Ranking svm

Ranking svm 的缺点

IR svm 对 ranking svm的改进

对于查询的文档，输入为X属于（），n 代表了特征的数量，输出为Y属于，q 代表了对于排名第输出。 并且在rq,之间存在排列顺序，的级别， 存在一个顺序，

preference relationship，偏好关系， 一系列的排名函数 存 在 ，并且每个都能欧决定。

存在一组排序函数f∈f，每个排序函数可以确定实例之间的优先关系:

Suppose that we are given a set of ranked instances   a  from the space X ×Y . The task here is to select the best function f\* from F that minimizes a given loss function with respect to the given ranked instances. 假设我们从空间X×Y中得到一组排序实例a。这里的任务是从f中选择最佳函数f\*，使给定的损失函数相对于给定的排序实例最小化。

对于给定的排序实例。Herbrich等人[8]提出将上述学习问题形式化为对实例进行分类的学习问题。首先，我们假设f是一个线性函数。

## 学习排序算法

下面首先介绍IR SVM，由于IR SVM是在ranking SVM基础上进行改进的结果，首先介绍Ranking SVM。最后介绍在需求追溯 时 特别对 IR SVM进行的改进。

Herbrich等人提出了Ranking svm的方法，该方法的基本思想是通过训练得到能对文档对于进行分类的分类器（SVM），然后利用分类器完成排序任务。

如图（）所示，假设在特征空间中有两组与查询关联的文档集合。每个文档集合中的文档，划分成了三个等级。 例如，第一组中的对象x1、x2和x3的级别分别rank3、rank2、rank1。

此时，可以使用第二章的公式（2.2）来进行排序，此处x1、x2和x3对应 2.2中的得到的对象向量，w仍代表每个特征向量的权重矩阵。根据该公式来执行排序操作，实际上是将文档的特征向量投影到各自权重向量中，并根据特征向量的投影来对文档进行排序。

这样的好处是可以把每个特征向量的权重w 进行组合，形成新的适合所有特征向量的权重。

此外，标签也被分配到新的特征向量。例如，x1 - x2, x1 - x3, x2 - x3是正的。注意，同一等级的特征向量或不同组的特征向量不用于创建新的特征向量。

* Figure 5 shows that the ranking problem in Figure 4 can be transformed to Linear SVM classification.

我们可以将图4中的排序问题进行，为通过线性SVM分类来解决。

首先，将同一组不同等级特征向量之间的差异作为新的特征向量处理，如x1 - x2, x1 - x3, x2 - x3。

此外，标签也被分配到新的特征向量。例如，x1 - x2, x1 - x3, x2 - x3是正的。

* Geometrically, the margin in the SVM model represents the closest distance between the projections of object pairs in two grades.

几何上，支持向量机模型中的边界线 表示两个等级的目标的投影之间的最近距离。

在在svm的超平面中，通过原实例 形成的 正、负实例 是成对相关的例如，x1 - x2和x2 - x1分别是正的和负的实例。因此我们在训练时可以只选择正实例，而负实例与正实例是一样的，可以不进行重复训练。

* The weight vector w of the SVM classifier corresponds to the ranking function.

此处，SVM分类器的权重向量w 其实就是 排序函数。

训练数据为{(x(1) i, x(2) i)， yi)}，

i = 1，···，m，

特征向量(x(1) i, x(2) i)代表两个训练实例， yi∈{+1，- 1}，表示 哪个特征向量应该排在前面。

将SVM的学习问题形式化的表示为下面一下QP（Quadratic Programming） 二次规划问题。

x(1) i和x(2) i分别表示一对特征向量中的第一个和第二个特征向量，||·||表示第二范式，m表示训练向量的代号，c为大于0的系数。它等价于下面的无约束优化问题，即，使损失函数最小化。

下面开始是IR SVM

由Cao等人提出的IR SVM。[15]是信息检索(IR)中排序SVM的扩展，

SVM将排序转化为两两分类，利用了学习过程中的0-1损失。损失函数与红外评价指标之间存在一定的差距。IR SVM试图通过修正0-1损失来弥补这一差距，即对SVM进行代价敏感的排序学习。首先，我们将使用图6中的示例来研究将排序SVM直接应用于文档检索所导致的问题。

**问题一：**分类支持向量机方向应用的一个问题是，分类支持向量机对不同等级的文档对一视同仁。例1指出了这个问题。同一个查询有两个排名。将位置1和2的文档从完美排名中交换到排名1中，将位置3和4的文档从完美排名中交换到排名2中。按照0-1的赔率或对赔率的顺序，每个排名只有一个错误。它们对SVM排序训练的效果是相同的，这是不可取的。从IR的角度来看，排名2应该比排名1好，因为排名顶部的结果更好。需要注意的是，对于一个IR系统来说，对排名靠前的文档具有较高的准确性是至关重要的，

**问题二：**支持向量机排序的另一个问题是，它平等地处理来自不同查询的文档对。在例2中，有两个查询，与它们关联的文档数量不同。查询-1 3-2年级有2对文档，3-1年级有4对文档，2-1年级有8对文档，共14对文档。对于query-2，有31对文档对。SVM的排序从query- 1中获取14个实例(文档对)，从query-2中获取31个实例(文档对)进行训练。因此，query-2对排名模型的影响将大于query-1对排名模型的影响。换句话说，所学习的模型将偏向于query-2。这与在IR评估查询中同等重要的事实形成了对比。注意，文档的数量通常因查询而异。

R SVM通过将0-1对分类转化为代价敏感的对分类，解决了上述两个问题。该方法通过修改支持向量机的合页损失函数(hinge loss)来实现。具体地说，它为不同级别和不同查询的文档对设置了不同的损失。

为了强调正确的排名在顶部的重要性，损失函数严重惩罚与顶部相关的错误。1）对于Top doc，即相似度等级较高的doc所在的pair，赋予较大的loss weight。

为了增加文档较少的查询的影响，对于doc数目较少的query，对其下面的doc pair赋予较大的loss weight。

IR SVM针对以上两个问题进行了解决，它使用了cost sensitive classification，而不是0-1 classification，即对通常的合页损失函数(hinge loss)进行了改造。具体来说，它对来自不同等级的doc pair，或者来自不同query的doc pair，赋予了不同的loss weight：

1）对于Top doc，即相似度等级较高的doc所在的pair，赋予较大的loss weight。

2）对于doc数目较少的query，对其下面的doc pair赋予较大的loss weight。

## 特征选择

研究证明，以某些文本特征作为输入进行学习排序在文本检索和需求追溯领域有着很好的效果，本文从常用的特征中选择以下五个特征，作为学习排序模型中构建文档向量的特征值。

（1）语义相似度，通过提出的语义相似度算法计算；

（2）广义Jaccard系数，通过计算文档向量的关系表示两个文档上下文相似度，在本文应用中即为查询语句和一个候选文档之间的上下文相似度，如公式(3.10)所示。

(3.10)

其中，向量和向量均为文档向量，此处文档向量使用词向量的平均值表示。和分别表示向量和向量的数值大小。

（3）IDF之和，在候选结果文本上中计算查询中的词或短语的IDF值，并求和，用来表示当前查询在某一候选结果上的重要程度，如公式(3.11)所示。

(3.11)

其中为查询中的单词或短语，表示在查询结果中的idf值。

（4）关键词数量，表示查询候选结果中关键词的数量，关键词的界定使用TFIDF的值，计算过程与加权策略中介绍的过程相同。

（5）文本长度，表示查询候选结果的长度，在一定程度上体现该结果的有效程度。

表3 学习排序算法选择的特征

|  |  |  |
| --- | --- | --- |
| **特征** | **描述** | **是否依赖查询** |
| 语义相似度 | 根据上一节的相似度算法计算文本语义相似度 | 是 |
| 广义Jaccard系数 | 查询语句和一个候选文档之间的上下文相似度 | 是 |
| IDF之和 | 候选结果文本上的当前查询中所包含的词或短语的IDF值的平均值 | 是 |
| 关键字数量 | 候选结果中关键字数量，通过计算TFIDF然后排序和取阈值得到 | 否 |
| 有效词数量 | 去停用词之后，所有不同的词的个数 |  |