排名列表和作为ground truth给出的排名列表来进行的。几种评价方法在红外光谱等领域得到了广泛的应用。这些包括NDCG(归一化累计折现增益)、DCG(累计折现增益)[53]、MAP(平均平均精度)[101]、WT A(赢家全取)、MRR(平均互惠秩)和Kendall 's Tau。

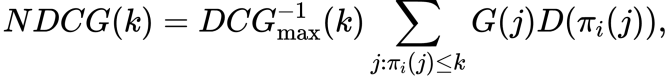
给定查询qi和关联文档Di，设πi是Di的排序表(排列)，yi是Di的标签(等级)的集合。DCG用标签来衡量排名的优劣。具体来说，定义qi在k位的DCG:



其中G(·)为增益函数，D(·)为头寸贴现函数。注意，πi(j)表示di,j在πi中的位置。因此求和取排序表πi1的前k位。DCG表示从位置1到位置k访问信息的累积增益



在头寸上有折扣。NDCG为归一化DCG，定义qi的k位NDCG:

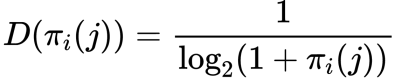


其中DCGmax(k)是归一化因子，并被选中使位置k处的π \* i的NDCG分数为1。在完美排序中，文件的等级越高，文件的等级越高。注意，对于一个查询和相关的文档，可以有多个完美的排名。

增益函数通常定义为等级的指数函数。也就是说，当关联度增加时，用户对信息的访问满意度呈指数增长。例如,

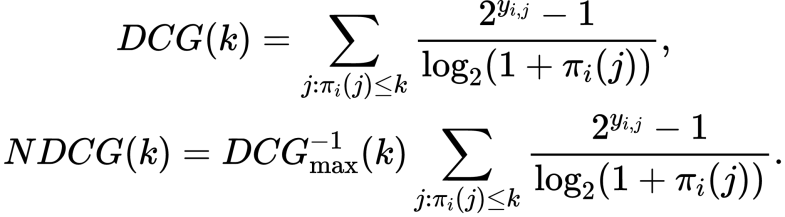


式中，yi,j为排序表πi中di,j的标号(grade)。贴现函数通常定义为头寸的对数函数。也就是说，以对数的方式获取信息的满意度当获取信息的位置增加时减少。

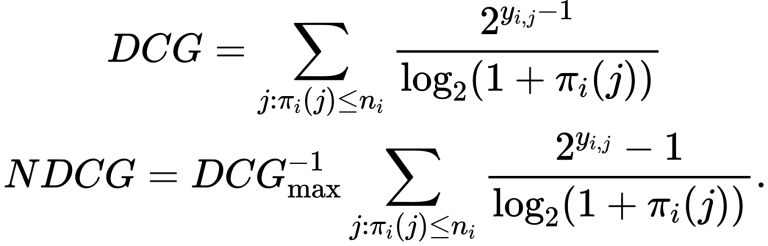


其中πi(j)为di,j在排序表πi中的位置。

因此，在气的位置k处的DCG和NDCG就变成了



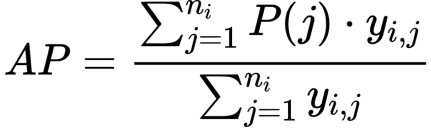
DCG和NDCG的整体排名为齐成为榜单



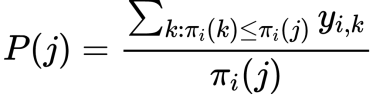
DCG和NDCG值在查询(i = 1，···，m)上进一步平均。

表2.4给出了两个排名表的NDCG值计算示例。NDCG (DCG)具有对相关文件排名高的排名表给予高分的效果。见表2.4中的例子。对于完美排名，每个位置的NDCG值都是1，而对于不完美排名，NDCG值都小于1。

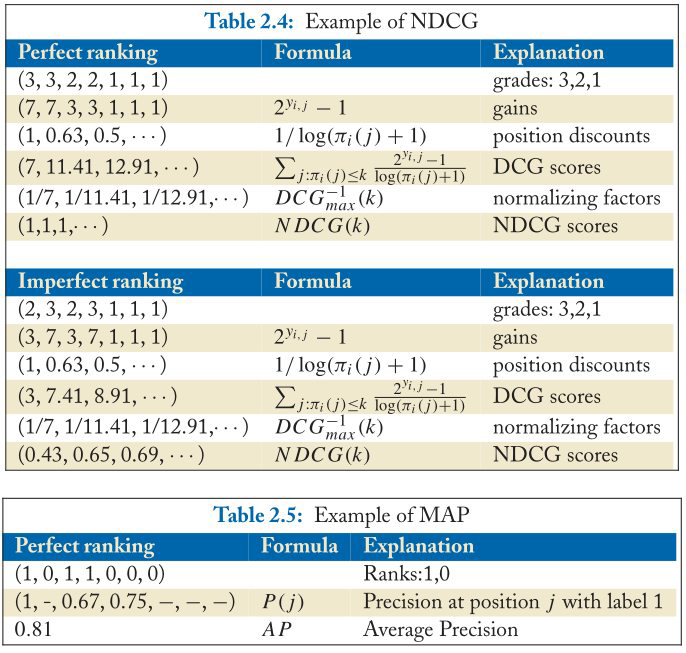
MAP是红外光谱中另一种广泛使用的测量方法。在MAP中，假设相关性等级在两个水平:1和0。给定查询qi，关联文档Di, Di上的排序表πi, Di的标签yi，定义qi的平均精度:



其中yi,j是di,j的标号(grade)，取1或0作为值，表示相关或不相关。定义查询qi的P (j):

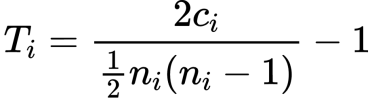


其中πi(j)是di,j在πi中的位置。P (j)表示到di位置的精度，j表示qi。注意，标签可以是1或0，因此可以定义精度(即标签1的比率)。Average Precision表示查询qi的标号为1的文档的所有位置上的平均精度。

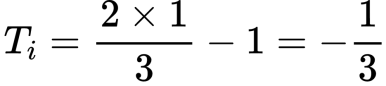


平均精度值在查询中进一步求平均值，成为平均精度(MAP)。表2.5给出了计算某一排行列表AP值的示例。

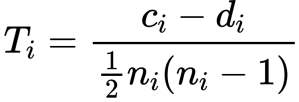
肯德尔Tau是一种统计方法。它是在两个排行榜上定义的:一个是基于排行榜模型的排行榜，另一个是基于地面真理的排行榜。等级表πi相对于地面真理π∗i的Kendall Tau定义为:



其中ci表示两个列表之间的一致性对的数量，ni表示两个列表的长度。例如(A,B,C)和(C,A,B)两个排名列表之间的Kendall’s Tau如下所示。



Kendall 's Tau的值在−1和+1之间。如果两个排名完全相同，则为+1。如果一个排序顺序与另一个排序顺序相反，则为−1。肯德尔的说法很容易验证Tau也可以写成



其中di表示两个表之间不协调的对的数目。

**2.2.5与其他学习任务的关系**

排序和其他机器学习任务有一些相似之处，如分类、回归和有序分类(有序回归)。然而，两者之间也存在差异。这就是为什么学习排名也是机器学习中一个有趣的研究问题。

分类的输入是一个特征向量x∈$ d,和输出是一个标签∈y, y代表一个类的集合类标签,和学习的目的是学会一个分类器f (x),可以确定一个给定的类标签y特征向量x。

在回归中,输入是一个特征向量x∈d,美元输出是一个实数y∈美元,和学习的目的是学习函数f (x),可以确定一个给定的实数y特征向量x。

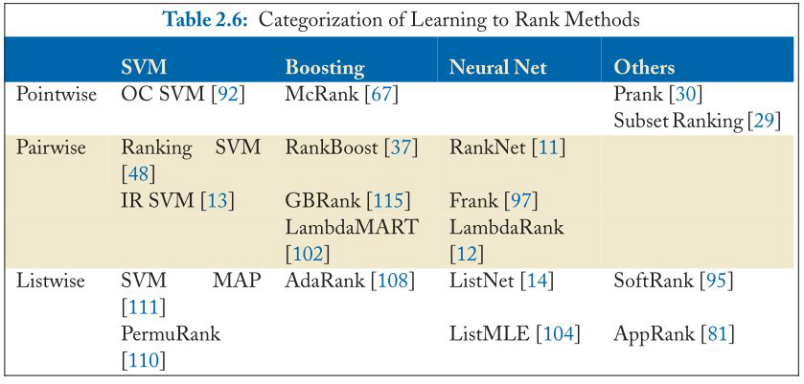
顺序分类(或序数回归)(30、67、92)接近排名,但也不同。输入是一个特征向量x∈$d，输出是一个标签y∈y，表示一个等级，其中y是一组等级标签。学习的目标是学习一个模型f (x)，它可以确定一个给定的特征向量x的等级标签y。模型首先计算分数f (x)，然后使用一些阈值来确定等级标签y。具体来说，该模型将实数轴分割成若干个区间，并为每个区间分配一个等级。然后取f (x)所处的区间的等级作为x的等级。

在排序中，人们更关心对象(提供)的精确排序，而在有序分类中，人们更关心对象的精确有序分类。有序分类的一个典型例子是产品评级。例如，给定一部电影的特征，我们要给这部电影分配一些星级(评级)。在这种情况下，正确分配星星的数量是至关重要的。排序的一个典型例子是文档检索。在文档检索中，给定一个查询，其目标是对文档给出正确的排名，尽管有时训练数据和测试数据在有序分类中被标记为多个等级。要排名的文档数量可以根据查询的不同而不同。有些查询可以在集合中找到更相关的文档，有些查询只能找到弱相关的文档。

如后面所见，可以通过分类、回归和有序分类来近似表示排序。

**2.3学习方法**

排名学习，特别是排名创建学习，近年来得到了广泛的研究。本文提出的方法可分为逐点法、两两法和列表法。还有一些方法可能不属于任何一种方法，例如依赖于查询的排序[41]和多个嵌套排序[74]。



逐点和两两的方法将排序问题转化为分类、回归和有序分类。listwise方法将对象的排名列表作为学习实例，根据排名列表学习排名模型。这些方法之间的主要区别实际上在于所采用的损失函数。

据观察，列表法和两两法通常优于逐点法。在最近的雅虎学习排名挑战中，属于两两方法的LambdaMART取得了最好的表现。

这些方法也可以根据它们所使用的学习技术进行分类。它们包括支持向量机技术、Boosting技术、神经网络技术等。

表2.6对现有方法进行了总结。下面将一一描述。

**2.3.1逐点的方法**

在逐点方法中，将排序问题(创建排序)转换为分类、回归或有序分类，并应用现有的分类、回归或有序分类方法。因此，这种方法忽略了排序的组结构。

更具体地说，逐点方法以图2.2中的训练数据作为输入。它忽略了组织结构和结合一起的组(ξ1,1 ), · · · , ( 习、镍、彝族、镍),我= 1 , · · · , m。训练数据成为典型的监督学习数据(代表映射从x到y)。当我们把y作为一个类标签,实数,与品位标签,那么这个问题就变成了分类、回归,和有序分类。然后，我们可以使用现有的分类、回归或有序分类方法来执行学习任务。

假设学习的模型f (x)输出实数。然后，给定一个查询，我们可以使用模型对文档进行排序(根据模型给出的分数对文档进行排序)。学习中的损失函数是逐点的，因为它是在单个对象(特征向量)上定义的。

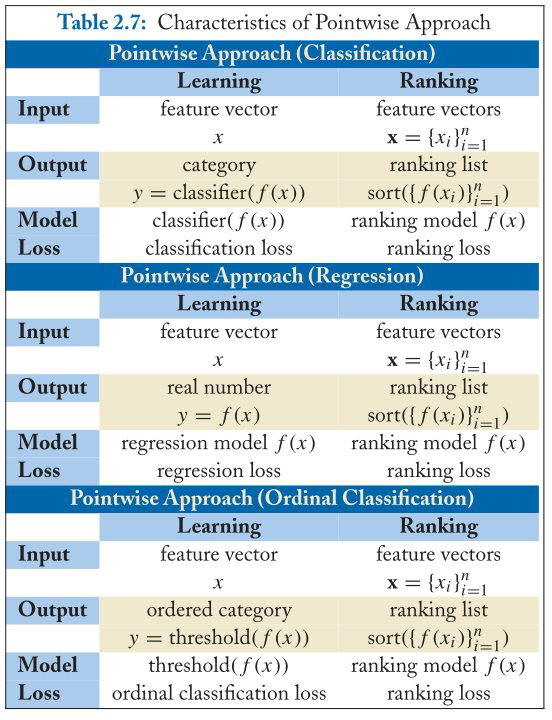


表2.7总结了逐点方法的主要特点。逐点方法包括恶作剧、OC支持向量机、m曲柄和子集排序。第四章详细解释了恶作剧支持向量机和OC支持向量机。(参见[22,23,24])。

Crammer & Singer[30]研究了序数分类，即对给定的对象进行分级。等级可以用于排名，因此他们的方法也可以被视为排名的方法。Crammer & Singer提出了一个简单而有效的在线算法，叫做恶作剧。给定训练数据，恶作剧迭代学习多个并行感知器模型，每个模型将两个相邻的等级分开。作者还从错误界的角度对恶作剧进行了分析。

Shashua & Levin[92]提出了一种大裕度的有序分类方法，在本书中称为OC SVM(有序分类支持向量机)。在他们的方法中，他们也试图学习平行超平面来分离相邻的等级，但他们的机制是大裕度原则。他们考虑了两种定义边际的方法。第一种假设，两者之间的边际相邻的年级相同，学习的边际最大。第二个允许不同的相邻等级有不同的边界，并且边界的和是最大化的。

Li等[67]将排序问题转化为多类分类，提出了m曲柄算法。作者的动机是基于动态分类的分类误差受多类分类误差的限制。他们学习并利用一种分类模型，该模型可以分配一个对象成为等级成员的概率。然后，他们计算对象的预期分数，并使用预期分数对对象进行排序。使用梯度推进树算法学习类概率。

Cossock和Zhang[29]开发了子集排序算法。他们首先考虑使用DCG作为评价指标。由于基于DCG的损失函数的最小化是一个非凸优化问题，因此该算法采用了替代损失函数，即原损失函数的上界。基于回归误差定义代理损失函数。因此，原来的排序问题可以作为回归问题来解决。Cossock和Zhang在回归的基础上推导出了任务的学习方法。他们还研究了泛化能力和学习方法的一致性。

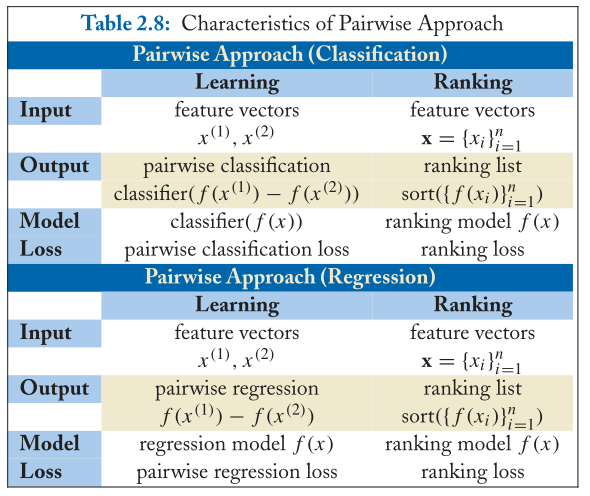
**2.3.2成对的方法**

在两两方法中，排序被转换成两两分类或两两回归。例如，可以创建用于对文档对排序顺序进行分类的分类器，并将其用于对文档进行排序。在两两方法中，也忽略了排序的组结构。

其中，两两方法以图2.2中的训练数据作为输入。从查询qi， (xi,1, yi,1)，···，(xi,ni, yi,ni)， i = 1，···，m的标注数据中创建特征向量(文档)的偏好对。例如，如果xi,j的等级高于xi,k (yi,j > yi,k)，那么xi,j % xi,k就变成了偏好对，这意味着xi,j领先于xi,k。在新的分类问题中，可以将偏好对视为实例和标签。例如，xi,j % xi,k是一个正实例。我们可以使用现有的分类方法来训练分类器进行分类。然后可以使用分类器f (x)进行排序。更准确地说，文档由f (x)分配分数，并根据分数进行排序。好排序模型的训练是通过对好分类模型的训练来实现的。学习中的损失函数是成对的，因为它是在一对特征向量(文档)上定义的。表2.8总结了两两方法的主要特点。

两两方法包括rank SVM、RankBoost、RankNet、IR SVM、GBRank、Frank、LambdaRank和LambdaMART。第四章详细介绍了rank SVM、IR SVM、GBRank、RankNet和LambdaRank。(参见[35,77,86,98,116])。

排序支持向量机是最早的学习排序方法之一，由Herbrich et al.[48]提出。其基本思想是将排序问题形式化为两两分类，并使用支持向量机技术来执行学习任务。不同等级的对象用于生成偏好对(哪个对象在哪个对象前面)，并将其视为表示对象对映射的数据



订单。Herbrich等人的研究表明，当分类器是线性模型(或应用核技巧后的线性模型)时，可以直接将其用于排序。

RankBoost算法由Freund et al.[37]开发。在他们的工作中，他们提出了一个正式的排序框架，即通过结合一些排序特征来学习排序对象的问题。然后他们提出了基于Boosting技术的RankBoost算法。他们展示了理论结果，描述了算法的行为，在训练数据和测试数据。给出了算法的一个有效实现。

Burges等人[11]提出了RankNet算法，该算法也是基于像Ranking SVM和RankBoost这样的两两分类。主要区别在于它采用神经网络作为排序模型，使用交叉熵作为损失函数。Burges等人也证明了交叉熵作为排序损失函数的良好性质。他们的方法使用梯度下降来学习最优神经网络模型。

上述方法的优点是可以直接应用现有的分类和回归学习技术。缺点是学习的目标可能与预测的目标不一致。事实上，对排名的评价通常是基于诸如NDCG之类的指标，这些指标是在对象列表中定义的。相比之下，两两方法的训练是通过提高两两分类或两两回归的准确性来驱动的，这是在成对的对象上定义的。

Cao et al.[13]提出采用成本敏感的排序支持向量机来克服这一缺点。作者指出，在对文档检索进行排序时必须考虑两个因素。首先，正确地将文档排在列表的顶端是至关重要的。二是相关数量文档可以因查询的不同而不同。他们修改了排序支持向量机中的铰链损失函数来处理这些问题，然后将学习任务形式化为代价敏感排序支持向量机。这种方法通常被称为红外支持向量机。分别采用梯度下降法和二次规划法求解学习中的优化问题。

Zheng等人[115]建议采用两两回归进行排名。提出了GB(梯度提升树)秩算法。他们首先引入了一个用于排序的回归框架，该框架使用了一个两两回归损失函数。然后，他们提出了一种新的基于梯度增强树算法的优化方法，以迭代地最小化损失函数。他们使用两种类型的相关性判断来获得训练数据:人工判断的绝对相关性判断和从点击数据中提取的相对相关性判断。

Tsai等人[97]提出的Frank方法是基于类似IR SVM的动机。作者建议在秩网中使用一种新的损失函数。损失函数“保真度损失”不仅保留了交叉熵损失的优良性质，而且还具有一些令人满意的新性质。特别是，保真度损失在0和1之间有界，这使得学习方法对噪声具有更强的鲁棒性。在Frank中，一个神经网络模型是使用梯度下降训练为排名模型。

由Burges等人开发的LambdaRank通过使用隐式的两两损失函数来解决这一挑战2。具体来说，LambdaRank考虑通过梯度下降优化损失函数来学习最优排序模型。事实上，它只是显式地定义了损失函数的梯度函数，称为Lambda函数。例如，LambdaRank使用了神经网络模型。Burges等人也给出了隐代价函数为凸的充要条件。

Wu等人[102]利用Boosting和LambdaRank中的Lambda函数提出了一种名为LambdaMART的新方法。采用MART (Multiple Additive Regression tree)算法[38]学习一棵增强回归树作为排序模型。MART实际上是一个在函数空间中进行梯度下降的算法。LambdaMART在MART中专门使用Lambda函数作为梯度。wu等人验证了LambdaMART的效率明显优于LambdaRank，其准确性也更高。

**2.3.3 LISTWISE方法**

列表方式以一种更自然的方式处理排序问题。具体来说，它将排名列表作为学习和预测的实例。保留了排序的群结构，可以更直接地将排序评价测度纳入损失函数中。

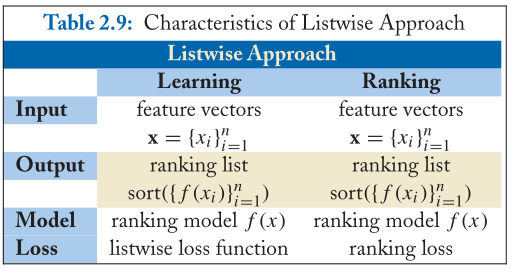
更具体地说，listwise方法将图2.2中的训练数据作为输入。它将与查询qi相关的标记数据(xi,1, yi,1)、···、(xi,ni, yi,ni)视为一个实例。该方法从训练数据中学习一个排名模型f (x)，该模型可以将分数分配给特征向量(文档)，并使用分数对特征向量进行排名，这样分数越高的特征向量排名越高。这是机器学习和传统技术面临的一个新问题机器学习不能直接应用。最近，高级学习排序技术已经发展，其中许多属于列表方法。表2.9总结了列表方式的主要特点。

列表方式包括ListNet、ListMLE、AdaRank、SVM MAP、Soft Rank和ap恶作剧。第四章详细介绍了ListNet、ListMLE、AdaRank、SVM MAP和Soft Rank。(参见[18,85,99,100])。

Cao等人[14]指出了使用列表方法进行排序的重要性，其中对象列表被视为“实例”。他们提出使用Luce-Plackett模型来计算对象列表的排列概率或top k概率。基于此思想的ListNet算法随后被开发出来。该方法采用神经网络模型作为模型，KL散度作为损失函数。一个对象列表的排列概率或top k概率由Luce-Plackett模型计算。KL发散度通过排列概率分布或top k概率分布来衡量学习到的排名列表和真实排名列表之间的差异。采用梯度下降法作为优化算法。Xia等[104]将ListNet扩展到ListMLE，其中日志似然定义为损失函数。ListMLE的学习相当于参数化Luce-Plackett模型的极大似然估计。

在红外等应用中，评价指标是在对象列表中定义的。理想情况下，学习算法会训练一个可以直接优化评估措施的排名模型。另一组列表法试图直接优化学习评价指标。其中包括Xu和Li开发的AdaRank[108]。AdaRank通过使用Boosting技术使直接定义在评估措施上的损失函数最小化。在重加权训练数据的基础上，反复构造“弱秩”，最后将弱秩线性组合进行排序预测。AdaRank是一个非常简单和高效的学习排序算法。

Yue等人[111]提出的SVM MAP算法也考虑了评价指标的直接优化，特别是在IR中使用的MAP。SVM MAP是一种基于MAP的SVM学习算法，它能高效地找到全局最优解，使损失函数的上界最小化。Xu等人[110]表明，可以将这个想法扩展到推出一组



算法。这些算法的关键思想是引入基于红外评价测度(定义在对象列表上)的损失函数，考虑损失函数的不同上界，并利用SVM技术对上界进行优化。不同的上界和不同的优化技术可以导致不同的算法，包括一个称为PermuRank的算法。

另一种列表方法试图近似评估措施。SoftRank就是其中的一个典型算法。直接优化评价措施的主要挑战是，由于损失函数的性质，这些措施不能直接优化。Taylor等人[95]提出了一种软(近似)计算对象秩分布的方法。用一个近似的等级分布，可以近似地计算评估措施，如NDCG。他们表明SoftRank是一种非常有效的优化评估措施的方法。

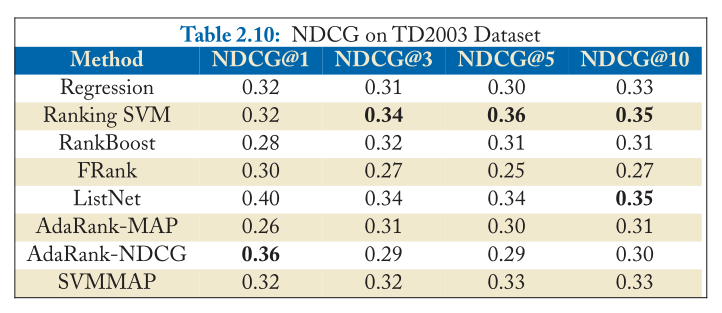
Qin等人[81]提出了在学习排序时直接优化IR测度的一般框架。在该框架中，将NDCG和MAP等IR度量近似为代理函数，然后对代理函数进行优化。该方法的关键思想如下。直接优化IR度量的困难在于，度量是基于等级的，因此相对于等级函数的分数输出是非连续的和不可微的。该方法通过一个连续可微的函数逼近文档的秩。在此基础上，开发了一种ap恶作剧算法。

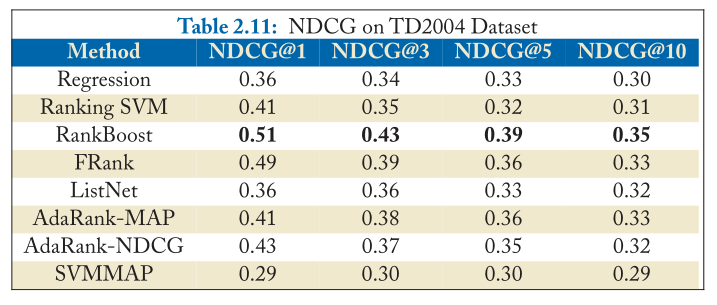
**2.3.4评价结果**

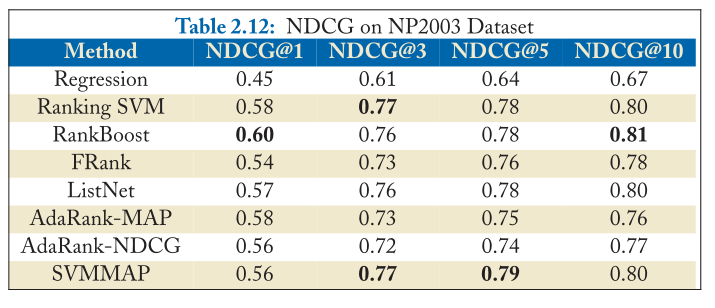
根据以往的研究，列表方式和两两方式通常比点方式更有效。就像在其他机器学习任务中一样，没有一种单一的方法总是能优于其他方法。这是学习排序方法的一般趋势。

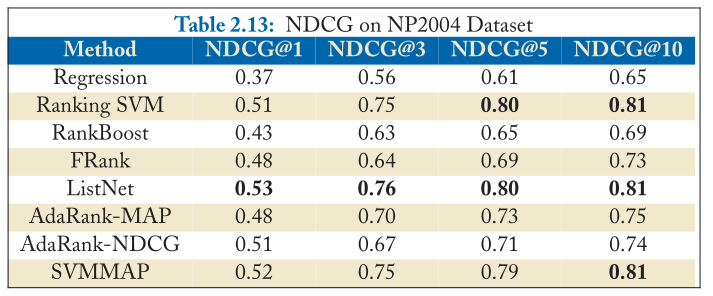
表2.10-2.16给出了在LETOR数据集上使用NDCG对多种方法的实验结果[82]。LETOR数据集是学习排名的基准数据，由Microsoft Research 3的研究人员从TREC数据中导出。我们可以大致了解这种方法所取得的效果。

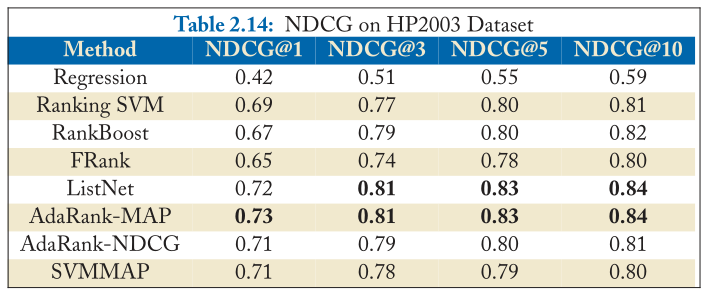
在Yahoo Learning to Rank Challenge 4中，LambdaMART的两两方法取得了最好的表现。事实上，表现最好的学生的准确率非常接近。

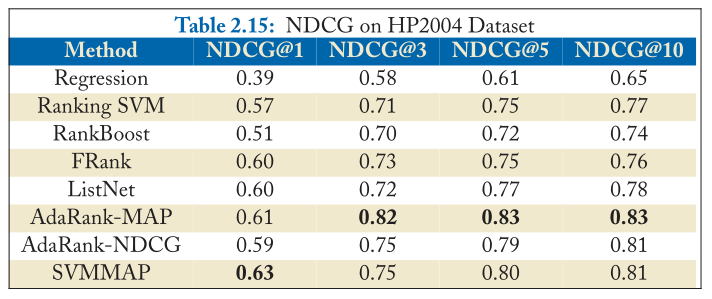


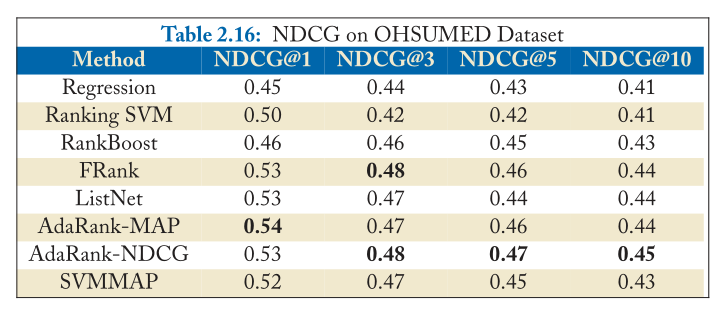












**排序聚合的学习**

本章介绍了排序聚合的学习方法。排名聚合的目的是将多个排名组合成一个排名，从评估度量的角度来看，这个排名比任何一个原始排名都好。排序聚合的学习就是利用机器学习技术建立排序聚合的排序模型。接下来，我们以元搜索为例进行说明。在不丧失通用性的前提下，所描述的技术可以应用于其他应用程序。

**3.1学习任务**

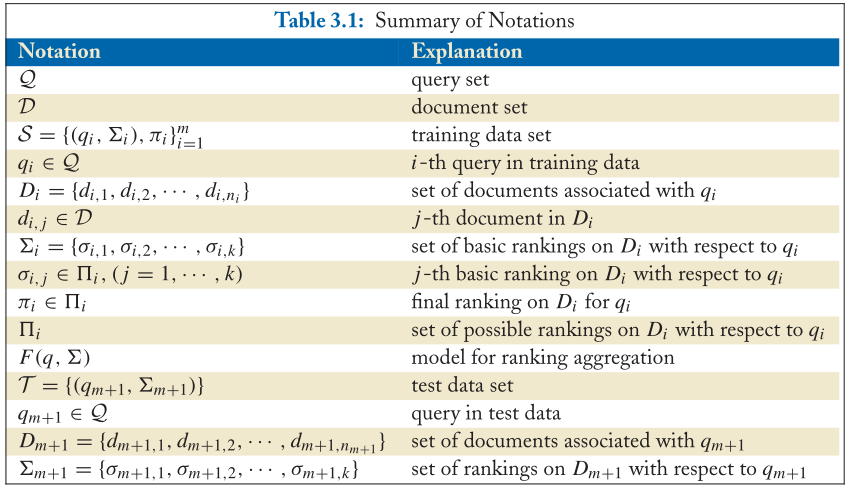
在元搜索中，用户的查询被发送到多个搜索系统，然后搜索系统的排名列表被组合在一起，并以单个排名列表的形式呈现给用户。由于来自单个搜索系统的排名列表可能不够准确，元搜索实际上对搜索排名列表采取了多数投票。接下来的问题是如何有效地执行多数投票。在这里我们将排名从单个搜索系统的基本排名，和排名在元搜索的最终排名。

排序聚合的学习既可以是无监督学习，也可以是有监督学习。在传统的信息检索中，排序聚合通常是基于无监督学习的。最近，有监督的聚合排序方法也被提出。

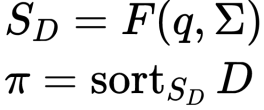
在进行排序聚合的监督学习中，训练数据包含查询、关联的文档、对文档的基本排名以及相应的最终排名。测试数据包括查询、关联文档和对文档的基本排名。

假设Q是查询集，D是文档集。进一步假设{q1, q2，···，qm}是训练数据中查询的集合。Di = {Di, 1,, 2 , · · · , di,倪}是一组文档与查询相关的气,“我={我σ1,σ2 , · · · , σ,k}是一组基本的排名在迪对文档查询气,和π我是最后的排名在迪对文档查询气。其中di,j表示di中的第j个文档，σi,j表示第i个基本排名，k表示基本排名的个数。训练集表示为S = {(qi， "i)， πi}mi=1。

在学习的典范,排名聚合构造,其形式的F (q,”):问×# k美元→n,问在哪里查询,D是一组相关的文件,“是一组基本的排名在D文件对q, n表示文档的数量,和k表示的数量基本排名。F (q，”)可以给D中的文档打分，对文档进行排序



根据分数，并生成最终排名。



注意，F是一个全局排序函数，因为它是在一组文档上定义的。

测试数据由查询qm+1、关联文档Dm+1和对文档m+1的基本排名组成。我们使用训练过的排名模型F (q，”)对Dm+1中的文档进行评分，并根据评分对其进行排序，得出最终的排名列表。测试数据集表示为T = {(qm+1， "m+1)}。

表3.1给出了符号的摘要。

到目前为止，排序聚合通常被定义为一个与查询相关的任务。在实践中，它通常被指定为一个查询独立任务。也就是说,



注意，我们使用整数来表示文档。设Di中的文档由整数{1,2，···，ni}标识。我们定义排列πi在Di上为{1,2，···，ni}到它自身的双射。我们用#i表示Di上所有可能的排序列表(排列)的集合，用πi(j)和π−1i (j)分别表示第j个文档的秩和第j个文档的秩。

注意，为了简单起见，这里假设所有的基本排名都位于同一组文档Di上。在实践中，可以对文档的不同子集提供不同的基本排名。在这种情况下，我们可以取子集的并并定义为Di。

排序聚合中的评价措施可以是学习排序的任何措施，这取决于地面真相是如何给定的。例如，如果ground truth中的最终排名作为一个排名列表给出，那么可以使用Kendal’s Tau。如果最终的排名是基于等级的，那么可以使用MAP或者NDCG。

**3.2学习方法**

现有的排序聚合方法包括无监督学习方法，如Borda Count和Markov Chain，以及监督学习方法，如Cranking。在第四章中，我们对波达数、马尔可夫链和曲柄进行了详细的说明。也见[26,59,71]。

博尔达计数是根据选民对候选人的排名选出最佳候选人的选举方法。Aslam & Montague[8]将其应用于元搜索，并验证了该方法在元搜索中的有效性。Borda计数是一种简单的方法，如下。首先，每个投票人给出n个候选人(对象)的排名。对于每个排名，排名最高的候选人获得n分，排名第二的候选人获得n-1分，以此类推。然后按照总分降序排列候选人，生成最终的排名列表，得分最多的候选人获胜。

基于马尔可夫链的排序聚合假设对象上存在马尔可夫链。利用对象的基本排序构造马尔可夫链，根据排序中的顺序关系估计转移概率。然后利用马尔可夫链的平稳分布对最终排序的对象进行排序。Dwork等人[34]提出了四种构造马尔可夫链转移概率矩阵的方法，从而产生了四种不同的马尔可夫链排序聚合方法。

黎巴嫩和Lafferty[63]提出的曲柄操作采用了malallows模型的一种泛化方法进行排序聚合。广义马洛斯模型以指数函数的形式，定义了在给定多个基本等级的情况下，物体最终等级的条件概率分布。该模型可以看作是排序聚合的一般概率模型。Lebanon和Lafferty提出了几种在不同设置下估计广义malallows模型参数的学习方法，特别是当训练数据中的排名只有部分可用时。

**学习排名的方法**

本章描述了细节11排名方法创建,包括恶作剧[30],OC SVM[92],排名SVM(47、48),红外SVM [13], GBRank(114、115),RankNet [11], LambdaRank[12日32],ListNet & ListMLE [104], AdaRank [108], SVM地图[111],和SoftRank[95],和三个排名聚合方法,包括Borda Count[34]、Markov Chain[34]和CRanking[63]。

**4.1 PRANK**

恶作剧(Perceptron Ranking)是Crammer & Singer[30]提出的一种在线排序算法。我们可以在排名(排名创建)中使用它作为逐点方法。恶作剧的基本思想是利用和学习多个并行感知器模型，每个模型在相邻的等级之间进行分类。

**4.4.1模型**

设X⊆$d, Y ={1,2，···，l}，其中Y有一个总序。X∈X是一个对象(特征向量)，Y∈Y是一个表示等级的标签。给定对象x，我们的目标是预测它的标号(grade) y，也就是说，这是一个序数分类问题。我们雇佣的线性模型(感知器),w, x -−br (r = 1 , · · · , l−1)预测,其中w∈d是一个权重向量和br∈美元,(r = 1 , · · · , l)偏差满足b1≤···≤提单−1≤提单= +∞。模型对应平行超平面，w, x-−br = 0，分隔等级r和r + 1， (r = 1，···，l - 1)，其中-表示点积。图4.1给出了一个恶作剧模型的例子。如果x满足，w, x-−br−1≥0且，w, x-−br < 0，则y = r， (r = 1，···，l)，可以写成minr∈{1，···，l}{r|，w, x-−br < 0}。

**4.1.2学习算法**

恶作剧采用感知器学习算法[91]在线同时学习线性模型。感知器学习算法是基于随机梯度下降，恶作剧也是如此。恶作剧在每一轮中需要一个输入对。假设在当前一轮中，输入对为(x, y)，我们将更新权值w和偏差br， (r = 1，···，l - 1)。为简单起见，我们省略了表示这一轮的上标。给定一个特征向量x，当前的模型可以预测它的评级为ˆy。具体来说,如果x满足,w, x -−br−1≥0,w x -−br < 0,那么预测年级应该ˆy = r, (r = 1 , · · · , l)。另一方面,考虑到真正的等级标签y,也有可能说,模型应该预测的特征向量作为正面例子和模型

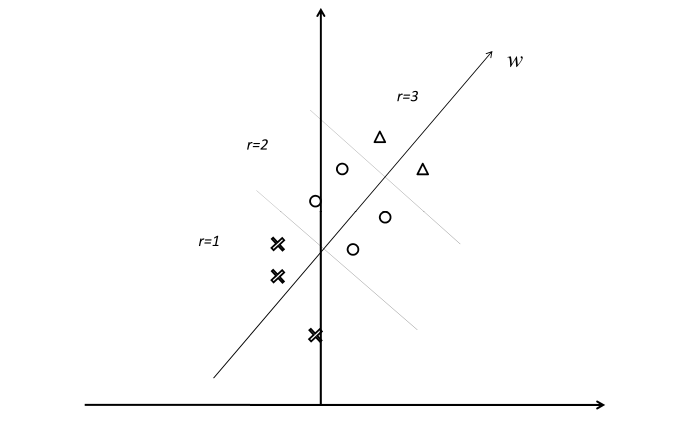


图4.1:恶作剧模型

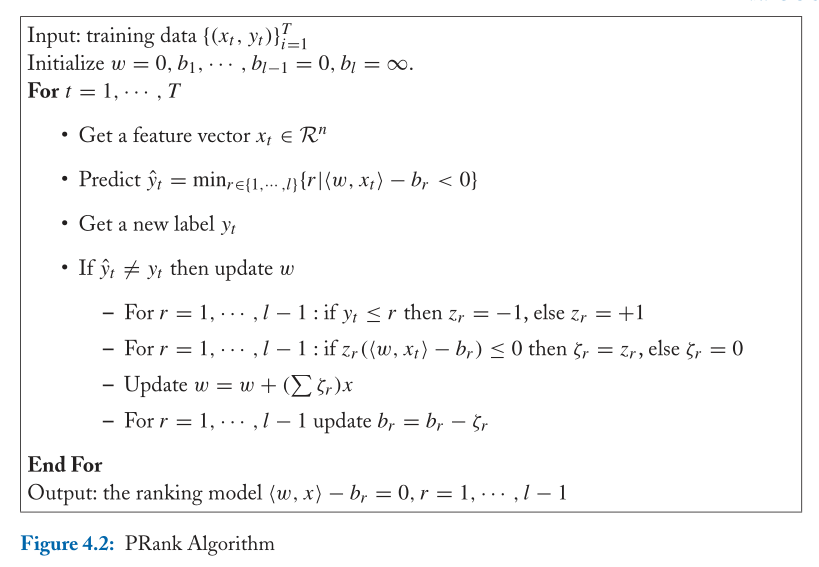
把它作为反面例子来预测。我们使用变量(z1 , · · · , zl−1)= (+ 1 , · · · , + 1,−1 , · · · , −1)代表事实,首先y−1变量对应的模型应该做出积极的预测(+ 1),其余变量的模型应负预测(−1)。因此，如果模型的预测是正确的，则zr(，w, x-−br) > 0成立;如果预测不正确，则zr(，w, x-−br)≤0保留(r = 1，···，l−1)，当出现错误时，恶作剧会对所有产生错误的模型进行权重调整。具体来说，胡闹用br−zr更新这些模型的偏差br，用w + ((zr)x更新权值w，其中权值的总和取代了产生误差的模型。恶作剧算法如图4.2所示。

**4.2 OC SVM**

Shashua & Levin[92]提出的方法也是利用多个平行超平面作为排序模型。他们的方法通过大裕度原理学习平行超平面。在一个实现中，该方法试图为所有相邻的年级最大化固定的边际。

**4.2.1模型准备**

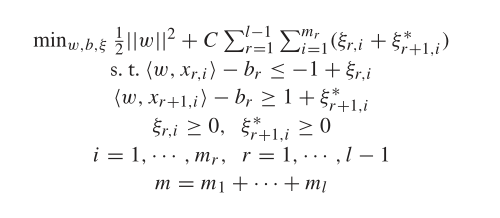
设X⊆$d, Y ={1, 2，···，l}，其中Y上存在一个总序。X∈X是特征向量，Y∈Y是表示一个等级的标号。作为在恶作剧,我们采用平行超平面,w, x -−br = 0, (r = 1 , · · · , l−1)预测给定特性的标签y向量x,其中w∈d是一个权重向量和br∈美元,(r = 1 , · · · , l)偏差满足b1≤···≤提单−1≤

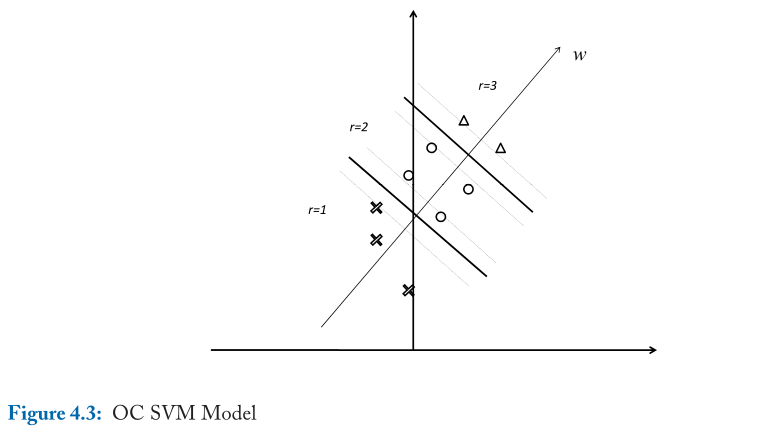


提单= +∞。如果x满足，w, x-−br−1≥0，且，w, x-−br < 0，则y = r， (r = 1，···，l)，也就是说，预测是基于minr∈{1，···，l}{r|，w, x-−br < 0}。

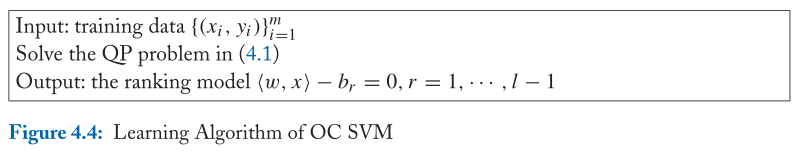
**4.2.2学习算法**

OC SVM假设平行超平面将任意两个相邻等级中的实例以相同的大间隔分开(图4.3)。假设训练数据为:对于每个年级r = 1，···，l，都有mr实例:xr,i, i = 1，···，先生。学习任务形式化为下面的二次规划(QP)问题。





其中xr,i表示第r级的第i个实例，ξr+1,i和ξ∗r+1,i表示相应的松弛变量，||·||表示L2范数，m为训练实例数，C > 0为系数。具体解决方法见[92]。图4.4为OC SVM的学习算法。

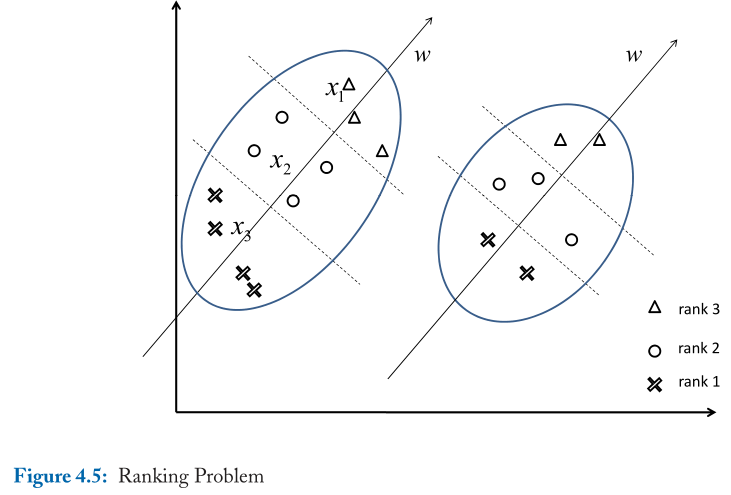


**4.3排名支持向量机**

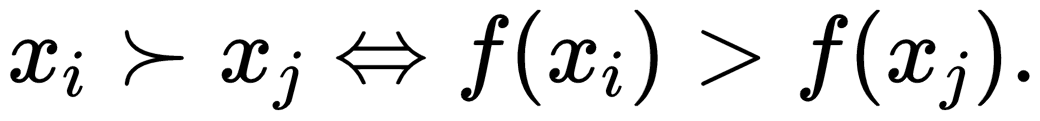
排序支持向量机是最早的学习排序方法之一，由Herbrich等人提出[47,48]。排序支持向量机的基本思想是将排序转化为两两分类，利用支持向量机技术[27]来完成学习任务。

**4.3.1线性模型作为排序函数**

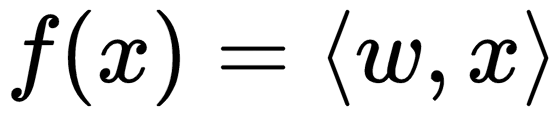
假设X⊆$d是特征空间，X∈X是空间中的一个元素(特征向量)。进一步假设f是一个评分函数f: X→$。即给定任意两个特征向量xi∈X和xj∈X，如果f (xi) > f (yj)，



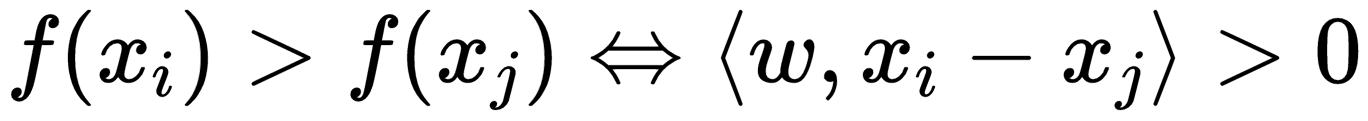
那么xi应该排在xj前面，反之亦然。



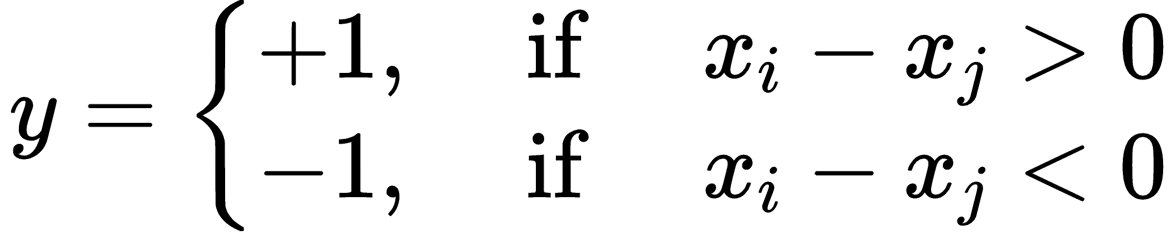
原则上，函数f (x)可以是任何函数。为了简单起见，我们暂时假设f (x)是一个线性函数。



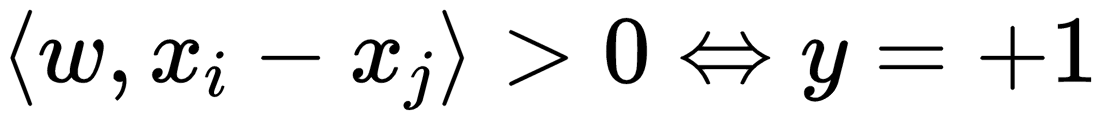
其中w为权重向量，·-为内积。如果评分函数是一个线性函数，我们可以将排序问题转化为一个二元分类问题。原因如下。首先，当f (x)为线性函数时，对于任意两个特征向量xi和xj都有如下关系。



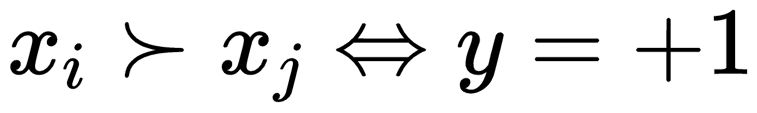
接下来，对于任意两个特征向量xi和xj，我们可以考虑一个关于特征向量xi−xj之差的二元分类问题。具体来说，我们给它分配了一个标签y。



因此，



因此，下列关系成立。也就是说，如果xi排在xj前面，那么y = +1，否则y =−11。



**4.3.2对SVM模型进行排序**

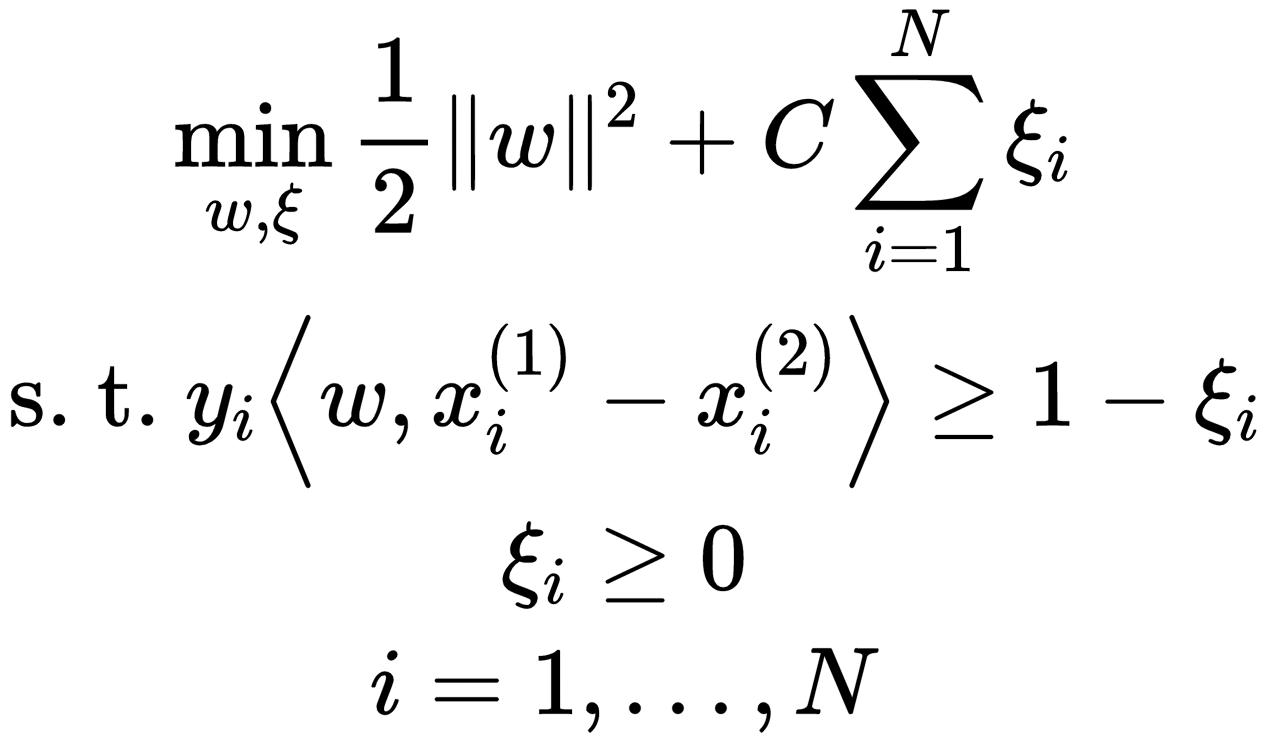
我们可以学习和利用线性分类器，例如线性支持向量机的排序任务。该分类器可以直接用作排序模型。我们也可以使用核技巧将线性模型扩展到非线性模型。我们将上述方法称为“排序支持向量机”。

图4.5显示了一个排序问题的示例。假设特性空间中有两组对象(与两个查询关联的文档)。进一步假设有三个等级(水平)。例如，第一组的对象x1, x2，和x3是在三个不同的等级。权值向量w对应线性函数f (x) =，w, x-，可以对对象进行评分和排序。使用函数对对象进行排序相当于将对象投影到向量中，然后根据向量上的投影对对象进行排序。如果排名函数是“好的”，那么3级对象的排名应该比2级对象的排名要高，等等。请注意，属于不同组的对象是不可比较的。

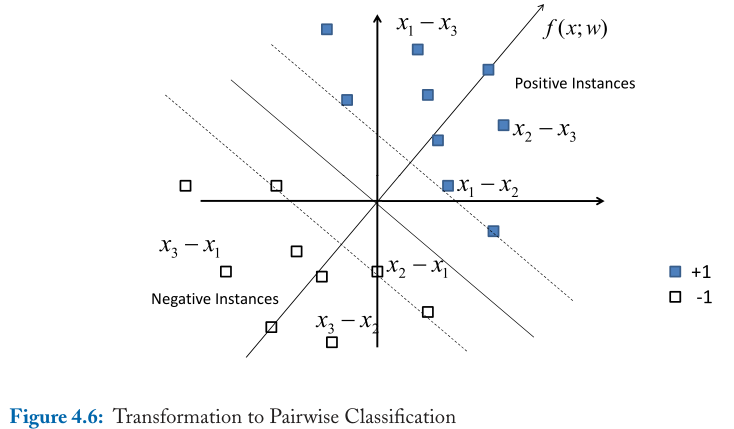
图4.6展示了如何将图4.5中的排序问题转化为线性SVM分类。将同一组中不同等级的两个特征向量之间的差异作为新的特征向量，如x1−x2、x1−x3、x2−x3。此外，对新的特征向量也进行了标签赋值。例如:x1−x2, x1−x3, x2−x3为正。注意，相同等级的特征向量或不同组的特征向量不会被用来创建新的特征向量。我们可以训练一个线性SVM分类器，将新的特征向量分离出来，如图4.6所示。需要注意的是，SVM分类器的超平面会传递原始数据，正实例和负实例都是反对称的。例如，x1−x2, x2−x1为正实例，x2−x1为负实例。事实上，我们可以在学习中抛弃消极的例子，因为它们是多余的。

**4.3.3学习算法**

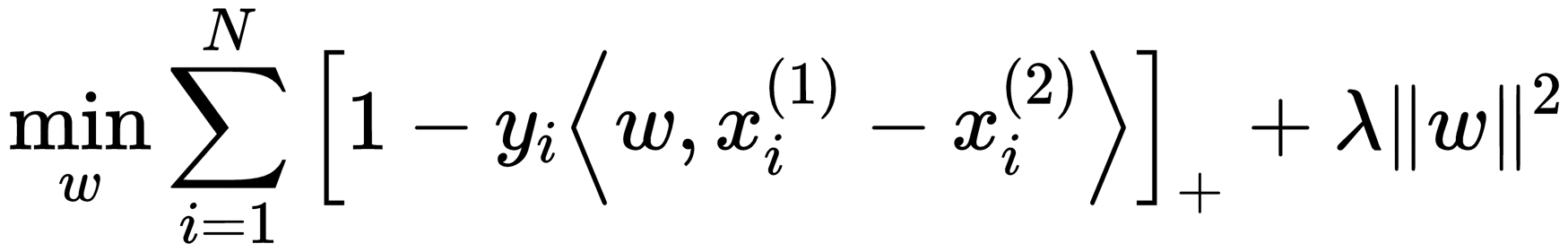
更正式地说，排序支持向量机被形式化为以下的约束优化问题(二次规划)。我们首先考虑线性情况，



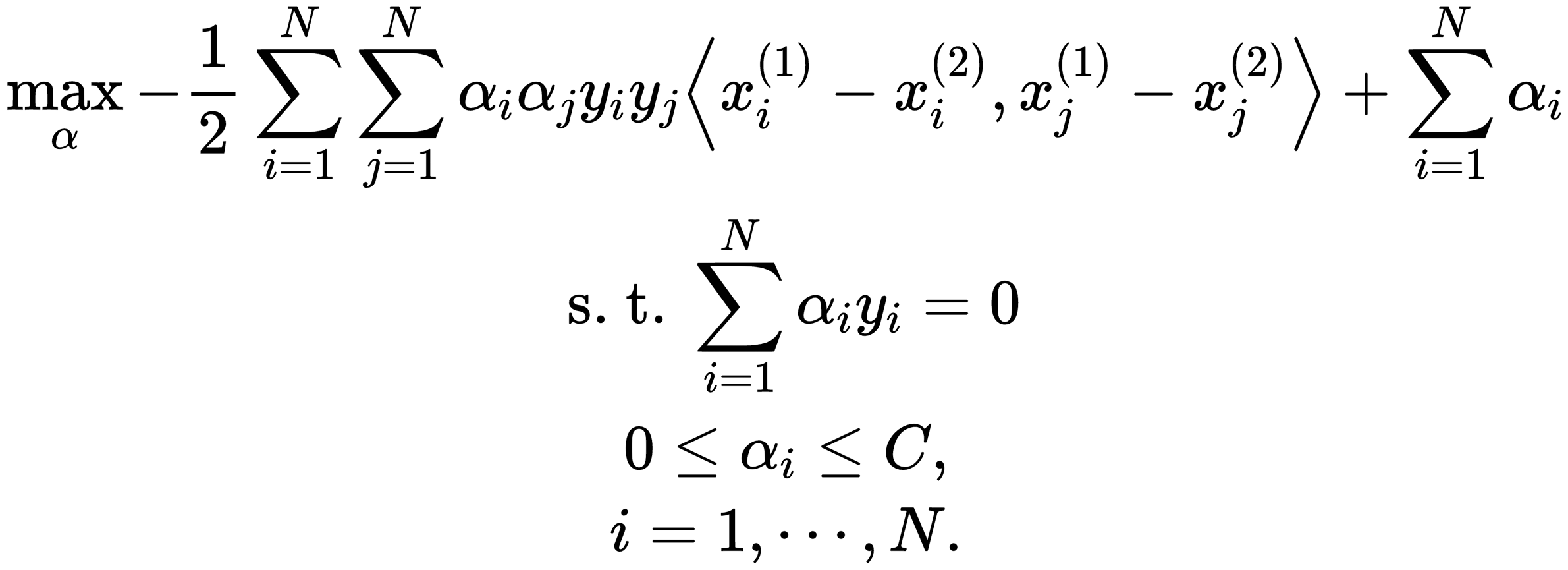
其中x(1) i和x(2) i表示一对特征向量中的第一个和第二个特征向量，||·||为L2范数，N为训练实例数，C > 0为系数。



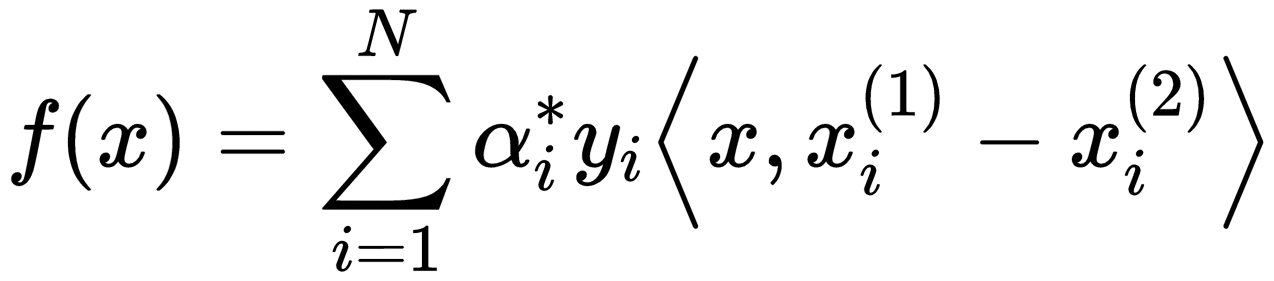
等价于以下无约束优化问题，即正则化铰损函数的最小化。



其中[x]+表示函数max(x, 0)， λ = 12C。原QP问题可以通过求解对偶问题来求解。

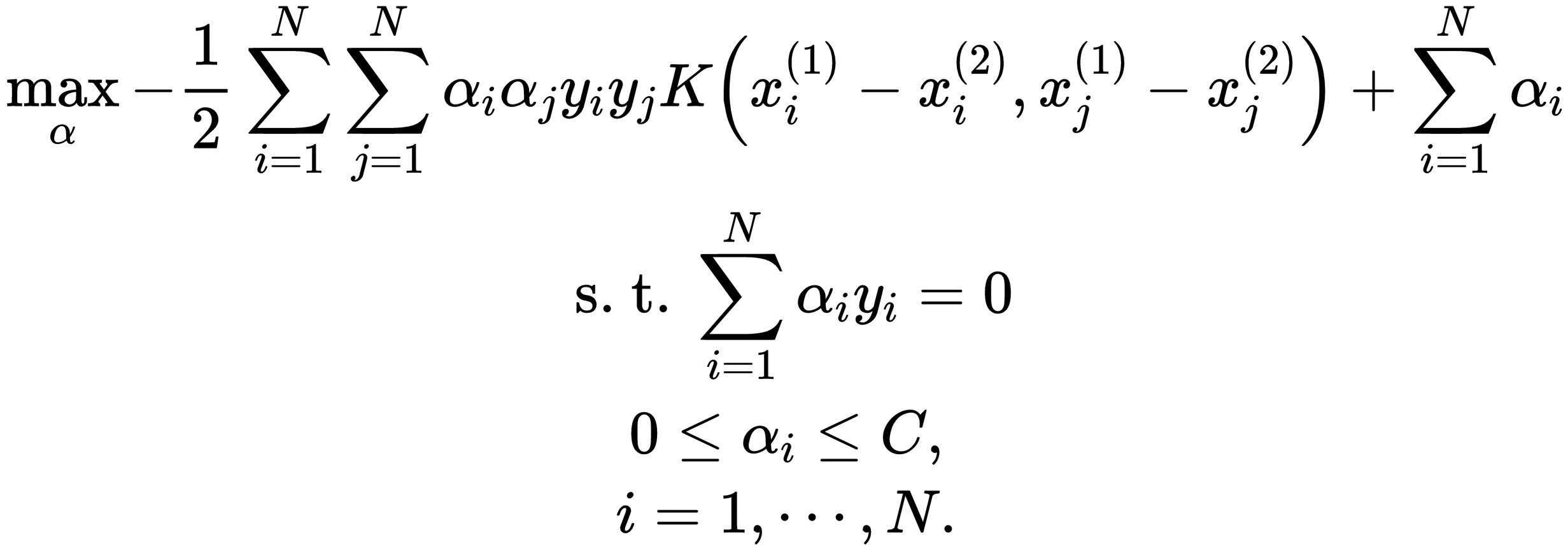


最优解作为排序函数，给出为

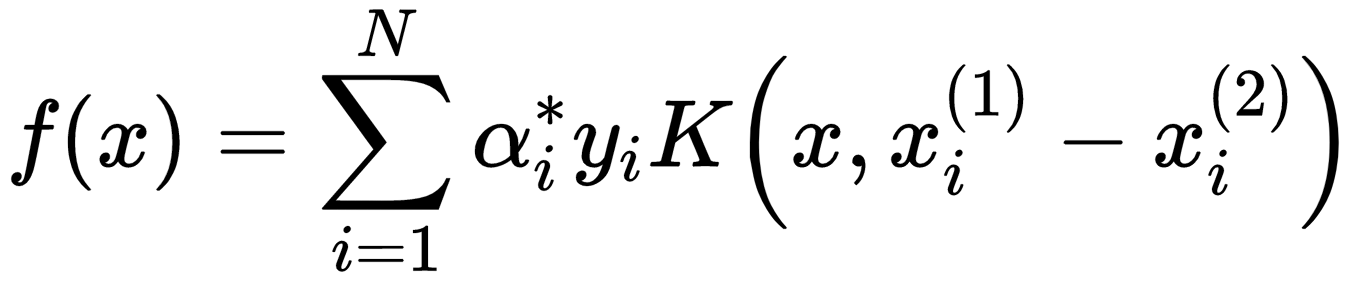


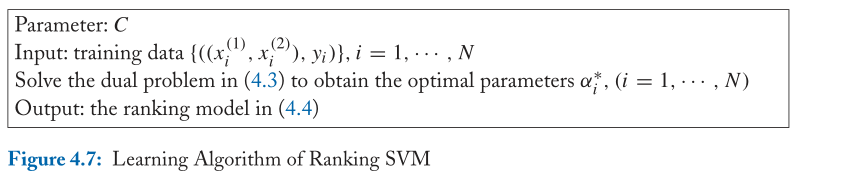
Ranking SVM的学习算法如图4.7所示。

我们也可以利用核技巧将上述问题推广到非线性情况



最优排序函数如下所示





**4.4 IR SVM**

Cao等人[13]提出的红外支持向量机(IR SVM)是对信息检索排序支持向量机(Ranking SVM)的扩展，其思想也可以应用到其他应用中。

**4.4.1修正损失函数**

排序支持向量机将排序转化为两两分类，因此在学习过程中实际上是利用了0-1的损失。损失函数与IR评价指标之间存在差距。IR SVM试图通过修改0-1损失来弥补这一差距，即对Ranking SVM进行代价敏感学习。

我们首先看看直接应用Ranking SVM进行文档检索所带来的问题，如图4.8所示。

直接应用Ranking SVM的一个问题是，它平等地对待不同等级的文档对。在示例1中，有三对文档。分别是文件对和标签对(等级对)3-2、3-1、2-1。对SVM进行排序对文档对使用相同的0-1损失。这与不同的文档对在排序中应该具有不同重要性的事实形成了对比。实际上，对3-1(3排在1前面)进行正确的排序比其他对更重要。例2从另一个角度说明了问题。同一查询有两种排名。在rank -1中，位置1和2的文档由完全位置互换，而在rank -2中，位置3和4的文档由完全位置互换



从完美的排名交换。在0-1的损失方面，每个排名只有一个错误。因此，它们在训练rank SVM上的效果是一样的，这是不理想的。实际上，从IR的角度来看，rank -2应该比rank -1更好，因为它的top的结果更好。同样，对排名靠前的文档具有较高的准确性对IR系统至关重要，这反映在IR评估措施中。

排序SVM的另一个问题是，它平等地对待来自不同查询的文档对。文档的数量通常因查询的不同而不同。在示例3中，有两个查询，并且与它们关联的文档数量是不同的。对于query3, 3-2级之间有2个文档对，3-1级之间有4个文档对，2-1级之间有8个文档对，共14个文档对。对于query4，有31个文档对。Ranking SVM从query3中提取14个实例(文档对)，从query4中提取31个实例(文档对)进行训练。因此，query4对训练过程的影响将大于query3的影响。换句话说，学习到的模型将偏向于query4。这与IR评估查询同等重要的事实形成了对比。

IR SVM通过将0-1分类转变为代价敏感分类，解决了以上两个问题。这是通过修改排序支持向量机的铰链损耗函数来实现的。具体来说，它为不同级别和不同查询的文档对设置了不同的损失。为了强调正确排名的重要性，loss函数会严重惩罚排名靠前的错误。为了增加使用较少文档的查询的影响，loss函数会严重惩罚此类查询产生的错误。

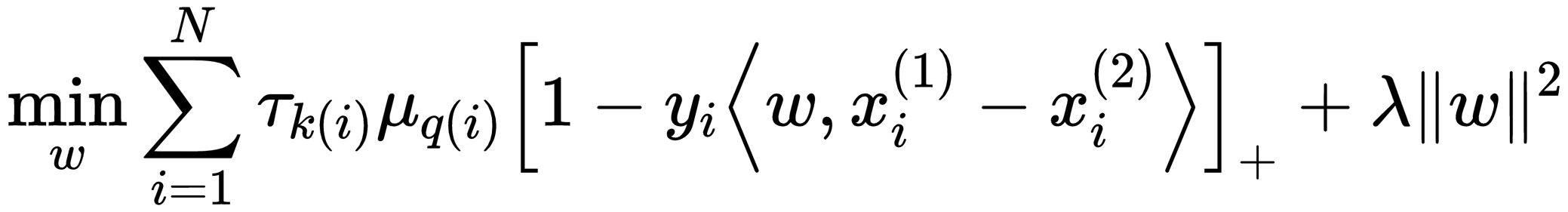
图4.9绘制了不同惩罚参数下不同铰损函数的形状。x轴表示yf (x(1) i−x(2) i)， y轴表示损耗。当yf (x(1) i−x(2) i)≥1时，损耗为零。当yf (x(1) i−x(2) i) < 1时，损耗为不同斜率的线性递减函数。如果斜率为-1，则函数为正常铰损函数。IR SVM对铰链损耗函数进行了修改，具体对不同等级对的坡度进行了修改



不同的查询。它为属于重要等级对的文档对赋予较高的权值，并根据查询对文档对的权值进行规范化。

**4.4.2学习算法**

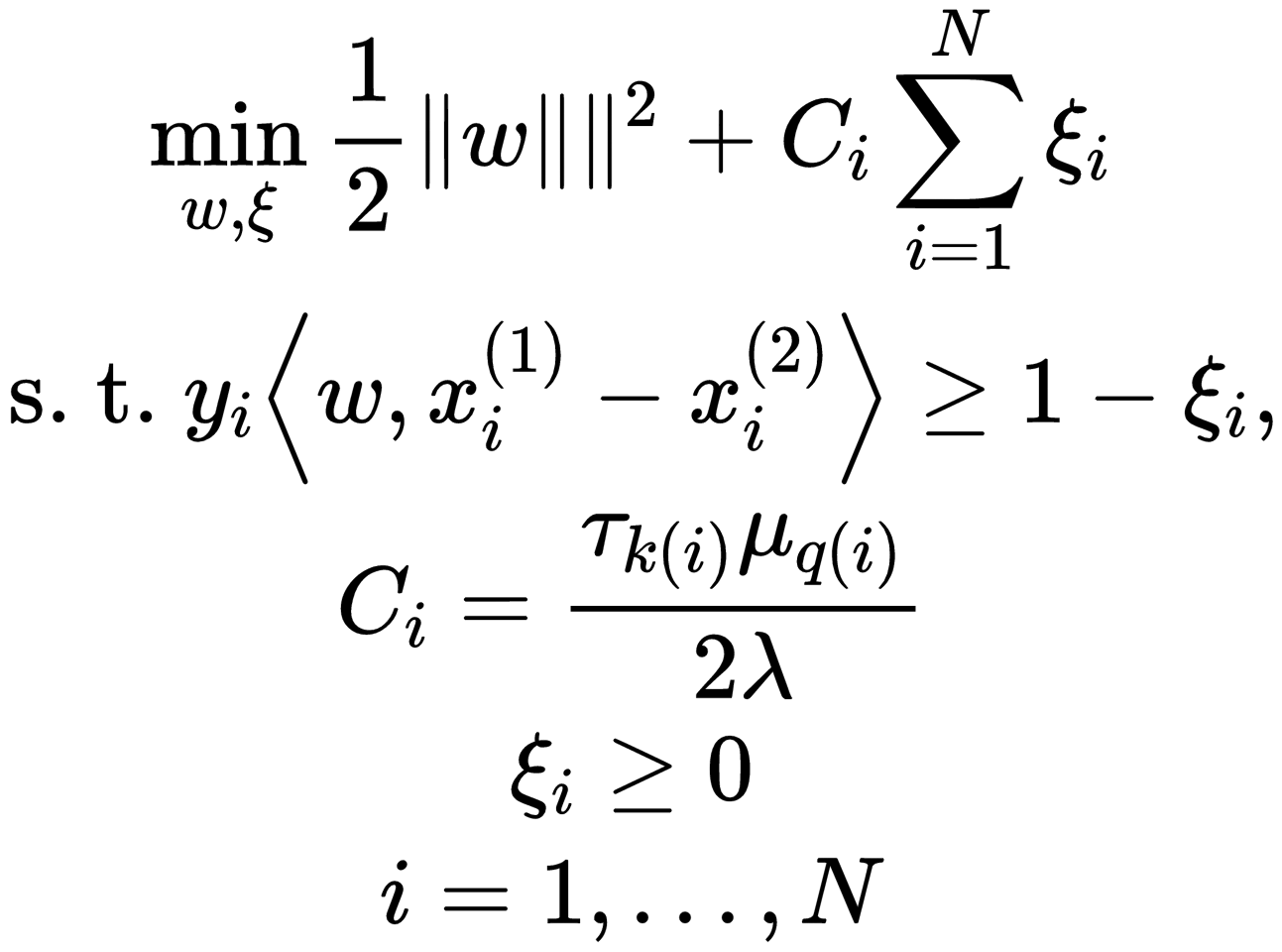
红外支持向量机的学习等价于下面的优化问题。其中，改进正则化铰链损耗函数的最小化，



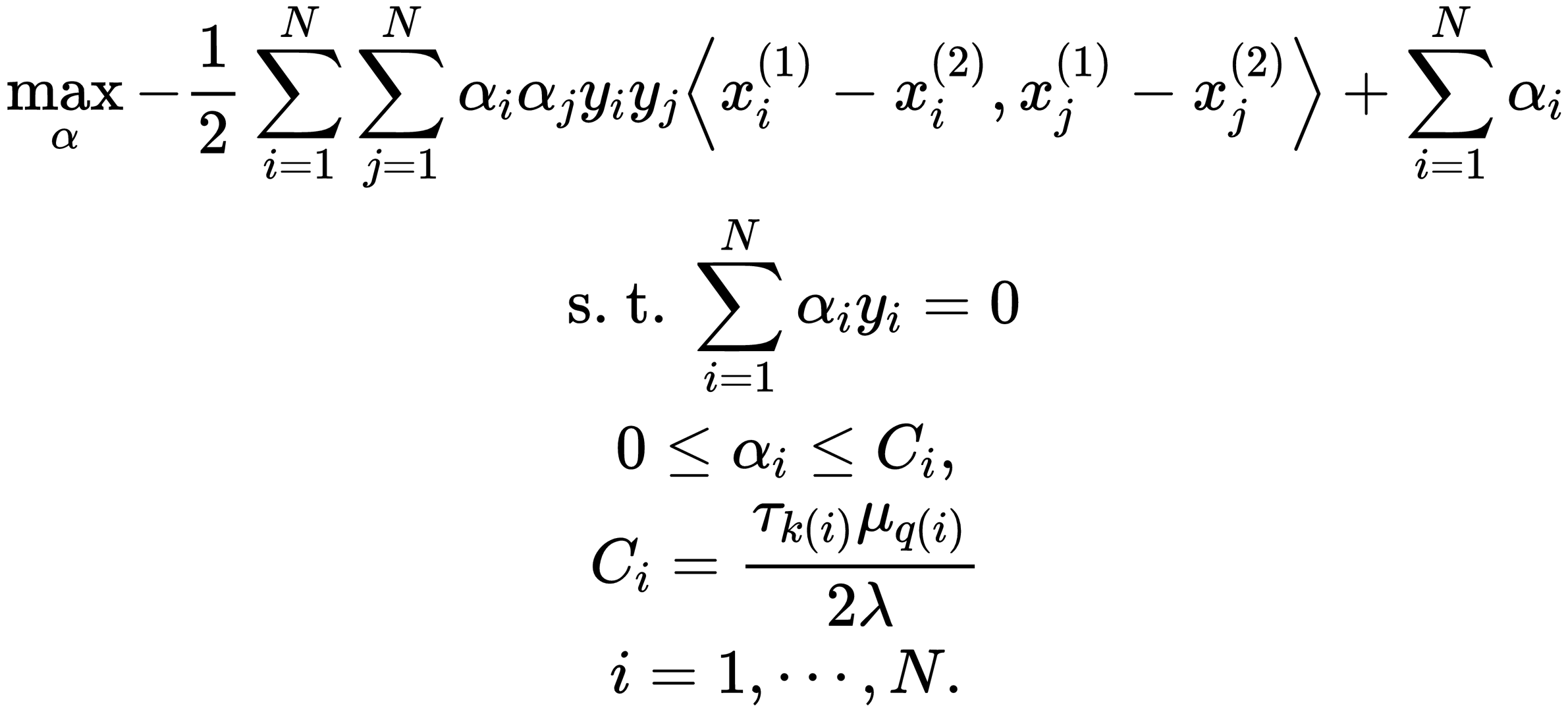
其中[x]+表示函数max(x, 0)， λ = 12C， τk(i)和µq(i)为权重。参见Ranking SVM的损失函数(4.2)。

其中τk(i)表示实例i的权重，其标签对属于第k级对。Xu等人提出了一种确定τk值的启发式方法。该方法以随机改变属于等级对的文献位置时平均减少NDCG@1为τk值。此外，µq(i)表示查询q对实例i的归一化权重，µq(i)的值简单计算为1nq，其中nq为查询q的文档对数。

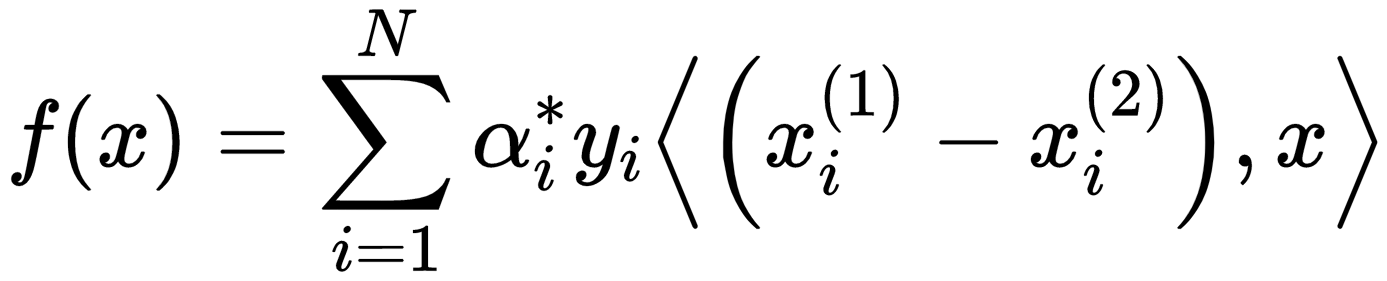
等效约束优化(二次规划)问题如下:



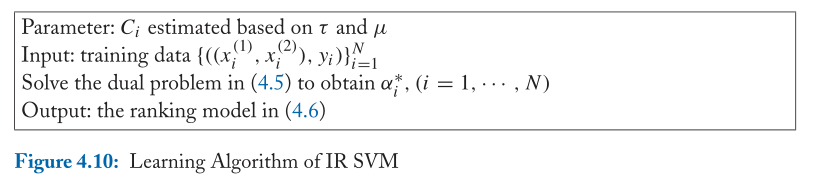
原始问题可以通过解决对偶问题来解决。



最优解的形式如下:



IR SVM的学习算法如图4.10所示。

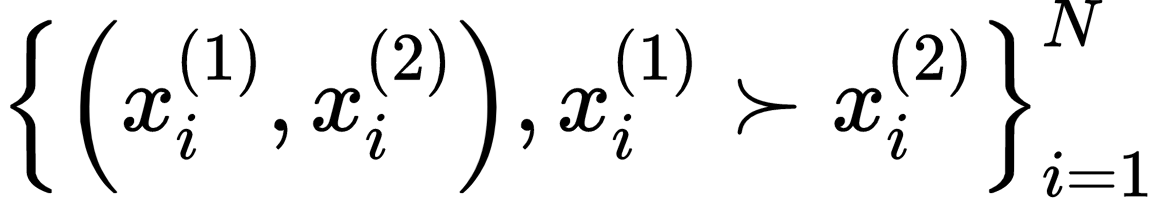


**4.5 GBRANK**

Zheng等人[114,115]提出的GBRank也是一种基于Boosting Decision Tree的两两方法。

**4.5.1损失函数**

GBRank采用偏好对作为训练数据

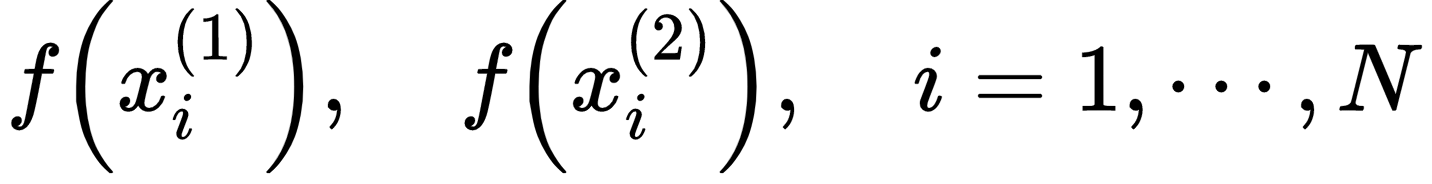


在GBRank中，使用以下两两损失函数。

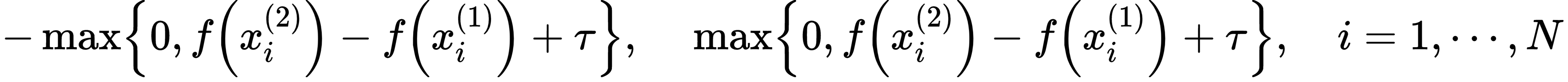


其中f (x)为排序函数，τ(0 < τ≤1)为参数。注意，假设x(1) i % x(2) i保持不变。对于损失函数的直观解释是，在τ时，如果f (x(1) i)大于f (x(2) i)，则损失为零;反之，损失为12(f (x(2) i)−f (x(1) i) + τ)2 (cf.，排序支持向量机(4.2)中的损失函数)。

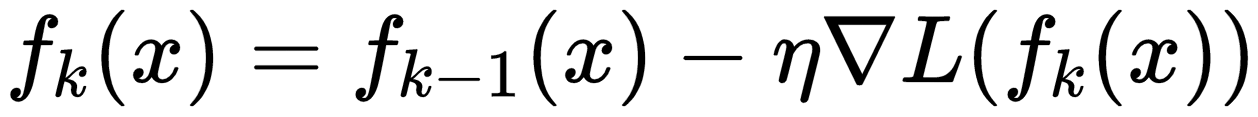
我们可以使用函数梯度体面优化损失函数对训练实例。首先,我们认为



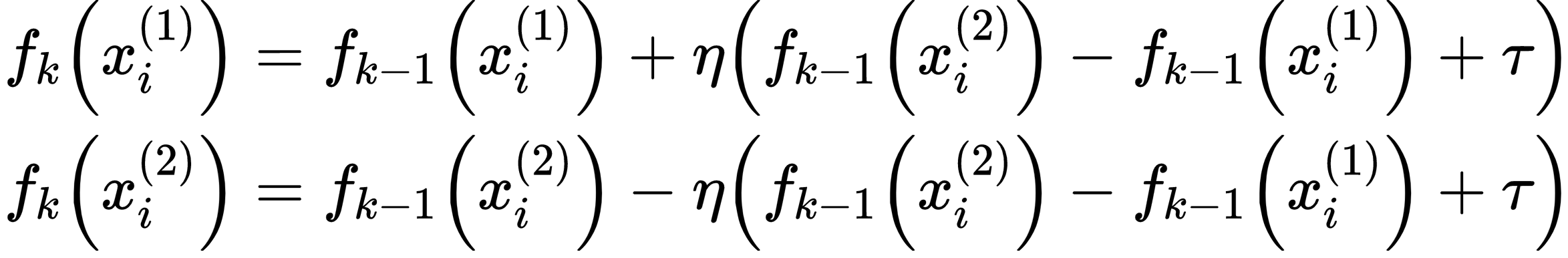
作为变量，计算L(h)相对于训练实例的梯度



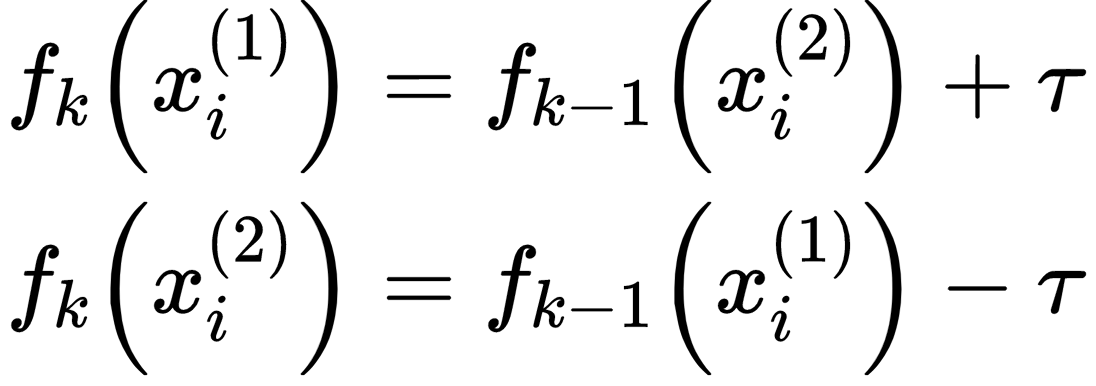
当f (x(1) i)−f (x(2) i)≥τ时，对应的损失为零，且不需要改变排序函数。如果f (x(1) i)−f (x(2) i) < τ，对应的损失是非零的，我们使用梯度下降改变排序函数



在哪里!表示梯度，η表示学习速率。更具体地说,

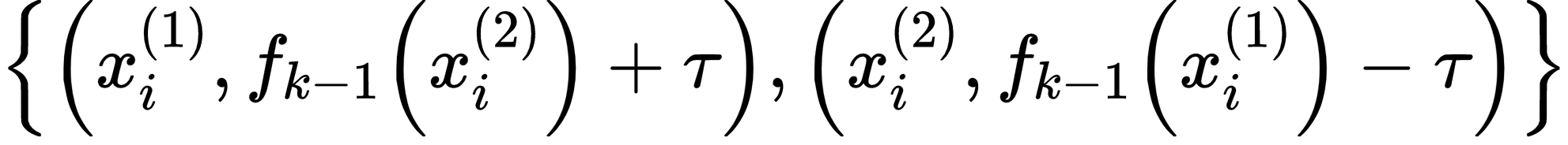


其中fk(x)和fk−1(x)分别表示f (x)在第k次和(k−1)次迭代中的值，η为学习速率。如果η等于1，那么我们只需要按以下方式更新函数(第k次迭代)。

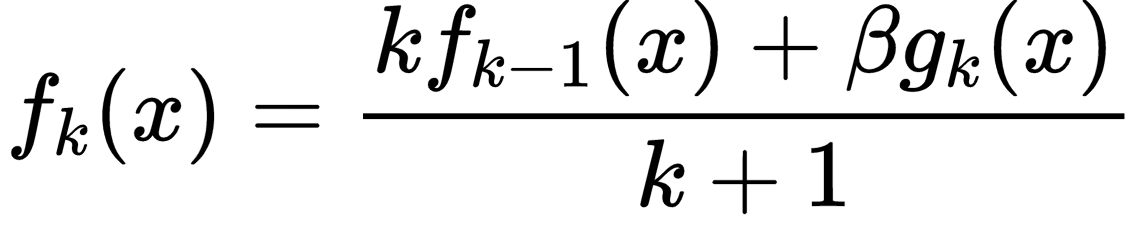


**4.5.2学习算法**

GBRank收集所有具有非零损失的对(在第k次迭代中)

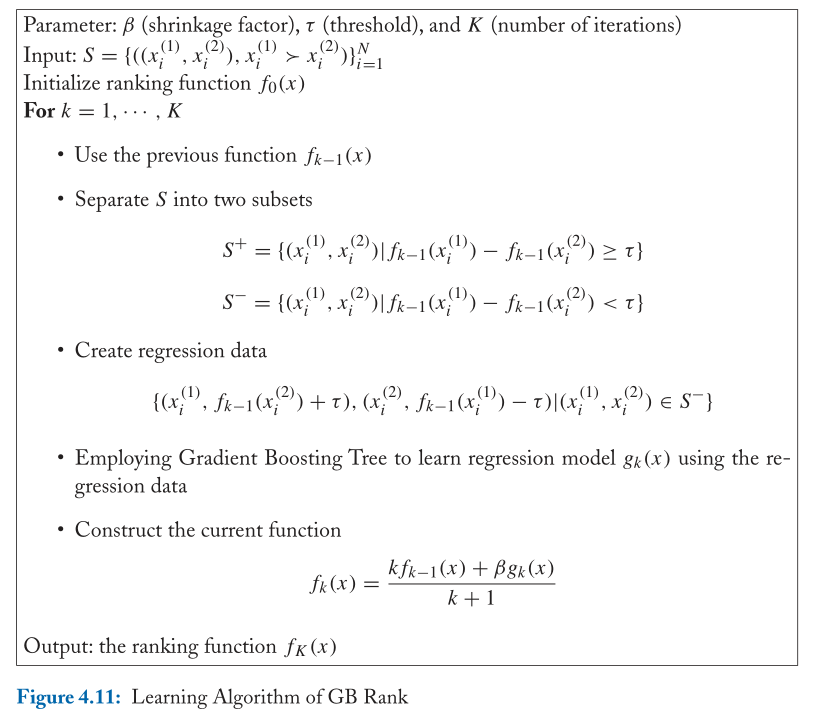


将其视为回归数据，使用Gradient Boosting Tree[38]学习可以对回归数据进行预测的模型gk(x)。然后将学习到的模型gk(x)与现有模型f(k−1)(x)线性结合，创建一个新的模型fk(x)(在第k次迭代中)



式中β为收缩因子。

GBRank算法如图4.11所示。



**4.6 RANKNET**

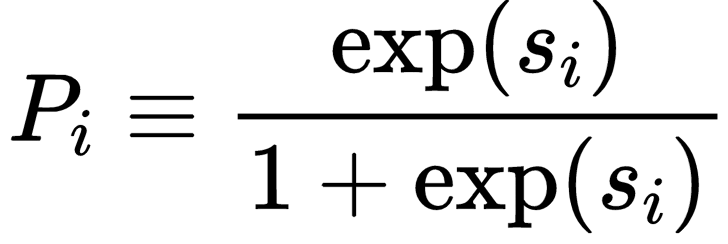
Burges等人开发的RankNet .[11]也是一种两两比较的方法。

**4.6.1损失函数**

RankNet采用交叉熵作为学习的损失函数。

首先，假设在训练数据中，每对对象都有一个概率。对于对象对(文档对)x(1) i和x(2) i，概率¯Pi给定，表示x(1) i领先于x(2) i的概率(例如，x(1) i的等级高于x(2) i)。例如，¯Pi = 1表示x(1) i肯定在x(2) i前面。¯Pi = 0.5表示无法确定哪个在哪个前面(例如，它们属于同一个等级)。

其次，假设使用排序函数计算每对对象的概率。对于对象对(文档对)x(1) i和x(2) i，计算概率Pi。假设排序函数f: $d→$给对象分配分数。让年代(1)我= f (x(1)),(2)我= f (x(2)),如果=(1)我−年代(2)。然后我们定义



如果f (x(1) i) > f (x(2) i)，则x(1) i以Pi的概率排在x(2) i前面。交叉熵，它测量两个概率分布之间的“距离”被定义为



我们利用交叉熵作为损失函数，对一对物体的阶数进行预测。将(4.7)代入(4.8)产率

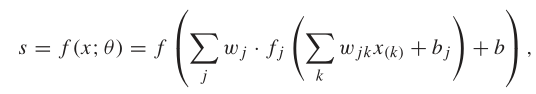


当¯Pi = 1时，交叉熵损失为logistic损失



**4.6.2模型**

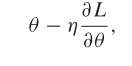
RankNet采用神经网络作为模型(图4.12)。这就是为什么这个方法被称为RankNet。神经网络应该是一个三层网络，只有一个输出节点，表示为



其中x(k)为输入x、wjk、bjk的第k个元素，fj为第一层的权值、偏移量、激活函数，wj、b、f分别为第二层的权值、偏移量、激活函数，s为最终的输出。θ表示参数向量。激活函数为s形函数(非线性函数)。

**4.6.3学习算法**

RankNet采用了Back Propagation算法(相当于随机梯度下降算法)来学习网络的参数。给定训练数据{(x(1) i, x(2) i)， Pi}， i = 1，···，n



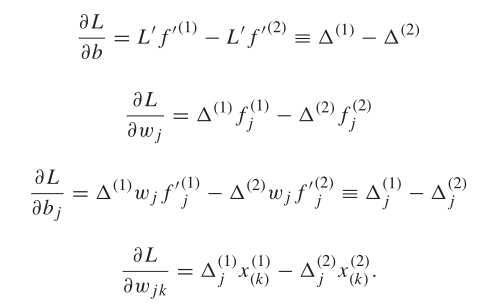
η表示学习速率。为简单起见，我们省略了表示迭代索引的上标。



偏好对的损失计算为L = L(s(1)−s(2))。损失L相对于参数θ的梯度∂L∂θ计算为

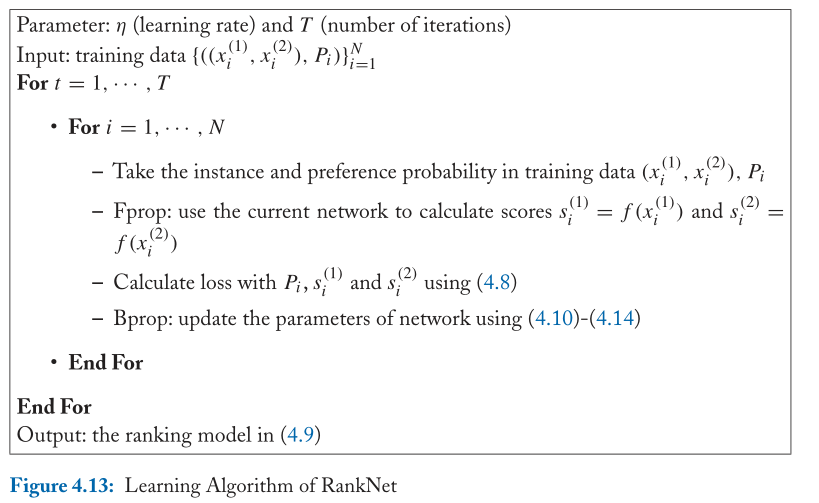


其中，L ' = dL(s) dsl ' = dL(s) dsl ' = dL(s) dsl ' = dL(s) dsl ' = dL(s) dsl ' = dL(s) dsl ' = dL(s) ds



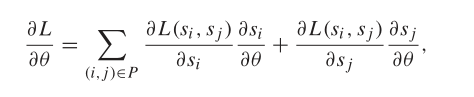
学习实际上是通过神经网络的向前和向后传播来完成的。Foreword propagation (fprop)对应的是根据新参数重新计算最终得分，backprop (backprop)对应的是重新计算参数。

RankNet使用一个验证数据集进行参数选择，即在参数选择上采用交叉验证。这样，任何红外测量都可以作为评价测量。Burges等人使用神经网络、交叉熵损失、反向传播和交叉验证作为RankNet方法的主要特征。



**4.6.4训练速度加快**

Burges等人也提出了一种有效的算法来加速RankNet[12]的训练过程。两种成分被用来加速。首先，该算法没有采用随机梯度下降法，而是采用批量梯度下降法。其次，它存储和重用一些中间结果，假设该算法被应用于可以利用querydocument结构的搜索。假设P是训练数据中出现的文档对(i, j)的集合。进一步假设D是所有文档的集合，Pi\_是P中{i, j}对的文档j的集合，P\_j是P中{i, j}对的文档i的集合。设m表示查询的数量，n表示每个查询的文档数量。该算法计算损失函数的梯度与参数对所有的训练数据



式中，L(·)为损失函数，θ为模型参数，si和sj为i和j该算法进一步将梯度改写为



在查询的每次迭代中，将执行n个fprops来计算最终的分数si。接下来，对于每个文档，(j∈Pi\_∂L(si,sj)∂si和(i∈P\_j∂L(si,sj)∂sj被计算和存储。然后，用n个fprops和n个backprops计算梯度∂si∂θ和∂sj∂θ。这样，∂L(si,sj)∂si和∂L(si,sj)∂sj的计算效率从O(n2)阶提高到O(n)阶，其中n是每个查询的文档数。

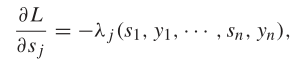
**4.7 lambdarank**

**4.7.1损失函数**

排序评价结果(学习中的目标函数)通常是不可连续可微的，取决于排序。(排序函数本身也不是连续可微的)。Burges等人提出的LambdaRank[12,32]考虑采用梯度下降法来优化评价结果，并尝试直接定义和利用评价结果的梯度函数。

假设给出了排序模型、查询和文档。然后，每个文档从排名模型中接收到一个分数，可以根据分数对文档进行排序，从而创建一个排名列表。由于文件被分配地面真实标签，可以得到基于IR测度的排序评价结果。假设我们使用一个替代损失函数L来近似IR评估测度。然后，可以得到基于替代损失函数L的评价结果。这正是LambdaRank试图不断优化的评价结果。

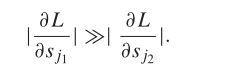
代理丢失函数是在文档列表上定义的。从这个意义上说，LambdaRank也可以被视为一种列表方法。LambdaRank没有明确给出损失函数的定义。相反，它定义了替代损失函数的梯度函数。更具体地说，梯度函数定义为



其中s1, s2，···，sn表示文档的分数，y1, y2，···，yn表示文档的标签。请注意，索引j位于单个文档上。也就是说，文档的梯度取决于其他文档的分数和标签。符号的选择使得文档的正值意味着文档必须减少损失。在当前模型为查询生成文档的排名列表之后，将计算文档的梯度。梯度函数被称为Lambda函数，这就是为什么这个方法被称为LambdaRank。

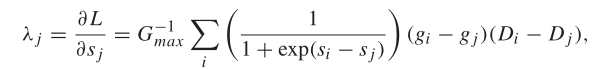
接下来的问题是如何指定Lambda函数，从而有效地优化排名评估结果。一个想法是增加文档在顶部位置的梯度。

假设有两个相关文档d1和d2。一个在位置2，另一个在位置n - 2。d1上移到最顶端的分数变化应该小于d2上移到最顶端的分数变化。因此，我们更愿意花费一点点的容量来提高d1而不是花费大量的容量来提高d2。也就是说，d1的梯度(得分)应该远远大于d2的梯度(得分)通常，对于任何两个位于j1和j2两个位置的文档。假设j1 3 j2(即前一个文档的级别比后一个文档高得多)，我们将满足梯度



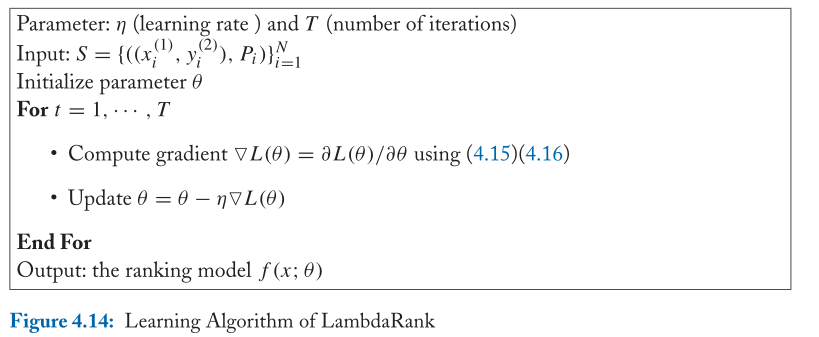
**4.7.2学习算法**

在实现LambdaRank时，我们只需要考虑一种计算Lambda函数(损失函数的梯度)的方法。一种常见的方法是基于NDCG计算它



其中gi、Di、si分别表示文档Di的增益、折扣和得分。G−1max为NDCG的归一化因子。Lambda函数实际上是一个两两损失函数。也就是说，“LambdaRank”的传统实现是一种两两配对的方法。

LambdaRank采用神经网络作为排名模型。事实上，它可以被看作是RankNet的延伸。学习算法与RankNet(快速版)相似，只是使用了不同的损失函数。图4.14总结了算法。



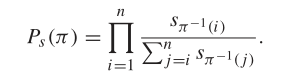
**4.8 listnet和listmle**

ListNet和ListMLE是Cao等人[14]和Xia等人提出的学习排序的概率方法和listwise方法[104]。这些方法利用了统计学研究中的Plackett-Luce模型。参见[44]。

**4.8.1 PLACKETT-LUCE模型**

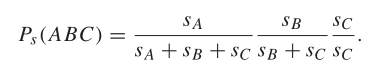
让我们先看看Plackett-Luce模型(简称PL模型)。PL模型定义了物体排列的概率分布，称为排列概率。假设有一组对象o = {o1, o2，···，on}。设π表示对象的排列(排序表)，π−1(i)表示在π中第i级(位置)的对象。进一步假设给这些对象分配了非负的分数。设s = {s1, s2，···，sn}表示对象的得分。

PL模型根据分数s定义置换π的概率如下:

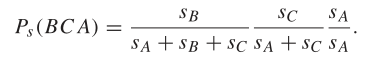


排列的概率自然地形成了概率分布。

例如，假设有三个对象A、B、C和sA, sB、sC是对象的分数(sA > sB > sC)。排列ABC的概率为



排列BCA的概率为



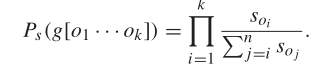
这很容易验证



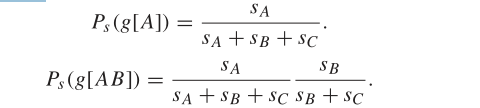
排列概率有以下解释，如下面的例子所解释。给定三个对象A、B和C以及它们的分数，我们随机地对它们生成一个排列。如果我们先根据A的相对分数在A、B、C中选择A，然后根据B的相对分数在B、C中选择B，最后选择C，那么我们就生成了概率为Ps(ABC)的置换ABC。置换概率Ps(ABC)表示在此过程中产生置换ABC的可能性。

PL模型有一些很好的属性。第一，分数降序排列的概率最大，分数降序排列的概率最小。在上例中，p (ABC)最大，p (CBA)最小排列概率。此外，给定以分数降序排列的排列，交换排列中的任意两个对象将降低概率。在上面的例子中，ABC中的B和C交换产生ACB，我们有Ps(ABC) > Ps(ACB)。

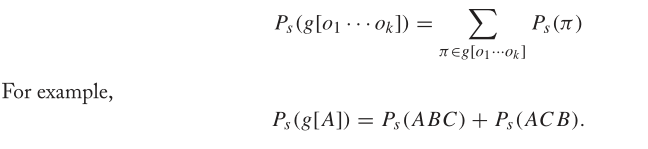
PL模型还定义了顶部k个子群的概率分布，称为顶部k概率。给定对象和对象的排列，我们可以在对象上定义top k个子组。上k个子群g[o1···ok]表示上k个对象为o1···ok的所有排列。子群g[o1···ok]的top k概率定义为



在上面的例子中，我们有



top k概率与置换概率之间的下列关系很容易验证成立，这是PL模型的另一个性质。

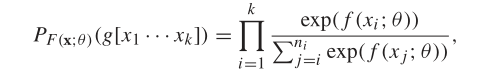


**4.8.2 LISTNET**

ListNet使用参数化的Plackett-Luce模型。该模型既可以基于排列概率，也可以基于top k概率，但出于效率考虑，通常基于top k概率。排列概率的计算时间复杂度为O(n!)阶，而第k个排列概率的计算时间复杂度为O(n!)阶。/ (n−k) !)。

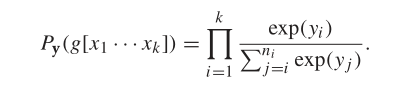
在文档检索中，假设对于查询q及其关联的文档d1、d2、···、dn，给定相应的相关标签y1、y2、···、yn。从查询q和文档d1, d2，···，dn，创建特征向量x1, x2，···，xn。

给定特征向量x1, x2，···，xn，子群g[x1···xk]的top k概率可计算为

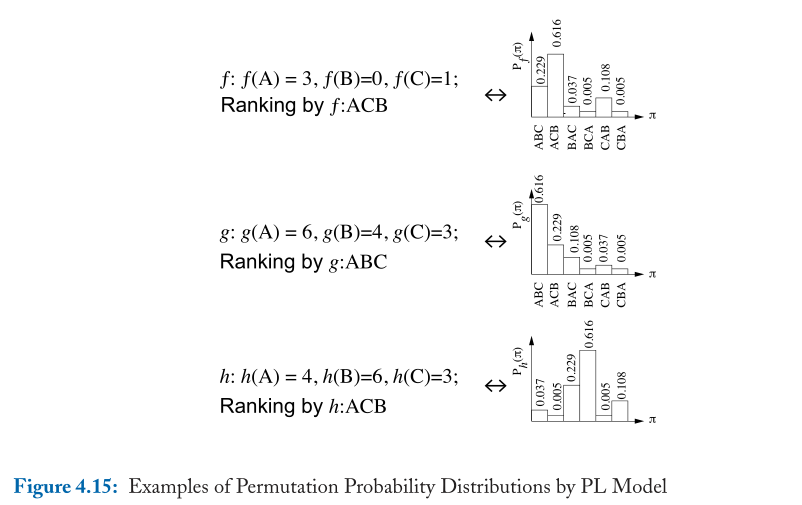


f (x);θ)是参数θ和F (x;θ)是一个由神经网络给出的分数列表。也就是说，xi的分数是由神经网络的一个指数函数决定的，它可以作为排名模型。

对应的标签y1, y2，···，yn也可以转换为分数。具体来说，xi除以yi的分数可以由yi的指数函数来确定。然后用类似的方法计算标签的前k个概率。



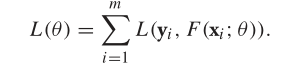
如果两个评分函数在排名上有相似的效果，那么它们的排列分布(或前k个概率分布)应该在形状上相似。图4.15给出了基于两个评分函数f和g的两个排列分布。可以使用KL发散来衡量两个评分函数之间的差异。这正是ListNet的理念。



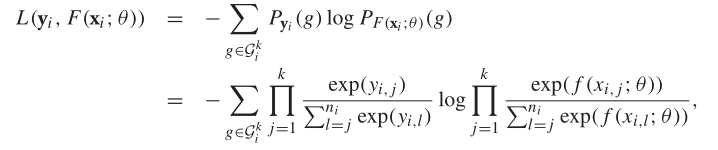
ListNet通过神经网络模型测量top k概率，通过KL发散测量ground truth的top k概率。两个概率分布之间的散度定义为D(P ||Q) = (i pi log piqi，其中P和Q是两个概率分布。这里只使用了(i - pi log qi)，因为(i pi log pi是常数。(请注意KL散度是不对称的)。

假设训练数据S = {(xi, yi)}mi=1。(xi, yi)表示为((xi,1, xi,2，···，xi,ni)， (yi,1, yi,2，···，yi,ni))。基于局部排序函数定义全局排序函数:F (xi) = (F (xi1)， F (xi2)，···，F (xini))。

ListNet将所有训练实例的KL发散作为总损失，通过最小化总损失来学习排名模型。总损失函数定义为

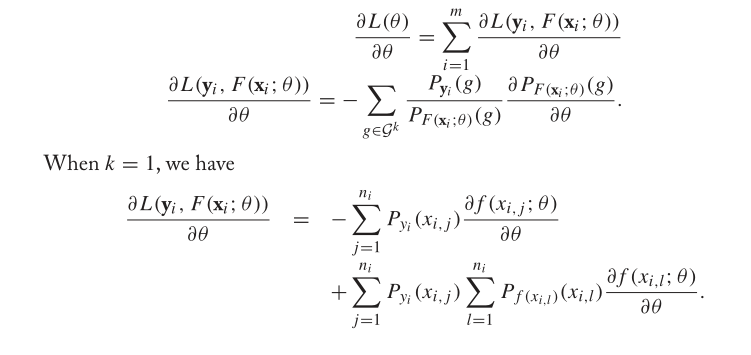


这里每个实例的loss函数定义为



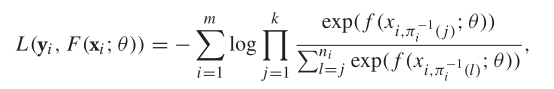
式中Pyi (g)通过基态yi表示子群g的上k个概率，PF (xi;θ)(g)通过神经网络F (xi;θ)， Gki表示所有上k个子群。

ListNet使用梯度体面来执行优化。图4.16给出了学习算法。

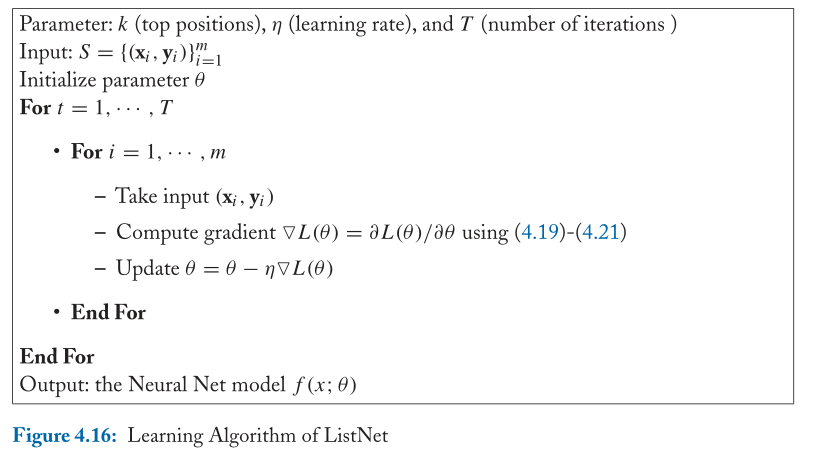


**4.8.3 LISTMLE**

另一种算法是ListMLE，它采用参数化的Plackett-Luce模型(4.17)(4.18)和极大似然估计。在对数损失的基础上最大化以下总损失函数



πi是一个完美的排名。ListMLE的学习算法与ListNet的学习算法相似。注意，当k = 1时，ListMLE退化为Logistic回归。



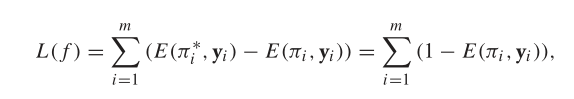
**4.9 ADARANK**

由于IR中的评价指标是基于列表的，所以直接优化学习中的列表损失函数进行排序更自然、更有效。AdaRank是Xu和Li[108]提出的方向优化算法之一。

**4.9.1损失函数**

假设训练数据表示为特征向量的列表及其对应的标签(等级)列表S = {(xi, yi)}mi=1。我们要学习排名模式f (x)上定义的对象(特征向量)x。给出一个新的列表的对象(特征向量)x,学习排名模型可以将分数分配给每个对象x, x∈x。我们可以根据分数排序对象生成一个排名π(排列)。评价是在列表级进行的，具体来说，使用了一个列表级评价测度E(π， y)。

在学习中，理想情况下，我们会创建一个排名模型，它可以最大化训练数据的列表评估方法的准确性，或者等价地，最小化下面定义的损失函数，



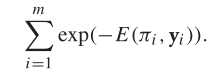
其中πi是通过排序模型f对特征向量xi的排列，yi是相应的等级列表。我们将损失函数L(·)称为“真实损失函数”(或“经验风险函数”)，而那些设法将真实损失函数最小化的方法称为“直接优化方法”。

由于损失函数不光滑且不可微，因此直接的优化评价可能行不通。相反，我们可以考虑优化损失函数的上界。

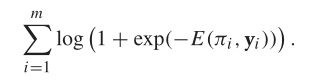
因为不平等



对，我们可以考虑优化下面的上界



其他上界也可以考虑，例如，



也就是说，指数函数和逻辑函数可以作为学习中的替代损失函数。注意，这两个函数对于E都是连续的，可微的，偶凸的。

**4.9.2学习算法**

AdaRank通过推进方法使指数损失函数最小化。AdaRank模仿著名的AdaBoost算法[36]，对指数损失函数进行逐步最小化。更具体地说，AdaRank重复重新加权训练实例，创建一个弱秩，并分配一个权重给弱秩的过程，以最小化损失函数。最后，AdaRank将弱秩线性组合作为排序模型。AdaRank算法如图4.17所示。

我们可以证明，在一定条件下，AdaRank能够在训练过程中不断减小经验损失函数，如[108]所示。当评价测度为点积时，AdaRank可以将损失减小到零。

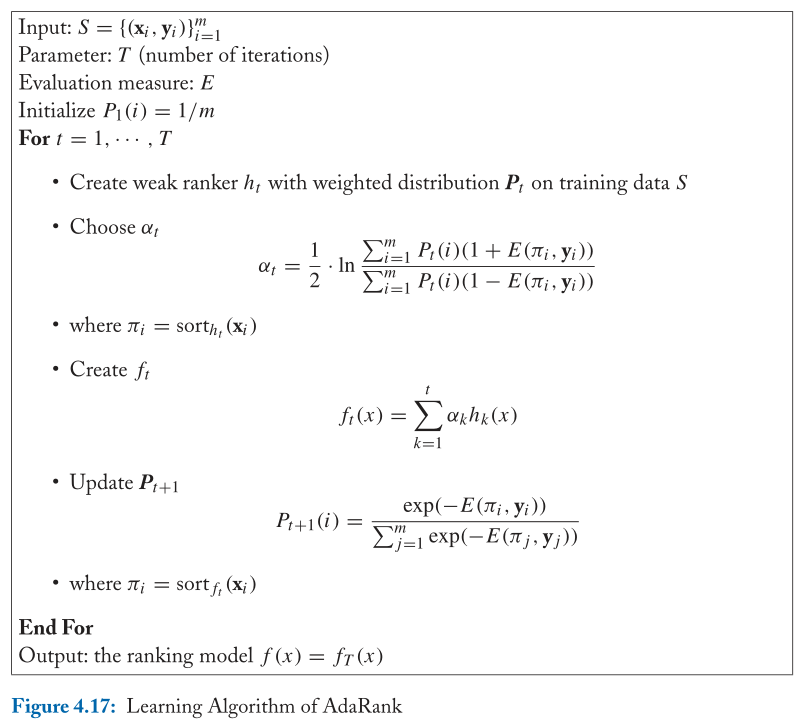
AdaRank的一个优点是简单，它可能是最简单的学习排序算法之一。

**4.10支持向量机地图**

另一种直接优化方法尝试使用结构支持向量机技术来学习排名模型。Yue等[111]开发的SVM MAP算法就是这样一种算法。Xu等[110]进一步将其推广到包括PermuRank在内的一组算法中。参见[15,62]。

**4.10.1损失函数**

在排序中，对于查询qi，排序模型f (xij)给每个特征向量xij打分，其中xij是由qi及其关联文档dij派生出来的特征向量。特征向量xi



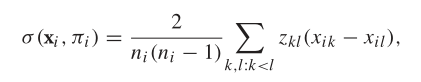
(文档di)根据其分数进行排序，得到标识为πi的排序。特征向量xi的标号也表示为yi。为简便起见，假设排名模型f (xij)是一个线性模型:



其中w为权向量。我们考虑使用一个评分函数S(xi， πi)来衡量一个给定的排列(排序)πi的优度。S(xi， πi)定义为



其中w仍然是权向量，向量σ (xi， πi)定义为

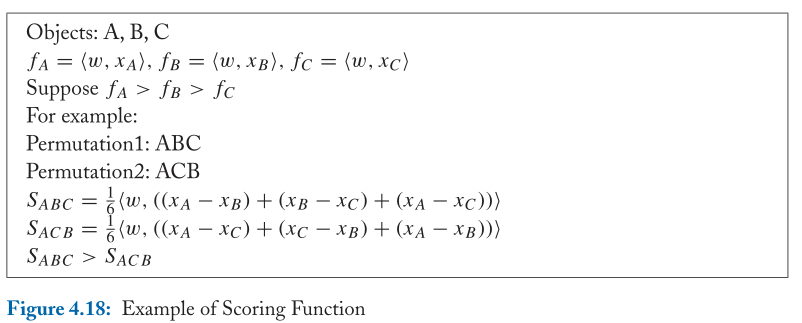


其中，当πi(k) < πi(l) (xik在πi前面)，则ZKL = +1，否则为−1。我们可以在学习中使用评分函数。对于查询qi，我们对每个排列πi计算S(xi， πi)，并选择得分最大的排列\_ πi:

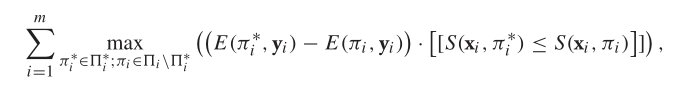


其中，#i表示xi的所有可能排列的集合。

可以很容易地看出，由式(4.24)选择的ranking (πi)等于由ranking模型f (xij)创建的ranking(当两者都是线性函数时)。图4.18给出了一个例子。很容易验证f (x)和S(xi， π)都将输出ABC为最优排序(排列)。



这样，我们可以将学习排序模型的问题看作是以下损失函数最小化的优化问题。



其中[[c]]为1，如果满足条件c;否则，它是零。π∗i∈#∗i⊆#i表示气的任何完美排列。

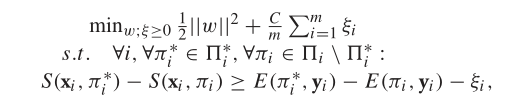
损失函数度量的是当排序模型的最优排序不是最优排序时的损失。可以证明(4.22)中的真实损失函数是(4.25)中的新损失函数的上界。

损失函数(4.25)仍然是不可连续可微的。我们可以考虑使用损失函数(4.25)的连续、可微、偶凸上界。

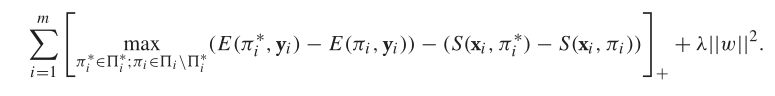
1)(4.25)中的0-1函数可以用其上界代替，如铰函数、指数函数、logistic函数



其中[x]+表示函数max(0, x)。2)max函数也可以用它的上界，即和函数来代替。这是因为(i xi≥maxi，如果xi≥0对所有i都成立。3)松弛1和2可以同时应用。例如，利用铰链函数，将真实损耗作为MAP，得到支持向量机MAP。更准确地说，SVM MAP解决了以下优化问题:

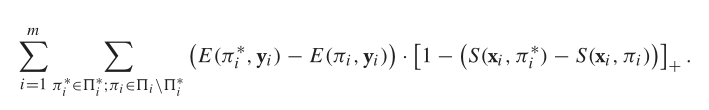


其中C为系数，ξi为查询qi的所有排列损失中最大的损失。同样，SVM MAP最小化以下正则化铰链损耗函数

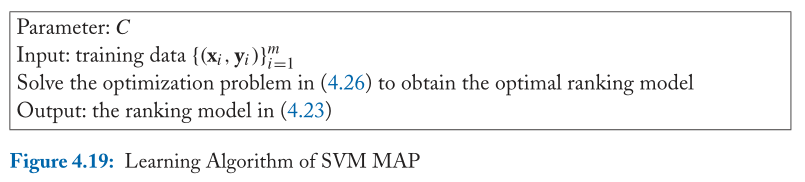


直观地说，在为每个查询选择最佳排列时，第一项计算总最大损失。具体来说，如果分数S(xi， π∗i)−S(xi， πi)之间的差值小于相应的评价测度E(π∗i, yi)−E(πi, yi)之间的差值，则存在损失，否则不存在。接下来，为每个查询选择最大损失，并对所有查询求和。我们也可以考虑该方法的NDCG版本，其公式类似。

由于c·[[x≤0]]< [c−x]+对所有c∈$+和x∈$都成立，很容易看出(4.27)的上界也限制了(4.22)中的真实损失函数。实际上，可以派生出许多优化上界(代理损失函数)的算法。Xu等人给出了一个他们称之为PermuRank的例子，他们表明PermuRank可以和SVM MAP一样好。PermuRank将下列正则化铰损函数最小化。



由于可能的排列数目是指数级的，因此直接实现SVM MAP和PermuRank是不可行的。SVM MAP和PermuRank都利用了一个工作集来解决这个难题。该集合在开始时包含任意的完美和不完美排名，每一轮学习时将最违犯的完美和不完美排名添加到集合中。图4.19总结了SVM MAP的学习算法



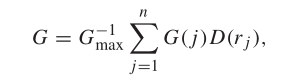
**4.11 SOFTRANK**

SoftRank是Taylor等人提出的一种直接学习排序的优化方法[43,95]。由于IR中的排名评价结果通常不是平滑的、不可微的，SoftRank尝试对排名评价结果的概率近似进行优化。其中，引入了一种近似于NDCG的软NDCG，根据软NDCG优化排名评价结果，并采用神经网络模型和梯度下降来完成学习任务。

**4.11.1 SOFT NDCG**

让我们先看看软NDCG的定义。为了便于解释，假设要对每个查询排序的文档数量相同，并且等于n。

我们重写NDCG的定义，并认为NDCG在位置n的排列(排序)π为

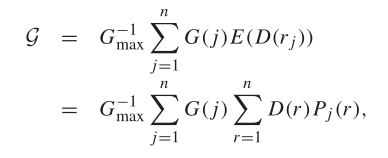


其中G(j)表示文件j的增益，rj表示文件j的秩(位置)，D(rj)表示文件j的位置折扣。

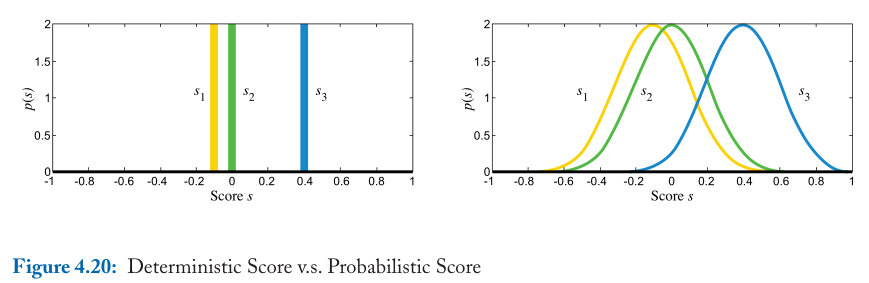
假设每个文档j都有一个得分sj (j = 1，···，n)，然后根据得分对文档进行排序，得到文档的排序结果。使用上面的定义可以计算出排名的NDCG值。NDCG的排序评价结果由评分和排序决定，具有非光滑性和不可微性。

在Soft NDCG的计算中，我们假设文档的排名是基于概率而不是确定性的文档得分来决定的。我们可以计算每个文档被排在一个位置的概率和每个文档的位置折扣的期望。该方法可以将基于NDCG的评价结果近似为基于Soft NDCG的评价结果，避免了排序的使用。

具体来说，软NDCG定义为



其中E(D(rj))表示文件j的头寸折现的期望，Pj (r)表示文件j处于r位的概率。接下来的问题是如何计算Pj (r)的概率分布和E(D(rj))的头寸折现的期望。

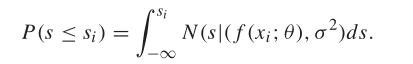


**4.11.2等级分布的近似**

可能有不同的方法来估计文档处于某个位置的概率。

对于给定的文档，SoftRank通过在n−1次伯努利试验(有不同的成功概率)中递归计算该文档排在其他n−1个文档前面或后面的概率，来计算该文档被排在每个位置的概率。

SoftRank假设模型f (xi;θ)服从高斯分布N((f (xi;θ)， σ 2)，已知方差σ 2(参见图4.20)。



因此，对于任意两个文档xi和xj，其得分之差服从高斯分布N((f (xi;θ)−f (xj;θ)、2σ 2)，则xi排在xj前面的概率为



SoftRank以递归的方式计算文档j在秩为r上的概率分布Pj (r)。假设它要在n个文档中定位文档j。首先，只有一个可用的rank，即1，文档j的rank是1。文档j的初始秩分布P (1) j(1)定义为



接下来，假设将剩余的n−1个文档逐个添加到秩分布中。当秩分布中有i−1个文档时，添加文档i，有两种可能的结果。文档我的分数大于文档j的分数,因此文档我排名前文档j。或者,文档我的分数小于文档j的分数,因此文档我排在后面的文档j。在前一种情况下,第j个文档的秩为r的概率等于第j个文档在前一次迭代中秩为r−1的概率。在后一种情况下，文档j的秩为r的概率与前一次迭代相同。这两种情况可以线性组合，可以定义第i次迭代时文档j的秩分布P (i) j (r)为



在计算过程中，假定



最后，定义第j个文档处于第r位的概率为

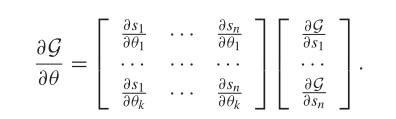


这样，每个文档都有一个等级分布，如图4.21所示。注意，这个分布是真实等级分布的近似。

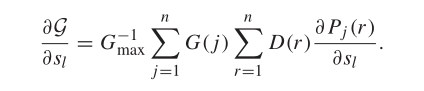


**4.11.3学习算法**

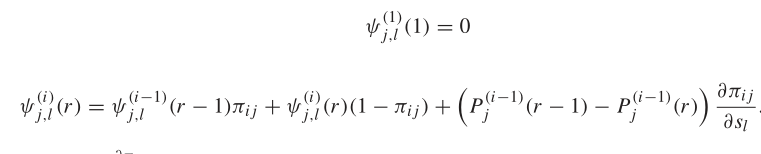
SoftRank采用神经网络作为模型，梯度下降作为优化技术。假设模型中有k个参数。在学习过程中，SoftRank计算n个文档的Soft NDCG相对于参数的梯度



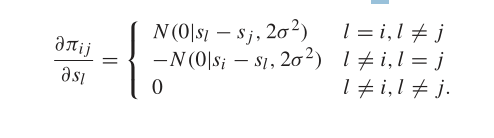
Soft NDCG对文档j得分的梯度计算为(l = 1，···，n)。

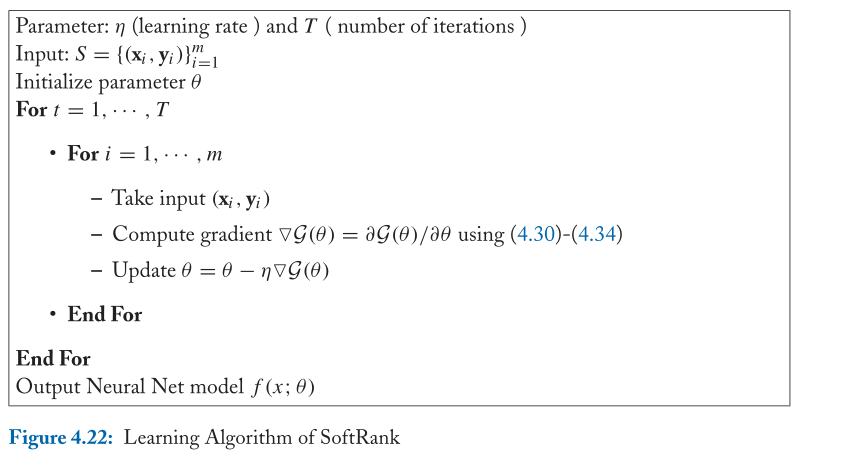


由于Pj (r)是一个递归定义的函数，它的导数也需要递归计算。表示ψj,l(r) =∂Pj (r)∂sl，我们递归地计算导数如下:



此外，∂πij∂sl可以在三种情况下计算(注意i 0= j)。

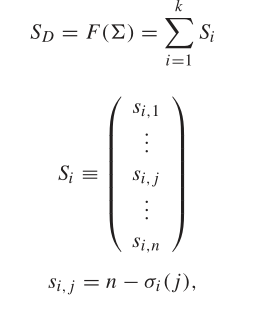




**4.12 BORDA点数**

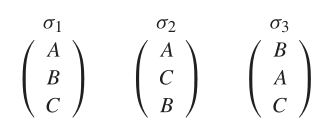
Borda Count是一种对聚合进行排序的无监督方法。Aslam & Montague[8]建议在元搜索中使用Borda Count。在这种情况下，Borda Count根据文档在基本排名中的位置对文档进行最终排名。更具体地说，在最终的排名中，文档是根据基本排名中排在它们下面的文档数量进行排序的。如果一个文档在许多基本排名中排名靠前，那么它在最终排名列表中也会排名靠前。

在最终的排名SD中，文档的排名得分为

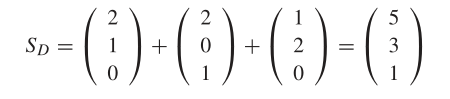


其中si,j表示在基本排序σi中排在第j位的文档个数，σi(j)表示在基本排序σi中第j位的文档个数，n表示文档个数。

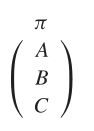
例如，文档A、文档B、文档C按照3个基本的排序:σ1、σ2、σ3。



文档SD的得分排序如下:



最终的排名列表π是由Borda Count根据分数SD创建的。

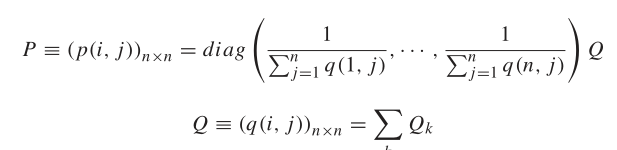


Borda Count可以被看作是给每个文档赋值k个向量(即n−σi(j))的一种方法，并根据向量的L1范数对文档进行排序。人们可以很容易地想出其他的替代方法，例如，根据向量的中位数或向量的Lp范数进行排序。这导致了几种不同的方法。

**4.13马尔可夫链**

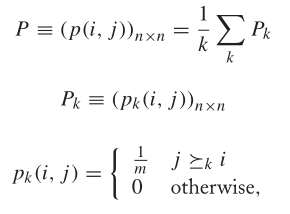
排序聚合的马尔科夫链方法，简称马尔科夫链，假设待排序的文档存在马尔科夫链，基本排序中文档之间的偏好关系表示马尔科夫链中文档之间的转换。然后利用马尔可夫链的平稳分布对文档进行排序。Dwork等人[34]提出了四种构造马尔可夫链转移概率矩阵的方法(记为MC1、MC2、MC3、MC4)。

MC1的定义如下:如果当前状态是文档我,然后选择下一个状态一致的文档集的排名高于或等于我基本的排名,也就是说,从多重集∪k {j | 7 k我},在j 7 k我意味着排名高于或等于我排名k。转移概率矩阵定义如下。



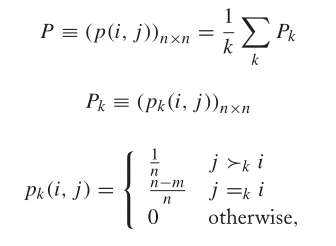


MC2的定义如下:如果当前状态为文档i，则下一个状态的确定方法是:首先从所有排名中统一选择一个基本排名σk，然后从排序高于或等于i的文档集合中统一选择文档j: {j|j 7k i}。



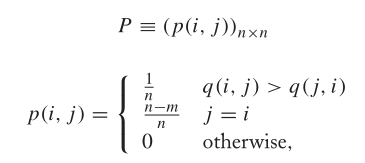
其中，m = |{j|j 7k i}|。

在MC3中，如果当前状态是文档i，则下一个状态如下所示。首先，我们从基本排序中统一选择排序σk，然后对于文档j，如果j %k i，则我们再到j;否则，我们保持在i。

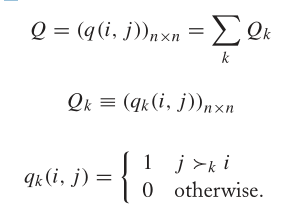


其中，m = |{% |} % b2。

在MC4中，如果当前状态是文档i，则决定下一个状态如下所示。文档j是从所有文档的联合中统一选择的。如果j %k i在大多数基本排名中是成立的，那么我们就去j;否则，我们保持在i。



其中m = |{j|q(i, j) > q(j, i)}|

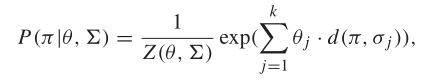


**4.14起动**

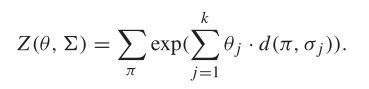
上述无监督方法在其最终排名决定中进行多数投票。事实上，这些方法平等地对待所有的基本排名列表，并给那些在大多数基本排名列表中排名靠前的文档打高分。然而，统一的重量假设在实践中可能不成立。例如，在元搜索中，不同的搜索引擎生成的排名列表可能具有不同的准确性和可靠性。人们可能想要学习基本排名列表的权重。黎巴嫩和Lafferty[63]提出的监督学习方法可以解决这个问题。

**4.14.1模型**

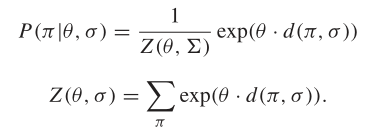
曲柄操作采用以下概率模型



其中π表示最终排名，“= (σ1，···，σk)表示基本排名，d表示两个排名之间的距离，θ表示权重参数。例如，距离d可以是肯德尔Tau。此外，Z是所有可能排名的正常化因子，定义如下。

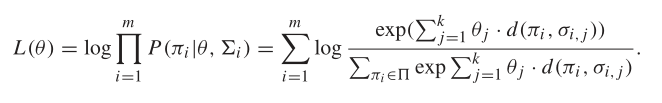


《Cranking》中的模型是统计学中的Mallows模型的扩展，在Mallows模型中只有一个“基本排名”。

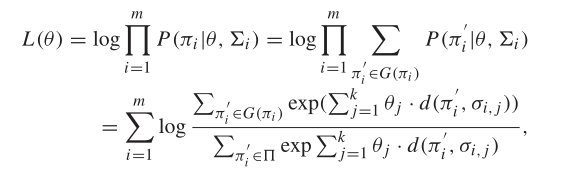


**4.14.2学习算法**

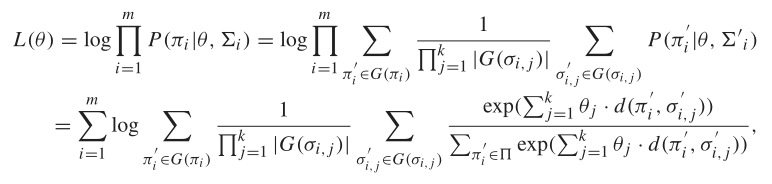
在学习过程中，训练数据表示为S = {("i， πi)}mi=1，目的是基于该数据建立排序聚合模型。我们可以考虑使用极大似然估计来学习模型的参数。如果最终排名和基本排名都是训练数据中的全排名列表，则计算log似然函数如下:



我们可以使用梯度下降来估计最优参数。在实践中，有时训练数据中只给出部分列表。如果最终的排名列表以部分列表的形式给出，则计算似然函数为



其中πi为偏集，G(πi)表示以πi为顶集的全表群。如果最终的排序表和基本的排序表都是部分排序表，则计算似然函数为

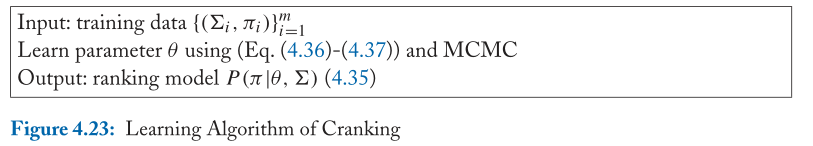


其中，πi是一个偏链表，G(πi)表示以πi为顶链表的全链表组，同样的，σi,j是一个偏链表，G(σi,j)表示以σi,j为顶链表的全链表组。此外，本文还假设G(σi,j)群的全表是均匀分布的。

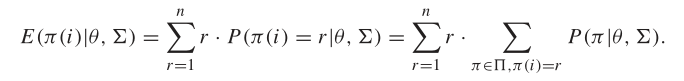
上述两种似然函数(式(4.36)-(4.37))不能直接优化。曲柄系统采用马尔可夫链蒙特卡罗(MCMC)进行参数估计。图4.23总结了学习算法。

**4.14.3预测**

在预测中，给出学习的模型(即参数θ)和基本排名”，Cranking首先计算最终排名π: P (π|θ，”)的概率分布。它使用了概率



分布来计算每个文档的期望等级。



然后，它根据文档的预期级别对其进行排序。

**排名学习的应用**

学习排序可以广泛应用于信息检索和自然语言处理领域。典型的应用有文档检索、专家搜索、定义搜索、元搜索、个性化搜索、在线广告、协同过滤、问题回答、关键字提取、文档摘要和机器翻译。

在应用程序中，要排序的对象(产品)可以是文档、文档单位(如句子和段落)、实体(如人和产品)。排名可以基于重要性、偏好和质量，可以作为端到端解决方案或解决方案的一部分使用。

本章介绍了一些学习排名(创建排名)的示例应用。

**网络搜索**

学习排名已经成功地应用到网络搜索中。众所周知，许多网络搜索引擎的排名模型都是通过学习排名技术建立起来的。模型中通常使用大量表示相关性的信号作为特征。培训数据是由一组专业的评委创建的。此外，采用了强大的计算平台，可扩展和高效地训练排名模型。

学习排名也应用于web搜索中的不同问题，包括上下文感知搜索[105]、近因排名[31]、联邦搜索[79]、个性化搜索、在线广告等。

**协作费尔增长率**

协同过滤，也称为推荐系统，其任务如下:用户被要求给这些项目打分。系统检查用户对物品的评级，并向每个用户提供物品排名列表。排名列表代表系统向用户推荐的内容，而排名越高的项目越有可能受到用户的青睐。

协同过滤可以被形式化为有序分类或分类问题，因为用户给项目打分。有时候，更自然的做法是将其形式化为排名(创造排名)。这是因为来自不同用户的评级是在不同的范围内，并且没有直接的可比性，因此最好将每个用户的评级视为一个排名列表。

Freund等[37]将RankBoost应用于协同过滤，特别是电影推荐。RankBoost是一种两两排序的方法，使用AdaBoost作为学习算法。Freund等人的方法将目标用户的评价作为训练数据，将其他用户的评价作为特征。使用训练数据创建RankBoost模型。然后使用训练后的模型对目标用户的所有电影(包括未分级的电影)进行排名。参见[46]。

**定义搜索**

在定义搜索中，给定一个术语查询，系统返回术语定义(定义段落)的排序列表。Xu等人提出了一种使用学习进行排序的定义搜索方法[107]。

该方法首先使用若干启发式规则自动从文档中提取所有可能的定义段落。例如，第一个句子为“X是a”的段落被当作候选段落。然后，他们的方法应用一个排名支持向量机模型，为所有候选段落分配代表它们成为良好定义的可能性的分数，删除冗余段落，并将这些段落存储在一个以术语为键的数据库中(例如，“X是a”中的X)。在定义搜索中，给定一个术语，系统检索相关的定义段落并返回定义段落的排名列表。

排名支持向量机模型利用了许多特征，包括积极和消极特征。例如，如果术语(例如，“X是a”中的X)反复出现在段落中，那么这段很可能是该术语的定义。如果像“她”、“他”或“说过”这样的词出现在段落中，那么这段很可能不是一个定义。

**关键词提取等**

关键字提取的问题如下。给定一个文档，输出一些短语(通常是名词短语)，它们可以精确而紧凑地表示文档的内容。传统的关键字提取是通过分类来形式化的，并采用了决策树和朴素贝叶斯等分类方法。Jiang等人将关键字提取问题形式化为排序而不是分类[55]。事实上，关键字提取可以看作是文档检索的反问题。

假设有一些训练数据，其中有一些文档被分配了关键短语和非关键短语。Jiang等人的方法将有序短语对作为训练实例，每个训练实例由一个关键短语和一个非关键短语组成，并利用训练数据建立一个Ranking SVM模型。然后，该方法使用训练过的模型对新文档的候选短语进行排序，并选择排名最高的候选短语作为关键短语。实验结果表明，排序支持向量机分类方法显著优于朴素贝叶斯分类方法。

Jiang等人给出了两个原因，说明排序方法比分类方法的性能更好。首先，在相对意义上考虑一个短语成为关键短语的可能性比在绝对意义上考虑这个短语的可能性更自然。其次，判断一个短语是否是关键短语的功能也是相对的。

**查询相关的总结**

当搜索系统向用户显示搜索结果时，显示文档的标题和摘要是很重要的，因为它们有助于用户判断文档是否相关。这就是所谓的依赖查询的摘要或片段生成问题。与查询相关的摘要通常包括两个步骤:相关句子的选择和摘要的生成。在句子选择中，识别出信息量最大的句子。

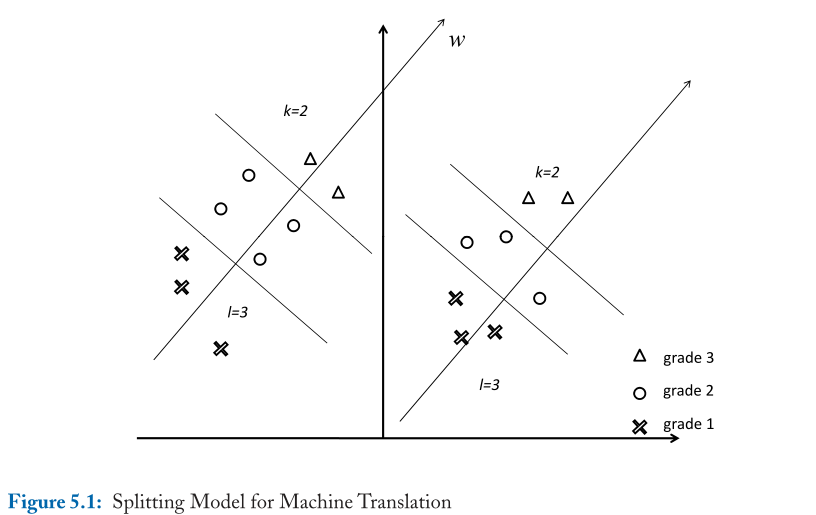
Metzler和Kanungo[76]提出在依赖查询的摘要中使用学习排序技术进行句子选择。他们将GBRank和支持向量回归应用到任务中。给定一个查询和一个检索到的文档，他们的方法将文档中的所有句子视为候选句子，并根据它们与查询的相关性和在摘要中作为一个句子的适宜性对这些句子进行排序。他们用一些有标签的数据训练一个排名模型。模型中定义了许多特性。例如，查询在句子中是否有精确的匹配、句子中查询词的部分、句子的长度(不推荐短句子或长句子)以及句子在文档中的位置。他们证明了GBRank是一种有效的算法。

**机器翻译**

机器翻译中的重排序也是一个典型的排序问题。目前最先进的机器翻译方法使用生成模型生成许多候选译文，然后使用判别模型对候选译文进行重新排序，然后选择排名靠前的译文。在重新排序模型中使用了能够区分好翻译和坏翻译的特征。

采用重新排名的方法有几个优点。首先，判别模型可以进一步利用全局特征和判别训练进行最终的翻译选择，从而提高翻译的准确性。第二，翻译的效率可以提高。首先用生成模型选出前n个候选翻译，然后从一小组候选翻译中选出最佳翻译。

例如，Shen at .提出利用学习排序技术对机器翻译进行重新排序。他们提出了两种类似于恶作剧算法的方法[93]。其中一种算法叫做分裂算法，它试图找到平行超平面，将每个句子的k和l预先确定的，顶部的k个好的翻译，底部的l个不好的翻译以及中间的翻译分开。图5.1展示了分裂模型。



**学习排名理论**

本章给出了排名学习(排名创建)的统计学习公式，并说明了排名学习理论研究中的问题。

**6.1统计学习公式**

学习排名(创建排名)是一个有监督的学习任务。设X为特征向量列表组成的输入空间，Y为成绩列表组成的输出空间。进一步假设x是x中的一个元素，表示一个特征向量列表，y是y中的一个元素，表示一个等级列表。设P (X, Y)是一个未知的联合概率分布，其中随机变量X取X的值，随机变量Y取Y的值。

假设F是一个从特征向量x映射到分数列表的函数。学习任务的目标是自动学习一个函数ˆF (x)，给定训练数据(x1, y1)， (x2, y2)，…(xm, ym)。每个训练实例由特征向量xi和对应等级yi (i = 1，···，m)组成，其中m为训练实例数。

F (x)和y可以进一步写成F (x) = [F (x1)， F (x2)，···，F (xn)]和y = [y1, y2，···，yn]。其中f (x)为局部排序函数，n为特征向量个数和等级。特征向量对应待排序对象，记为O =[1,2，···，n]。

我们利用一个损失函数L(··)评估的预测结果F (x)。首先,特征向量x的排名是根据F (x),然后排名结果进行评估与相应的分数y。如果成绩排名高的特征向量,然后将损失小。否则，损失将会很大。损失函数具体表示为

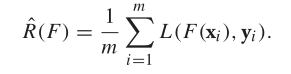


注意，排序的损失函数与其他统计学习任务中的损失函数略有不同，因为它使用了排序。

我们进一步定义风险函数R(·)为联合分布P (X, Y)下的期望损失函数，



给定训练数据，我们计算经验风险函数如下:



我们可以将学习任务形式化为经验风险函数最小化，就像在其他学习任务中一样。我们也可以引入正则化器来对正则化的经验风险函数进行最小化。

由于损失函数的性质(它不是连续的，并且使用排序)，最小化经验风险函数可能是困难的。我们可以考虑使用表示为的替代损失函数



相应的风险函数和经验风险函数定义如下:

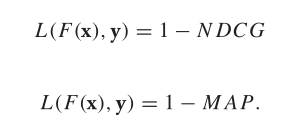


在这种情况下，学习问题就变成了基于代理损失的(正则化)经验风险函数最小化问题。

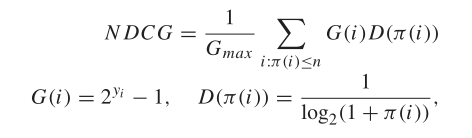
请注意，我们在这里采用了机器学习公式。在IR中，特征向量x来源于查询及其相关文档。等级y表示文档与查询的相关度。我们使用一个全局排序函数F (x)，在实践中，它通常是一个局部排序函数F (x)， x中可能的特征向量的数量可以非常大，甚至无穷大。然而，评估(损失函数)只涉及n个结果。在IR中，n可以由池策略确定(参见第2.2.2节)。

**6.2损失函数**

在二元分类中，真正的损失函数通常是0-1损失。相比之下，在排序中，真正的损失函数有不同的定义方式。在红外光谱中，真正的损失函数可以是基于NDCG(归一化折现累积增益)和MAP(平均精度)定义的函数。具体地说,我们有

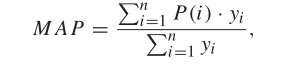


给定排列π乘以F (x)，它的NDCG(对于n个对象)定义如下:

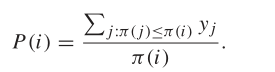


其中yi为对象i的等级，π(i)为对象i在π中的等级，G(·)为增益函数，D(·)为位置折现函数，Gmax为归一化因子。

给定置换π乘以F (x)，它的MAP(对于n个对象)1定义如下:



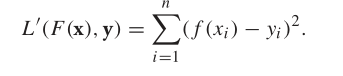
其中yi表示对象i取1或0作为值的等级，π(i)表示对象i在π中的等级，P (i)表示对象i到等级为止的精度，定义为



注意，真正的损失函数(NDCG loss和MAP loss)不是连续的，它们依赖于F (x)排序。

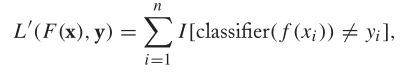
对于替代损失函数，也有不同的定义方法，这导致了不同的学习排序方法。例如，可以分别定义逐点损失、两两损失和列表损失函数。

平方损失，即逐点损失，定义为



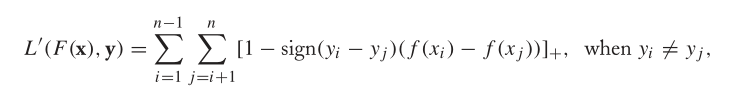
损失函数是在子集回归中使用的函数。

m兰克的逐点损失如下

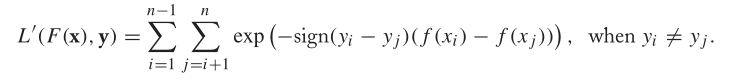


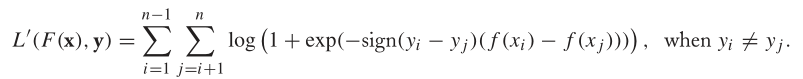
其中I[·]为指示函数，分类器(f (xi))的输出为一个标签(grade)。

两两损失可以是铰链损失、指数损失和逻辑损失，定义如下。它们分别用于对SVM、RankBoost和RankNet进行排序。



其中，当yi = yj时，假设L ' = 0。





Listwise损失可以是KL损失和对数损失，分别在ListNet和ListMLE中使用。

ListNet中的KL Loss定义为



其中D(·||·)为KL发散，Py(π)为y诱导的排列概率分布(或top k概率分布)，PF (π)为F (x)诱导的排列概率分布(或top k概率分布)，这两个分布均采用Plackett-Luce模型计算。

ListMLE中的对数损耗定义为



其中PF (π∗y)是由y完全排列的概率，由F (x)和PlackettLuce模型计算得出。

显然，AdaRank中的代理损失函数也是一种列表损失。



式中，NDCG由F (x)和y计算。

**6.3损失函数之间的关系**

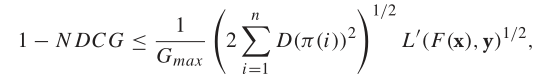
前人的研究表明，现有方法中的点态损失、成对损失(6.4-6.6)和表态损失(6.8)是真实损失(6.1-6.2)的上界。



这意味着现有的学习排序方法，如子集排序、m曲柄、排序支持向量机、RankBoost、RankNet、ListMLE和AdaRank都是优化不同替代损失函数的方法。

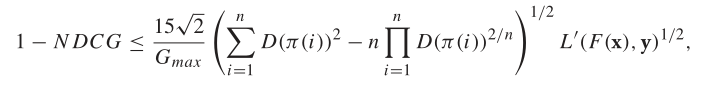
下面我们总结一下现有方法中使用的代理损失函数和真实损失函数(1-NDCG)之间的关系。

子集排名中的逐点损失函数为(1-NDCG)[29]的上界。



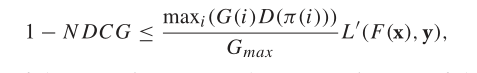
其中D(π(i))为对象i的位置折扣，L ' (F (x)， y)为子集排序中的代理损失函数。

m曲柄中的逐点损失函数是(1-NDCG)的上界[67]。



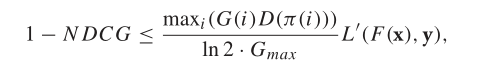
其中D(π(i))是对象i的位置折扣，L ' (F (x)， y)是m曲柄中的替代损失函数。

rank SVM、RankBoost和RankNet中的两两损失函数为(1-NDCG)[20]的上界。



其中G(i)为目标i的增益，D(π(i))为目标i的位置折扣，L ' (F (x)， y)为上述两两方法中的替代损失函数。

ListMLE中的listwise loss函数为(1-NDCG)[20]的上界。



其中G(i)是对象i的增益，D(π(i))是对象i的位置折扣，L ' (F (x)， y)是ListMLE中的替代损失函数。

**6.4理论分析**

关于学习排序的理论分析存在两个主要问题，即泛化能力和统计一致性。

方法的泛化能力表示经验风险函数与期望风险函数之间的关系。它通常由两个风险函数之间的边界表示。Cossock & Zhang给出了子集排序的泛化能力。Lan等人给出了Ranking SVM、IR SVM、ListNet和ListMLE的泛化界限[60,61]。最近，Chen等人在一个更自然的框架[21]中证明了两两方法的泛化界。

统计一致性是回答代理损失函数的优化是否能导致真实损失函数的优化的问题。Xia等人研究了ListNet和ListMLE的一致性[103,104]。

其他关于学习排名的理论分析工作见[2,5,25,28]。

**正在进行和未来的工作**

排序学习是机器学习及其相关领域的一个热点，包括信息检索、自然语言处理和数据挖掘，目前正在进行深入的研究。

第四章介绍了学习排序的方法。开发更先进的技术仍然是必要的。从第5章和第6章的讨论中也可以清楚地看出，关于学习排名的理论和应用，仍有许多有待解决的问题。

让我们来看看关于学习排名，特别是学习创建排名的一些正在进行和未来的工作。

•训练数据创建

•半监督学习和主动学习

•特征学习

•可扩展和高效的训练

•领域适应

•集成学习排序

•全局排序

•图中对象排序

**培训数据创建**

与其他机器学习任务一样，训练数据的质量在很大程度上影响学习排名的性能。如果训练数据的质量较低，那么训练后的模型的准确性也会较低。所谓的“垃圾进垃圾出”现象也出现在学习排名中。另外，降低训练数据构建的成本是需要考虑的另一个问题

在IR中，用于排名的训练数据通常是由人工注释的，这是昂贵且容易出错的。因此，训练数据的数量往往较少，数据的质量无法得到保证。

如前所述，应对挑战的一种方法是从点击数据中自动获得训练数据。点击数据代表了用户的隐式反馈，是训练数据创建的重要数据源。我们需要解决的问题是消除噪声和位置偏差。例如，可以采用Joachims[57]提出的方法来使用文档在搜索中的跳跃作为相对相关性判断的信号。Radlinski和Joachims[87]开发的另一种方法也可以被利用，它利用了搜索会话中的查询和点击。具体来说，在同一会话中，使用当前查询单击的文档比使用前一个查询检查但未单击的文档更可取，前提是用户可能没有找到与前一个查询相关的结果。

由于人类裁判标注的训练数据不可避免地包含错误，另一个相关的问题是如何自动纠正训练数据中的错误。一种方法是使用点击数据进行错误检测和修正。例如，Xu等人[106]提出了一种利用点击数据检测人类标注错误的方法。本文提出了一种从点击模式预测关联标签的判别模型。

有关训练数据创建的其他方法，请参见[4,7,88]。

**半监督学习和主动学习**

由于训练数据的创建是昂贵的，使用标记和未标记的数据学习排名自然成为一个重要的问题来研究。关键问题是如何利用无标签数据中的有用信息来提高学习性能。关于半监督学习已经提出了几种方法[6,33,49,56,66]。看来有必要对这个问题作进一步的调查。

另一个相关的问题是主动学习。Long等人[73]指出主动学习的一般原则，称为期望损失最小化(ELO)，可以在排序中使用，就像在分类、回归和其他任务中一样。ELO建议选择预期损失最大的查询或文档。他们提出了一种名为ELO-DCG的算法，用于在查询和文档级别进行主动学习。

**学习功能**

在实践中，排序模型中使用的特征对于学习排序的准确性更为关键。开发强大的功能是建立实用排名系统的重要一步。

在排名模型中，BM25和LM4IR(无监督排名模型)可以作为排名模型的特征。BM25和LM4IR实际上代表了查询和文档的相关性，使用它们的术语的匹配度。如何丰富匹配模型并从数据中学习匹配模型是一个有趣的话题。Metzler & Croft[75]提出使用马尔可夫随机场模型来表示查询与文档的匹配程度，并将该模型用于相关度排序。其关键思想是考虑查询中术语之间的依赖关系，并在概率依赖图(MRF)中表示它们的关系。文中还提出了一种学习MRF模型的算法。参见[96]。

PageRank是一个在学习排名时广泛使用的文档特性。我们也可以考虑增强模型。如Liu等人提出利用基于用户行为数据的用户浏览图，在图上构造连续时间马尔可夫模型，并计算平稳分布作为页面的重要性。他们的算法被称为BrowseRank，是PageRank的自然扩展[72]。

当然，我们还需要对排序特征的监督学习或非监督学习进行更多的研究。功能的自动选择也需要更多的研究。

**可扩展和高效的培训**

与其他学习任务一样，用于学习排名的训练数据也可能很大。如何使学习训练具有可扩展性和效率是一个重要的问题。Chapelle和Keerthi[16]开发了一种训练Ranking SVM的高效算法。他们使用原始的牛顿方法来加速训练过程，并表明他们实现的排序支持向量机比SVMLight(广泛使用的排序支持向量机学习工具)快5个数量级。

**领域适应研究**

领域适应或迁移学习是机器学习的一个热门研究课题，排名也是如此。另一个相关的问题是多任务学习。有些领域很容易获得训练数据并建立可靠的排名模型，而有些领域则不是这样。如何将一个在一个领域训练的模型适应到另一个领域就变得很重要了。

已经提出了领域适应、迁移学习和多任务学习的方法[9,17,19,40]。如Chapelle等人[17]提出了一种多任务排序的提升算法。他们的方法为几个不同的任务学习了一个联合模型，该模型用任务特定的参数来处理每个任务的细节，以及使用共享参数的任务之间的共性。

**综合学习排序**

为了提高排序的准确性，可以采用分治的方法。也就是说，对于文档检索中的不同查询，可以创建和利用不同的排名，并最大化总体排名的准确性。

在一般的网络搜索中，用户的搜索需求是非常多样化的，因此采用查询依赖的方法显得更加必要。如何将查询自动分类到类中，为每个类训练一个排名模型，并将排名模型组合起来成为一个重要的研究领域。Geng等人[41]提出了一种基于查询的排序方法。给定一个查询，该方法尝试找到k个最接近的训练查询，并利用邻域内的数据构建一个排序模型。给出了进行k近邻训练的有效方法。在这一方向上存在着挑战性而又有趣的问题。

**全球排名**

在创建排名时，通常使用一个局部模型。局部模型为每个对象分配一个分数，并根据其分数对对象进行排序。使用局部模型进行排名有一定的效果加工效率等优点。然而，正如在第一章中所解释的，排名的创建本质上是一个全球排名问题，因此最好学习和利用全球排名模型。

在文献检索中，不仅要根据相关性进行排序，还要根据多样性、新颖性等因素进行排序。Qin等[83]提出采用连续条件随机场模型进行全局排序。该模型将文档及其得分表示为顶点，并将文档得分之间的关系表示为无向图中的边。本文还提出了一种从有监督学习数据中学习CRF模型的方法。结果表明，CRF模式能有效利用文档之间的相似关系和链接关系。Yue等[112]提出了一种基于多样性的排序方法(对于像“Jaguar”这样的模糊查询，最好将所有主要感官的相关文档排在前面)。该方法将相关文档作为输入，然后将它们分组到不同的子集。它使用结构支持向量机将模型的训练形式化为一个学习问题。其他相关工作见[52,54,84,89]。

**图中节点的排名**

有时还可以获得关于要排序的对象之间关系的信息。这种关系通常用对象的有向图或无向图来表示。因此，如何利用排名中的信息成为一个有趣的问题。这种设置在社交搜索和社交数据挖掘中尤为常见。请注意，PageRank[78]和BrowseRank[72]也是对图中对象进行排序的方法，但它们只利用了链接信息，是无监督学习方法。

例如，Agrawal等人提出了一种监督学习方法来对图中的对象进行排序。他们的方法采用了Markov随机行走模型，就像在PageRank中一样，并从训练数据中的对象偏好对中自动学习转移概率。该方法将学习任务形式化为一个以最大熵为目标的有约束的网络流问题，将马尔可夫性和偏好对表示为约束。

**参考文献**

[1] Alekh Agarwal, Soumen Chakrabarti, and Sunny Aggarwal. Learning to rank networked

entities. In KDD, pages 14–23, 2006. DOI: 10.1145/1150402.1150409 88

[2] Shivani Agarwal and Partha Niyogi. Stability and generalization of bipartite ranking algo-

rithms. In COL T, pages 32–47, 2005. 83

[3] Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorpo-

rating user behavior information. In Proceedings of the 29th annual international ACM SIGIR

conference on Research and development in information retrieval, SIGIR ’06, pages 19–26, New

Y ork, NY, USA, 2006. ACM. DOI: 10.1145/1148170.1148177 17

[4] Rakesh Agrawal, Alan Halverson, Krishnaram Kenthapadi, Nina Mishra, and Panayi-

otis Tsaparas. Generating labels from clicks. In WSDM, pages 172–181, 2009.

DOI: 10.1145/1498759.1498824 86

[5] Nir Ailon and Mehryar Mohri. An efficient reduction of ranking to classification. In COL T,

pages 87–98, 2008. 83

[6] Massih Reza Amini, Tuong Vinh Truong, and Cyril Goutte. A boosting algorithm for

learning bipartite ranking functions with partially labeled data. In SIGIR ’08: Proceedings of

the 31st annual international ACM SIGIR conference on Research and development in information

retrieval, pages 99–106, New Y ork, NY, USA, 2008. ACM. DOI: 10.1145/1390334.1390354

86

[7] Javed A. Aslam, Evangelos Kanoulas, Virgil Pavlu, Stefan Savev, and Emine Yilmaz.

Document selection methodologies for efficient and effective learning-to-rank. In SI-

GIR ’09: Proceedings of the 32nd international ACM SIGIR conference on Research and de-

velopment in information retrieval, pages 468–475, New Y ork, NY, USA, 2009. ACM.

DOI: 10.1145/1571941.1572022 86

[8] Javed A. Aslam and Mark Montague. Models for metasearch. In Proceedings of the 24th annual

international ACM SIGIR conference on Research and development in information retrieval,

SIGIR ’01, pages 276–284, New Y ork, NY, USA, 2001. ACM.DOI: 10.1145/383952.384007

35, 68

[9] Jing Bai, Ke Zhou, Guirong Xue, Hongyuan Zha, Gordon Sun, Belle Tseng, Zhaohui Zheng,

and Yi Chang. Multi-task learning for learning to rank in web search. In CIKM ’09: Proceedingof the 18th ACM conference on Information and knowledge management, pages 1549–1552, New

Y ork, NY, USA, 2009. ACM. DOI: 10.1145/1645953.1646169 87

[10] Michael Bendersky, W. Bruce Croft, and Yanlei Diao. Quality-biased ranking of web docu-

ments. In WSDM, pages 95–104, 2011. DOI: 10.1145/1935826.1935849 18

[11] Chris Burges,Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg

Hullender. Learning to rank using gradient descent. In ICML ’05: Proceedings of the 22nd inter-

national conference on Machine learning, pages 89–96, 2005. DOI: 10.1145/1102351.1102363

22, 25, 37, 49

[12] C.J.C. Burges, R. Ragno, and Q.V. Le. Learning to rank with nonsmooth cost functions. In

Advances in Neural Information Processing Systems 18, pages 395–402. MIT Press, Cambridge,

MA, 2006. 22, 26, 37, 52, 53

[13] Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon.

Adapting ranking SVM to document retrieval. In SIGIR’ 06, pages 186–193, 2006.

DOI: 10.1145/1148170.1148205 22, 25, 37, 44

[14] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from

pairwise approach to listwise approach. In ICML ’07: Proceedings of the 24th international

conference on Machine learning, pages 129–136, 2007. DOI: 10.1145/1273496.1273513 22,

27, 37, 55

[15] Soumen Chakrabarti, Rajiv Khanna, Uma Sawant, and Chiru Bhattacharyya. Structured

learning for non-smooth ranking losses. In KDD ’08: Proceeding of the 14th ACM SIGKDD

international conference on Knowledge discovery and data mining, pages 88–96, New Y ork, NY,

USA, 2008. ACM. DOI: 10.1145/1401890.1401906 60

[16] Olivier Chapelle and S. Sathiya Keerthi. Efficient algorithms for ranking with svms. Inf.

Retr., 13(3):201–215, 2010. DOI: 10.1007/s10791-009-9109-9 87

[17] Olivier Chapelle, Pannagadatta K. Shivaswamy, Srinivas Vadrevu, Kilian Q. Weinberger,

Ya Zhang, and Belle L. Tseng. Multi-task learning for boosting with application to web

search ranking. In KDD, pages 1189–1198, 2010. DOI: 10.1145/1835804.1835953 87

[18] Olivier Chapelle and Mingrui Wu. Gradient descent optimization of smoothed information

retrieval metrics. Inf. Retr., 13(3):216–235, 2010. DOI: 10.1007/s10791-009-9110-3 27

[19] Depin Chen, Yan Xiong, Jun Yan, Gui-Rong Xue, Gang Wang, and Zheng Chen.

Knowledge transfer for cross domain learning to rank. Inf. Retr., 13(3):236–253, 2010.

DOI: 10.1007/s10791-009-9111-2 87

[20] Wei Chen, Tie-Yan Liu, Yanyan Lan, Zhi-Ming Ma, and Hang Li. Ranking measures and

loss functions in learning to rank. In NIPS ’09, 2009. 83

[21] Wei Chen, Tie-Yan Liu, and Zhi-Ming Ma. Two-layer generalization analysis for ranking

using rademacher average. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel,

and A. Culotta, editors, Advances in Neural Information Processing Systems 23, pages 370–378.

2010. 83

[22] Weiwei Cheng, Jens Hühn, and Eyke Hüllermeier. Decision tree and instance-based

learning for label ranking. In ICML ’09: Proceedings of the 26th Annual International

Conference on Machine Learning, pages 161–168, New Y ork, NY, USA, 2009. ACM.

DOI: 10.1145/1553374.1553395 23

[23] Wei Chu and Zoubin Ghahramani. Gaussian processes for ordinal regression. J. Mach. Learn.

Res., 6:1019–1041, 2005. 23

[24] Wei Chu and S. Sathiya Keerthi. New approaches to support vector ordinal regression. In

ICML ’05: Proceedings of the 22nd international conference on Machine learning, pages 145–152,

2005. DOI: 10.1145/1102351.1102370 23

[25] Stéphan J.M. Clémençon and Nicolas Vayatis. Empirical performance maximization for linear

rank statistics. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, Advances in

Neural Information Processing Systems 21, pages 305–312. 2009. 83

[26] W. William Cohen, R. E. Schapire, and Y oram Singer. Learning to order things. JAIR,

10:243–270, 1999. 35

[27] C. Cortes and V. Vapnik. Support-vector networks. Machine Learning, 20(3):273–297, 1995.

DOI: 10.1023/A:1022627411411 40

[28] Corinna Cortes, Mehryar Mohri, and Ashish Rastogi. Magnitude-preserving ranking algo-

rithms. In ICML ’07: Proceedings of the 24th international conference on Machine learning, pages

169–176, New Y ork, NY, USA, 2007. ACM. DOI: 10.1145/1273496.1273518 83

[29] David Cossock and Tong Zhang. Subset ranking using regression. In COL T ’06:

Proceedings of the 19th Annual Conference on Learning Theory, pages 605–619, 2006.

DOI: 10.1007/11776420\_44 22, 24, 82

[30] Koby Crammer and Y oram Singer. Pranking with ranking. In NIPS, pages 641–647, 2001.

21, 22, 23, 37

[31] Anlei Dong, Yi Chang, Zhaohui Zheng, Gilad Mishne, Jing Bai, Ruiqiang Zhang, Karolina

Buchner, Ciya Liao, and Fernando Diaz. Towards recency ranking in web search. In WSDM,

pages 11–20, 2010. DOI: 10.1145/1718487.1718490 75

[32] Pinar Donmez, Krysta M. Svore, and Christopher J.C. Burges. On the local optimality of

lambdarank. In SIGIR ’09: Proceedings of the 32nd international ACM SIGIR conference onResearch and development in information retrieval, pages 460–467, New Y ork, NY, USA, 2009.

ACM. DOI: 10.1145/1571941.1572021 37, 53

[33] Kevin Duh and Katrin Kirchhoff. Learning to rank with partially-labeled data. In SI-

GIR ’08: Proceedings of the 31st annual international ACM SIGIR conference on Research and

development in information retrieval, pages 251–258, New Y ork, NY, USA, 2008. ACM.

DOI: 10.1145/1390334.1390379 86

[34] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for

the web. In Proceedings of the 10th international conference on World Wide Web, WWW ’01,

pages 613–622, New Y ork, NY, USA, 2001. ACM. DOI: 10.1145/371920.372165 35, 37,

69

[35] Jonathan L. Elsas, Vitor R. Carvalho, and Jaime G. Carbonell. Fast learning of document

ranking functions with the committee perceptron. In WSDM ’08: Proceedings of the interna-

tional conference on Web search and web data mining, pages 55–64, New Y ork, NY, USA, 2008.

ACM. DOI: 10.1145/1341531.1341542 24

[36] Y. Freund and R. Schapire. A short introduction to boosting. Journal of Japanese Society for

Artificial Intelligence, 14(5):771–780, 1999. 60

[37] Y oav Freund, Raj D. Iyer, Robert E. Schapire, and Y oram Singer. An efficient boosting

algorithm for combining preferences. Journal of Machine Learning Research, 4:933–969, 2003.

DOI: 10.1162/jmlr.2003.4.6.933 22, 25, 76

[38] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. The

Annals of Statistics, 29(5):1189—1232, 2001. DOI: 10.1214/aos/1013203450 26, 48

[39] Jianfeng Gao, Haoliang Qi, Xinsong Xia, and Jian-Yun Nie. Linear discriminant model for

information retrieval. In SIGIR, pages 290–297, 2005. DOI: 10.1145/1076034.1076085 11

[40] Wei Gao, Peng Cai, Kam-Fai Wong, and Aoying Zhou. Learning to rank only using training

data from related domain. In Proceeding of the 33rd international ACM SIGIR conference on

Research and development in information retrieval, SIGIR ’10, pages 162–169, New Y ork, NY,

USA, 2010. ACM. DOI: 10.1145/1835449.1835478 87

[41] Xiubo Geng, Tie-Yan Liu, Tao Qin, Andrew Arnold, Hang Li, and Heung-Yeung Shum.

Query dependent ranking using k-nearest neighbor. In SIGIR ’08: Proceedings of the 31st an-

nual international ACM SIGIR conference on Research and development in information retrieval,

pages 115–122, New Y ork, NY, USA, 2008. ACM. DOI: 10.1145/1390334.1390356 21, 87

[42] Xiubo Geng, Tie-Yan Liu, Tao Qin, and Hang Li. Feature selection for ranking. In SI-

GIR ’07: Proceedings of the 30th annual international ACM SIGIR conference on Research

and development in information retrieval, pages 407–414, New Y ork, NY, USA, 2007. ACM.

DOI: 10.1145/1277741.1277811 87

[43] John Guiver and Edward Snelson. Learning to rank with softrank and gaussian processes.

In SIGIR ’08: Proceedings of the 31st annual international ACM SIGIR conference on Research

and development in information retrieval, pages 259–266, New Y ork, NY, USA, 2008. ACM.

DOI: 10.1145/1390334.1390380 37, 64

[44] John Guiver and Edward Snelson. Bayesian inference for plackett-luce ranking models. In

ICML ’09: Proceedings of the 26th Annual International Conference on Machine Learning, pages

377–384, New Y ork, NY, USA, 2009. ACM. DOI: 10.1145/1553374.1553423 55

[45] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with

trustrank. In Proceedings of the Thirtieth international conference on Very large data bases -

Volume 30, VLDB ’04, pages 576–587. VLDB Endowment, 2004. 18

[46] Edward F. Harrington. Online ranking/collaborative filtering using the perceptron algorithm.

In ICML, pages 250–257, 2003. 76

[47] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Support vector learning for ordinal

regression. May 20 1999. DOI: 10.1049/cp:19991091 37, 40

[48] Ralf Herbrich,Thore Graepel, and Klaus Obermayer. Large Margin rank boundaries for ordinal

regression. MIT Press, Cambridge, MA, 2000. 22, 24, 37, 40

[49] Steven C. H. Hoi and Rong Jin. Semi-supervised ensemble ranking. In Dieter Fox and

Carla P . Gomes, editors, AAAI, pages 634–639. AAAI Press, 2008. 86

[50] Yunhua Hu, Hang Li, Yunbo Cao, Dmitriy Meyerzon, and Qinghua Zheng. Automatic

extraction of titles from general documents using machine learning. In Mary Marlino,Tamara

Sumner, and Frank M. Shipman III, editors, JCDL, pages 145–154. ACM, 2005. 17

[51] Yunhua Hu, Guomao Xin, Ruihua Song, Guoping Hu, Shuming Shi, Yunbo Cao, and Hang

Li. Title extraction from bodies of HTML documents and its application to web page

retrieval. In Ricardo A. Baeza-Yates, Nivio Ziviani, Gary Marchionini, Alistair Moffat, and

John Tait, editors, SIGIR, pages 250–257. ACM, 2005. 17

[52] Jim C. Huang and Brendan J. Frey. Structured ranking learning using cumulative distribution

networks. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, Advances in Neural

Information Processing Systems 21, pages 697–704. 2009. 88

[53] Kalervo Järvelin and Jaana Kekäläinen. In evaluation methods for retrieving highly relevant

documents. In Proceedings of the 23rd annual international ACM SIGIR conference on Research

and development in information retrieval, SIGIR ’00, pages 41–48, New Y ork, NY, USA, 2000.

ACM. DOI: 10.1145/345508.345545 17

[54] Shihao Ji, Ke Zhou, Ciya Liao, Zhaohui Zheng, Gui-Rong Xue, Olivier Chapelle, Gordon

Sun, and Hongyuan Zha. Global ranking by exploiting user clicks. In SIGIR ’09: Proceedings

of the 32nd international ACM SIGIR conference on Research and development in information

retrieval, pages 35–42, New Y ork, NY, USA, 2009. ACM. DOI: 10.1145/1571941.1571950

88

[55] Xin Jiang, Yunhua Hu, and Hang Li. A ranking approach to keyphrase extraction. In James

Allan, Javed A. Aslam, Mark Sanderson, ChengXiang Zhai, and Justin Zobel, editors, SIGIR,

pages 756–757. ACM, 2009. 76

[56] Rong Jin, Hamed Valizadegan, and Hang Li. Ranking refinement and its application to

information retrieval. In WWW ’08: Proceeding of the 17th international conference on World Wide

Web, pages 397–406, New Y ork, NY, USA, 2008. ACM. DOI: 10.1145/1367497.1367552

86

[57] T . Joachims. Optimizing search engines using clickthrough data. In KDD’ 02, pages 133–142,

2002. DOI: 10.1145/775047.775067 15, 85

[58] Thorsten Joachims, Hang Li, Tie-Yan Liu, and ChengXiang Zhai. Learning to

rank for information retrieval (LR4IR 2007). SIGIR Forum, 41(2):58–62, 2007.

DOI: 10.1145/1328964.1328974 4

[59] Alexandre Klementiev, Dan Roth, and Kevin Small. Unsupervised rank aggrega-

tion with distance-based models. In Proceedings of the 25th international conference

on Machine learning, ICML ’08, pages 472–479, New Y ork, NY, USA, 2008. ACM.

DOI: 10.1145/1390156.1390216 35

[60] Yanyan Lan, Tie-Yan Liu, Zhiming Ma, and Hang Li. Generalization analysis of listwise

learning-to-rank algorithms. In ICML ’09: Proceedings of the 26th Annual International Con-

ference on Machine Learning, pages 577–584, New Y ork, NY, USA, 2009. ACM. 83

[61] Yanyan Lan, Tie-Yan Liu, Tao Qin, Zhiming Ma, and Hang Li. Query-level stability

and generalization in learning to rank. In ICML ’08: Proceedings of the 25th interna-

tional conference on Machine learning, pages 512–519, New Y ork, NY, USA, 2008. ACM.

DOI: 10.1145/1390156.1390221 83

[62] Quoc V. Le and Alexander J. Smola. Direct optimization of ranking measures. CoRR,

abs/0704.3359, 2007. informal publication; informal publication. 60

[63] Guy Lebanon and John D. Lafferty. Cranking: Combining rankings using conditional prob-

ability models on permutations. In ICML ’02: Proceedings of the Nineteenth International

Conference on Machine Learning, pages 363–370, San Francisco, CA, USA, 2002. Morgan

Kaufmann Publishers Inc. 35, 37, 71

[64] Hang Li, Tie-Yan Liu, and ChengXiang Zhai. Learning to rank for information retrieval

(LR4IR 2008). SIGIR Forum, 42(2):76–79, 2008. DOI: 10.1145/1480506.1480519 4

[65] Hang Li, Tie-Yan Liu, and ChengXiang Zhai. Learning to rank for information retrieval

(LR4IR 2009). SIGIR Forum, 43(2):41–45, 2009. DOI: 10.1145/1670564.1670571 4

[66] Ming Li, Hang Li, and Zhi-Hua Zhou. Semi-supervised document retrieval. Inf. Process.

Manage, 45(3):341–355, 2009. DOI: 10.1016/j.ipm.2008.11.002 86

[67] Ping Li, Christopher Burges, and Qiang Wu. Mcrank: Learning to rank using multiple

classification and gradient boosting. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors,

Advances in Neural Information Processing Systems 20, pages 897–904. MIT Press, Cambridge,

MA, 2008. 21, 22, 24, 83

[68] Tie-Yan Liu. Learning to rank for information retrieval. Foundations and Trends in Information

Retrieval, 3(3):225–331, 2009. DOI: 10.1561/1500000016 4

[69] Tie-Yan Liu, Thorsten Joachims, Hang Li, and Chengxiang Zhai. Introduction to spe-

cial issue on learning to rank for information retrieval. Inf. Retr., 13(3):197–200, 2010.

DOI: 10.1007/s10791-009-9120-1 4

[70] Tie-Yan Liu, Jun Xu, Tao Qin, Wenying Xiong, and Hang Li. Letor: Benchmark dataset for

research on learning to rank for information retrieval. In Proceedings of SIGIR 2007 Workshop

on Learning to Rank for Information Retrieval, 2007. DOI: 10.1561/1500000016 4

[71] Yu-Ting Liu, Tie-Yan Liu, Tao Qin, Zhi-Ming Ma, and Hang Li. Supervised rank aggre-

gation. In WWW ’07: Proceedings of the 16th international conference on World Wide Web, pages

481–490, New Y ork, NY, USA, 2007. ACM. DOI: 10.1145/1242572.1242638 35

[72] Yuting Liu, Bin Gao, Tie-Yan Liu, Ying Zhang, Zhiming Ma, Shuyuan He, and Hang Li.

Browserank: letting web users vote for page importance. In Sung-Hyon Myaeng, Douglas W.

Oard, Fabrizio Sebastiani, Tat-Seng Chua, and Mun-Kew Leong, editors, SIGIR, pages 451–

458. ACM, 2008. 18, 87, 88

[73] Bo Long, Olivier Chapelle, Ya Zhang, Yi Chang, Zhaohui Zheng, and Belle L. Tseng. Ac-

tive learning for ranking through expected loss optimization. In Fabio Crestani, Stéphane

Marchand-Maillet, Hsin-Hsi Chen, Efthimis N. Efthimiadis, and Jacques Savoy, editors,

SIGIR, pages 267–274. ACM, 2010. 86

[74] Irina Matveeva, Chris Burges, Timo Burkard, Andy Laucius, and Leon Wong. High accuracy

retrieval with multiple nested ranker. In SIGIR ’06: Proceedings of the 29th annual international

ACM SIGIR conference on Research and development in information retrieval, pages 437–444,

New Y ork, NY, USA, 2006. ACM. DOI: 10.1145/1148170.1148246 21

[75] Donald Metzler and W. Bruce Croft. A markov random field model for term dependencies.

In SIGIR ’05: Proceedings of the 28th annual international ACM SIGIR conference on Research

and development in information retrieval, pages 472–479, New Y ork, NY, USA, 2005. ACM

Press. DOI: 10.1145/1076034.1076115 86

[76] Donald Metzler and Tapas Kanungo. Machine learned sentence selection strategies for

query-biased summarization. sigir learning to rank workshop, 2008. 77

[77] Taesup Moon, Alex J. Smola, Yi Chang, and Zhaohui Zheng. Intervalrank: isotonic

regression with listwise and pairwise constraints. In WSDM, pages 151–160, 2010.

DOI: 10.1145/1718487.1718507 24

[78] L. Page, S. Brin, R. Motwani, and T . Winograd. The pagerank citation ranking: Bringing

order to the web. Technical report, Stanford University, Stanford, CA, 1998. 17, 18, 88

[79] Ashok Kumar Ponnuswami, Kumaresh Pattabiraman, Qiang Wu, Ran Gilad-Bachrach, and

Tapas Kanungo. On composition of a federated web search result page: using online users

to provide pairwise preference for heterogeneous verticals. In WSDM, pages 715–724, 2011.

DOI: 10.1145/1935826.1935922 75

[80] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval.

In Proceedings of the 21st annual international ACM SIGIR conference on Research and develop-

ment in information retrieval, SIGIR ’98, pages 275–281, New Y ork, NY, USA, 1998. ACM.

DOI: 10.1145/290941.291008 11

[81] Tao Qin, Tie-Yan Liu, and Hang Li. A general approximation framework for di-

rect optimization of information retrieval measures. Inf. Retr., 13(4):375–397, 2010.

DOI: 10.1007/s10791-009-9124-x 22, 28

[82] Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. Letor: A benchmark collection for re-

search on learning to rank for information retrieval. Inf. Retr., 13(4):346–374, 2010.

DOI: 10.1007/s10791-009-9123-y 28

[83] Tao Qin, Tie-Yan Liu, Xu-Dong Zhang, De-Sheng Wang, and Hang Li. Global ranking

using continuous conditional random fields. In D. Koller, D. Schuurmans, Y. Bengio, and

L. Bottou, editors, Advances in Neural Information Processing Systems 21, pages 1281–1288.

2009. 88

[84] Tao Qin, Tie-Yan Liu, Xu-Dong Zhang, De-Sheng Wang, Wen-Ying Xiong, and Hang Li.

Learning to rank relational objects and its application to web search. In WWW ’08: Proceeding

of the 17th international conference on World Wide Web, pages 407–416, New Y ork, NY, USA,

2008. ACM. DOI: 10.1145/1367497.1367553 88

[85] Tao Qin, Xu-Dong Zhang, Ming-Feng Tsai, De-Sheng Wang, Tie-Yan Liu, and Hang Li.

Query-level loss functions for information retrieval. Inf. Process. Manage., 44(2):838–855,

2008. DOI: 10.1016/j.ipm.2007.07.016 27

[86] Tao Qin, Xu-Dong Zhang, De-Sheng Wang, Tie-Yan Liu, Wei Lai, and Hang Li. Rank-

ing with multiple hyperplanes. In Proceedings of the 30th annual international ACM SIGIR

conference, pages 279–286, 2007. DOI: 10.1145/1277741.1277791 24

[87] Filip Radlinski and Thorsten Joachims. Query chains: learning to rank from implicit feedback.

In KDD ’05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge

discovery in data mining, pages 239–248, 2005. DOI: 10.1145/1081870.1081899 15, 86

[88] Filip Radlinski and Thorsten Joachims. Active exploration for learning rankings from click-

through data. In KDD ’07: Proceedings of the 13th ACM SIGKDD international conference

on Knowledge discovery and data mining, pages 570–579, New Y ork, NY, USA, 2007. ACM.

DOI: 10.1145/1281192.1281254 15, 86

[89] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with

multi-armed bandits. In ICML ’08: Proceedings of the 25th international conference on Machine

learning, pages 784–791, New Y ork, NY, USA, 2008. ACM.DOI: 10.1145/1390156.1390255

88

[90] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model

for probabilistic weighted retrieval. In Proceedings of the 17th annual international ACM SIGIR

conference on Research and development in information retrieval, SIGIR ’94, pages 232–241, New

Y ork, NY, USA, 1994. Springer-Verlag New Y ork, Inc. 11, 16, 18

[91] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization

in the brain. Psychological Review, 65:386–408, 1958. DOI: 10.1037/h0042519 37

[92] Amnon Shashua and Anat Levin. Ranking with large margin principle: Two approaches.

In S. Thrun S. Becker and K. Obermayer, editors, Advances in Neural Information Processing

Systems 15. MIT Press. 21, 22, 23, 37, 38, 40

[93] Libin Shen, Anoop Sarkar, and Franz Josef Och. Discriminative reranking for machine

translation. In HL T-NAACL, pages 177–184, 2004. 77

[94] Tao Tao and ChengXiang Zhai. An exploration of proximity measures in information re-

trieval. In Proceedings of the 30th annual international ACM SIGIR conference on Research and

development in information retrieval, SIGIR ’07, pages 295–302, New Y ork, NY, USA, 2007.

ACM. DOI: 10.1145/1277741.1277794 18

[95] Michael Taylor, John Guiver, Stephen Robertson, and Tom Minka. Softrank: optimiz-

ing non-smooth rank metrics. In WSDM ’08: Proceedings of the international conferenceon Web search and web data mining, pages 77–86, New Y ork, NY, USA, 2008. ACM.

DOI: 10.1145/1341531.1341544 22, 28, 37, 64

[96] Michael Taylor, Hugo Zaragoza, Nick Craswell, Stephen Robertson, and Chris Burges. Opti-

misation methods for ranking functions with multiple parameters. In CIKM ’06: Proceedings

of the 15th ACM international conference on Information and knowledge management, pages

585–593, New Y ork, NY, USA, 2006. ACM. DOI: 10.1145/1183614.1183698 86

[97] Ming-Feng Tsai, Tie-Yan Liu, Tao Qin, Hsin-Hsi Chen, and Wei-Ying Ma. Frank: a ranking

method with fidelity loss. In Proceedings of the 30th annual international ACM SIGIR conference,

pages 383–390, 2007. DOI: 10.1145/1277741.1277808 22, 26

[98] Nicolas Usunier, David Buffoni, and Patrick Gallinari. Ranking with ordered weighted

pairwise classification. In ICML ’09: Proceedings of the 26th Annual International Con-

ference on Machine Learning, pages 1057–1064, New Y ork, NY, USA, 2009. ACM.

DOI: 10.1145/1553374.1553509 24

[99] Hamed Valizadegan, Rong Jin, Ruofei Zhang, and Jianchang Mao. Learning to rank by

optimizing ndcg measure. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and

A. Culotta, editors, Advances in Neural Information Processing Systems 22, pages 1883–1891.

2009. 27

[100] Maksims N. Volkovs and Richard S. Zemel. Boltzrank: learning to maximize expected ranking

gain. In ICML ’09: Proceedings of the 26th Annual International Conference on Machine Learning,

pages 1089–1096, New Y ork, NY, USA, 2009. ACM. DOI: 10.1145/1553374.1553513 27

[101] Ellen M. Voorhees and Donna Harman. TREC: Experiment and Evaluation in Information

Retrieval. MIT , 2005. 17

[102] Qiang Wu, Christopher J. C. Burges, Krysta Marie Svore, and Jianfeng Gao. Adapt-

ing boosting for information retrieval measures. Inf. Retr., 13(3):254–270, 2010.

DOI: 10.1007/s10791-009-9112-1 22, 26

[103] Fen Xia, Tie-Yan Liu, and Hang Li. Statistical consistency of top-k ranking. In Y. Bengio,

D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, Advances in Neural

Information Processing Systems 22, pages 2098–2106. 2009. 83

[104] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach

to learning to rank: theory and algorithm. In ICML ’08: Proceedings of the 25th interna-

tional conference on Machine learning, pages 1192–1199, New Y ork, NY, USA, 2008. ACM.

DOI: 10.1145/1390156.1390306 22, 27, 37, 55, 83

[105] Biao Xiang, Daxin Jiang, Jian Pei, Xiaohui Sun, Enhong Chen, and Hang Li. Context-aware

ranking in web search. In Fabio Crestani, Stéphane Marchand-Maillet, Hsin-Hsi Chen,

Efthimis N. Efthimiadis, and Jacques Savoy, editors, SIGIR, pages 451–458. ACM, 2010. 75

[106] Jingfang Xu, Chuanliang Chen, Gu Xu, Hang Li, and Elbio Renato Torres Abib. Improving

quality of training data for learning to rank using click-through data. In Brian D. Davison,

Torsten Suel, Nick Craswell, and Bing Liu, editors, WSDM, pages 171–180. ACM, 2010. 86

[107] Jun Xu, Yunbo Cao, Hang Li, and Min Zhao. Ranking definitions with supervised

learning methods. In Special interest tracks and posters of the 14th international confer-

ence on World Wide Web, WWW ’05, pages 811–819, New Y ork, NY, USA, 2005. ACM.

DOI: 10.1145/1062745.1062761 76

[108] Jun Xu and Hang Li. Adarank: a boosting algorithm for information retrieval. In SI-

GIR ’07: Proceedings of the 30th annual international ACM SIGIR conference on Research

and development in information retrieval, pages 391–398, New Y ork, NY, USA, 2007. ACM.

DOI: 10.1145/1277741.1277809 22, 27, 37, 59, 60

[109] Jun Xu, Hang Li, and Chaoliang Zhong. Relevance ranking using kernels. In Pu-Jen Cheng,

Min-Yen Kan, Wai Lam, and Preslav Nakov, editors, AIRS, volume 6458 of Lecture Notes in

Computer Science, pages 1–12. Springer, 2010. 18

[110] Jun Xu, Tie-Yan Liu, Min Lu, Hang Li, and Wei-Ying Ma. Directly optimizing evaluation

measures in learning to rank. In SIGIR ’08: Proceedings of the 31st annual international ACM

SIGIR conference on Research and development in information retrieval, pages 107–114, New

Y ork, NY, USA, 2008. ACM. DOI: 10.1145/1390334.1390355 22, 27, 60

[111] Yisong Yue,Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method

for optimizing average precision. In Proceedings of the 30th annual international ACM SIGIR

conference, pages 271–278, 2007. DOI: 10.1145/1277741.1277790 22, 27, 37, 60

[112] Yisong Yue and T . Joachims. Predicting diverse subsets using structural SVMs.

In International Conference on Machine Learning (ICML), pages 271–278, 2008.

DOI: 10.1145/1390156.1390310 88

[113] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language mod-

els applied to information retrieval. ACM Trans. Inf. Syst., 22:179–214, April 2004.

DOI: 10.1145/984321.984322 11

[114] Zhaohui Zheng, Hongyuan Zha, Keke Chen, and Gordon Sun. A regression framework

for learning ranking functions using relative relevance judgments. In Proceedings of the 30th

annual international ACM SIGIR conference, 2007. DOI: 10.1145/1277741.1277792 37, 47

[115] Zhaohui Zheng, Hongyuan Zha, Tong Zhang, Olivier Chapelle, Keke Chen, and Gordon

Sun. A general boosting method and its application to learning ranking functions for web

search. In J.C. Platt, D. Koller,Y. Singer, and S. Roweis, editors,Advances in Neural Information

Processing Systems 20, pages 1697–1704. MIT Press, Cambridge, MA, 2008. 22, 26, 37, 47

116] Ke Zhou, Gui-Rong Xue, Hongyuan Zha, and Y ong Yu. Learning to rank with ties. In

Proceedings of the 31st annual international ACM SIGIR conference, pages 275–282, 2008.

DOI: 10.1145/1390334.1390382 24