**目 录**

[第一章 强化学习 1](#_Toc33120273)

[1.1 被动强化学习 1](#_Toc33120274)

[1.1.1 直接效用估计 1](#_Toc33120275)

[1.1.2 自适应动态规划 2](#_Toc33120276)

[1.1.3 时序差分学习 2](#_Toc33120277)

[1.2 主动强化学习 3](#_Toc33120278)

[1.2.1 探索 3](#_Toc33120279)

[1.2.2 学习行动-效用函数 3](#_Toc33120280)

[1.3 强化学习中的泛化 4](#_Toc33120281)

[1.4 策略搜索 4](#_Toc33120282)

[1.5 强化学习的应用 4](#_Toc33120283)

[第二章 自然语言处理 5](#_Toc33120284)

[2.1 语言模型 5](#_Toc33120285)

[2.1.1 n元字符模型 5](#_Toc33120286)

[2.1.2 n元模型的平滑 5](#_Toc33120287)

[2.1.3 模型评估 6](#_Toc33120288)

[2.1.4 n元单词模型 6](#_Toc33120289)

[2.2 文本分类 6](#_Toc33120290)

[2.2.1 数据压缩的分类方法 7](#_Toc33120291)

[2.3 信息检索 7](#_Toc33120292)

[2.3.1 IR评分函数 8](#_Toc33120293)

[2.3.2 IR系统评价 8](#_Toc33120294)

[2.3.3 IR的改进 8](#_Toc33120295)

[2.3.4 PageRank算法 9](#_Toc33120296)

[2.3.5 HITS算法 9](#_Toc33120297)

[2.3.6 问题回答 9](#_Toc33120298)

[2.4 信息抽取 9](#_Toc33120299)

[2.4.1 基于有限状态自动机的信息抽取 10](#_Toc33120300)

[2.4.2 信息抽取的概率模型 10](#_Toc33120301)

[2.4.3 基于条件随机场的概率模型 10](#_Toc33120302)

[2.4.4 从大型语料库中抽取本体 11](#_Toc33120303)

[2.4.5 自动模板构建 11](#_Toc33120304)

[2.4.6 机器阅读 11](#_Toc33120305)

[2.4.7 功能需求分析 12](#_Toc33120306)

[2.4.8 性能需求分析 12](#_Toc33120307)

# 强化学习

强化学习研究Agent在没有“做什么”的标注样例的情形下怎样学习“做什么”。

**强化学习的任务是：利用观察到的回报来学习针对某个环境的最优（或接近最优）策略。**

Agent设计主要有如下三种：

* 基于效用的Agent：学习关于状态的效用函数并使用它选择期望的结果效用最大化。
* Q-学习 Agent学习行动-价值函数，或称为Q函数，该函数提供在给定状态下采取特定行动的期望。
* 反射型Agent：学习一种策略，该策略直接将状态映射到行动。

## 被动强化学习

Agent的策略是固定的，其任务是学习状态的效用，还可能涉及对环境的模型进行学习。

Agent的策略是固定的：在状态，它总是执行行动。其目标只是简单的学习该策略有多好——即学习效用函数。

被动学习的任务类似于**策略评价** ，它是**策略迭代**算法的一部分。主要区别在于被动学习Agent对指定完成行动后从状态到达状态的概率的转移模型一无所知；并且也不知道指定每个状态的回报函数。

在该环境中，Agent应用其策略执行一组试验。每次试验，Agent从状态（1，1）开始，经历一个状态转移序列直至到达状态（4，2）或（4，3）。每个状态感知信息都用下标注明了所获得的回报。目标是利用关于回报的信息学习到与每个非终止状态相关联的期望效用。效用被定义为当遵循策略时所获得的回报的期望总和：

其中是状态的回报，是在时刻当执行策略时达到的状态，，在所有的公式中包含一个折扣因子 .

### 直接效用估计

**直接效用估计：把对一个给定状态全部观察到的未来回报用作学习效用的直接证据。**

直接效用估计成功地将强化学习问题简化为归纳学习问题，但是它忽略了一个重要的信息来源，即“状态的效用并非相互独立的”。每个状态的效用等于它自己的回报加上其后继状态的期望效用。也就是说，效用值服从固定策略的贝尔曼方程：

由于忽略了状态之间的联系，直接效用估计错失了学习的机会。我们可以把直接效用估计视为在比实际需要大得多的假设空间中搜索，其中包含许多违反贝尔曼方程组的函数。因此，该算法的收敛速度经常很慢。

### 自适应动态规划

**自适应动态规划：从观察中学习一个模型和一个回报函数，然后应用价值迭代或策略迭代获得效用或一个最优策略。**

就其价值估计的改进速度而言，ADP Agent受限于它学习转移模型的能力。从这个意义上，它提供了一个用以度量其他强化学习算法的标准。但对于大规模的状态空间来说，它是不可操作的。例如，在西洋双陆棋游戏中，将涉及处理大约个未知量的个方程。

### 时序差分学习

求解前一节内在的MDP并不是让贝尔曼方程承担学习问题的唯一方法，另一种方法是使用观察到的转移来调整观察到的状态的效用。

**基本思想：将效用估计朝着理想均衡方向调整，当效用估计正确时理想均衡是局部成立的。**

它的学习速度不如ADP Agent快，而且表现出更高的易变性，但是它更简单，每次观察所需的计算量也少得多。TD不需要一个转移模型来执行其更新。环境以观察到的转移的形式提供了相邻状态之间的联系。

ADP和TD方法是密切相关的。二者都试图对效用估计进行局部调整，以使每一状态都与后继状态相“一致”。一个差异在于TD调整一个状态使其与已观察到的后继状态相一致，而ADP则调整状态使其与所有可能出现的后继状态相一致，根据概率进行加权。由于转移集合中的每个后继状态的频率与其概率近似成正比，所以当TD调整的影响在大量的转移上计算平均的时候，上述差异便会消失。一个更重要的差异是，TD对每个观察到的转移都只进行单一的调整，而ADP为了重建效用估计和环境模型之间的一致性会进行尽可能多的所需调整。所以，TD可以被视为对ADP的一个粗略而有效的一阶近似。

## 主动强化学习

被动学习Agent有固定的策略决定其行为，主动学习Agent必须自己决定采取什么行动。

Agent同时必须还学习要做什么。主要的问题是探索：为了学会如何在环境中行动，Agent必须尽可能多地经历所处环境。

### 探索

**贪婪Agent：Agent在每一步都遵循其所学模型的最优策略的建议，在经历多次试验后，Agent一直坚持某个策略，再没有学习其他状态的效用，也从没有发现最优路径，称此Agent为贪婪Agent。**

greedy agent选择最优行动导致非最优结果，是因为学习到的模型与真实环境并不相同。ga忽视的是：行动不仅仅根据当前学习到的模型提供回报，也通过影响所接收的感知信息对真实模型的学习做出贡献。通过改进模型，Agent将在未来得到更高的回报。因此，一个Agent必须要在充分利用信息以最大化回报——反映在其当前效用估计上——和探索以最大化长期利益之间进行折中。

### 学习行动-效用函数

Q-学习的时序TD方法，它学习一种行动-效用而不是学习效用。用代表在状态进行行动的价值。Q-值与效用值直接相关：

学习Q函数的TD Agent不需要一个用于学习或行动选择的模型。另一方面，时序差分方法不需要状态转移模型——它只需要Q-值。时序差分Q-学习的更新公式为：

Q-学习能学习最优策略，但学习速度远远低于ADP Agent。这是因为局部更新不通过模型强制保持Q-值之间的一致性。

## 强化学习中的泛化

Agent学习到的效用函数和Q-函数是通过每个输入对应一个输出值的表格形式表示的。对于小规模的状态空间，这种工作方法效果很好。但随着空间的增大，收敛时间以及每次迭代的时候都会迅速增加。

在大规模状态空间中，为了在状态上进行泛化，一种方法是应用函数逼近。强化学习算法能够学习参数 ，以使评价函数 逼近真实效用函数。通过函数逼近器所获得的压缩允许学习Agent能由它访问过的状态向未访问过的状态进行泛化。

另一方面，所选择的假设空间内可能不存在任何函数能够对真实的效用函数进行充分好的近似。正如在所有的归纳学习中一样，在假设空间的大小和它对函数进行学习需要花费的时间存在着折中。较大的假设空间增加了找到一个好近似的可能性，但也意味着收敛可能被延迟。

函数逼近对于学习环境模型也是非常有帮助的。对于一个部分可观察的环境，学习问题要困难得多。如果我们知道隐变量是什么，而且知道它们之间和它们与可观察变量之间有什么样的因果联系，那么我们就能固定一个动态贝叶斯网的结构并使用EM算法来学习参数。

## 策略搜索

**策略搜索的思想是：只要性能还在改进就能保持对策略的调整，然后停止。**

策略搜索方法直接在策略的一个表示上进行操作，试图基于观察到的性能表现而对其进行改进。

## 强化学习的应用

* 时序差分西洋双陆棋有力地例证了强化学习技术的潜力。
* 小车连杆问题通过使用根据观察到的回报变化而适应性地对状态空间进行分割的算法可以获得改进的泛化和更迅速的学习。这项技艺已经远远超出了大多数人的能力。
* 直升机飞行问题使用策略搜索以及基于一个学习好的转移模型的进行仿真的PEGASUS算法，其性能远远超过一个人类专业驾驶员进行遥控的表现。

# 自然语言处理

人类因为具有语言的能力而区别于其他物种。我们想要计算机Agent能够处理自然语言，主要有两个原因：第一，使之能够与人类交流；第二，使之能够从书面文字中获取信息。

互联网上已有超过万亿数量的信息网页，而几乎所有这些页面都是用自然语言描述的。Agent想要获取知识，就需要理解人们所使用的具有歧义、杂乱的语言。具体的信息查找任务分为：文本分类、信息检索和信息抽取。解决这些问题的一个共同要素是采用语言模型：该模型用来预测语言表达的概率分布情况。

## 语言模型

自然语言不能描述为一个确定的语句集合，因此，通过句子的概率分布来定义自然语言模型要比通过确定集合来定义更为有效。

### n元字符模型

长度为n的书写符号序列称为**n元组**。通常用”unigram”、”bigram”、”trigram”分别表示一元组、二元组、三元组。

n个字符序列上的概率分布称为**n元模型**。

n元模型可以定义为n-1阶**Markov链**。在Markov链中字符的概率只取决于它前面的字符，而与其它字符无关。一个三元模型中有：

在三元模型中，运用Markov假设来定义字符序列的概率：

n元模型非常适合做语言识别的任务：给定一段文本，确定它是用哪种自然语言写的。

字符模型还可以完成其他任务，包括拼写纠错、体裁分类、命名实体识别。

### n元模型的平滑

n元模型的主要问题在于训练语料只提供了真实概率分布的估计值。有些非常罕见的单词可能会导致整个文本序列的概率为0，这显然是不合理的。我们要改进语言模型，使得在训练文本库中出现概率为0的序列会被赋予一个很小的非零概率值。这种调整低频计数的概率的过程叫做**平滑**。

常用的平滑方法有如下三种：

* Pierre-Simon Laplace平滑：如果一个随机布尔型变量在目前已有的个观察值中恒为false，那么的估计值应为 。
* 回退模型：首先进行n元计数统计，如果某些序列的统计值很低（或为零），就回退到（n-1）元。
* 线性插值平滑：一种通过线性插值将三元模型、二元模型和一元模型组合起来的后退模型。其定义概率估计值如下：

### 模型评估

我们使用交叉验证来评估一个模型。将语料分为训练语料和验证语料。先从训练数据中确定模型的参数值，然后再使用验证语料对模型进行评估。

使用复杂度来度量描述序列概率，它定义为：

复杂度可以看成是用序列长度进行规格化的概率的倒数，也可视为模型的分支系数的加权平均值。如果某些字符的可能性高于其他字符，而模型又能够反映这一点，那么这个模型的复杂度机会小于平均可能性。

### n元单词模型

单词模型和字符模型有着相同的机制，主要区别在于词汇——构成语料和模型的符号集合——比字符模型更大。

n元单词需要处理词汇表之外的单词。在字符模型中，我们不必担心有人发明字母表中的新字母。但在单词模型中，总有可能出现训练语料中没有的单词，所以我们需要在语言模型中明确地对其建模。

## 文本分类

**文本分类的任务是：给定某个文本，判断它属于预定义类别集合中的哪个类别。**

谈到归类，我们有两种方法。一种是语言模型方法，我们可以对垃圾邮件文件夹里的邮件进行训练，从而得到一个计算的n元语言模型；对收件箱里的邮件进行训练，可以得到计算的模型。然后我们可以应用贝叶斯规则来对新消息进行分类：

另一种方法是机器学习方法。我们把邮件信息看成是一组特征/值对，分类算法h根据特征向量进行判断。我们可以将n元组作为特征，这样语言模型和机器学习就可以融合了。

算法在巨大的特征向量工作将花费高昂的代价，因此特征选择用来挑选那些最能够区别垃圾邮件和非垃圾邮件的特征。通常来说，挑选最好的一百种左右的特征，就可以很好地区分不同类型了。

### 数据压缩的分类方法

分类问题可以看成是一个数据压缩。无损压缩算法可以在一串符号序列中检测其中的重复模型，然后重写一段比原串更为紧凑的符号序列。

实现压缩算法，首先要构建文本的子序列词典，然后引用词典中的条目。压缩算法实际上是在建立一种语言模型。为了通过压缩进行分类，我们首先把所有垃圾邮件的训练消息和在一起并压缩成一个单元，对于非垃圾邮件也作同样的处理。当给定一个要分类的邮件时，我们把它加到垃圾邮件集合中，再对更新后的集合做压缩。同样的，把它加到非垃圾邮件并做压缩。哪一个压缩更好，该邮件就属于那个类别。

基于压缩的文本分类的准确率与传统的分类算法相当，在运行速度上要比传统分类算法慢。

## 信息检索

**信息检索的任务是：寻找与用户的信息需求相关的文档。**

一个信息检索系统具有如下特征：

* 文档集合
* 使用查询语言描述的查询
* 结果集合
* 结果集合的展示

早期的IR系统采取布尔关键字模型工作。文档集合中的每个词都被当作一个布尔特征，如果这个词出现在某文档中，那个该文档的这个特征值为真，反之为假。布尔模型的优点在于容易解释和实现，但也存在一些缺点：

* 由于文档的相关度只用一个二进制位表示，所以无法为相关文档的排序提供指导。
* 对于非程序设计人员和非逻辑学家的用户来说，他们并不熟悉布尔表达式。
* 即使是对熟练的用户来说，写一个是适当的查询也可能是困难的。

### IR评分函数

**评分函数**：根据文档和查询计算并返回一个数值得分，最相关的文档得分最高。

在BM25函数中，得分由构成查询的每个单词的得分进行线性加权组合而成。有三个因素会影响查询项的权重：第一，查询项在文档中出现的频率；第二，词项的文档频率的倒数；第三，文档的长度。

BM25将上述三个因素都考虑在内。但是，对语料库中的每个文档都计算BM25评分函数是不现实的。相反，对于词汇表的每个单词，系统预先创建了索引，列出包含该单词的所有文档，被称为单词的**命中列表**。当给定一个查询时，我们对查询中的各单词的命中列表取交集，并对交集中文档计算评分就可以了。

### IR系统评价

**系统评价**：交给系统一组查询，人工对系统返回的结果集合进行相关性判断并评分。

传统上，在评分时有两个度量指标：准确率和召回率。

**准确率**：结果集合中实际相关的文档所占的比例。

**召回率**：结果集合中的相关文档在整个语料库的所有相关文档中所占的比例。

：综合上述两个指标的度量，是准确率和召回率两者的几何平均值。

### IR的改进

一种常见的改进是采取一种更好的文档长度对相关性的影响模型。

**枢轴文档长度标准化模式**：取一个文档长度的枢轴点，文档长度等于点值则采取原来的标准化方法，比该点值小就会增加，比该点值大就会减少。

**元数据**：文档文本之外的关于文档的数据。

### PageRank算法

**网页排名的任务是：解决TF评分问题。**

网页排名算法设计时会赋予来自高质量的网站的链接更高的权重。页面的PageRank定义为：

其中，是页面的网页排名，是语料库中总的网页数量，是链接到的页面，是页面链接出去的链接数。常量是阻尼因子。

### HITS算法

HITS算法也称为中心权威算法，它是另一个颇有影响力的链接分析算法。

和PageRank一样，该算法这样迭代执行：把页面的权威得分更新为指向它的所有页面的中心得分的总和，并把中心得分更新为它指向的所有页面的权威得分的综合。如果对得分进行标准化，并重复次，该过程便会收敛。

它许多方面与PageRank有所不同：它是一种依赖于查询的度量方法：它针对给定的查询对网页进行评估。给定一个查询，HITS首先找到一个与查询相关的网页集合，通过对查询的单词的命中列表作交集，然后增加这些网页的相邻网页。

### 问题回答

**问题回答的任务是：一个问题作为查询，一个简短的句子或短语作为回答。**

ASKMSR系统是一个典型的基于万维网的问题回答系统。它基于这样的直觉：在万维网上大部分问题都会被多次回答，所以问题回答应该看成关注准确率问题而不是召回率问题。

ASKMSR依赖万维网上内容的广度而不是其自身理解的深度。虽然它没有处理复杂推理模式的能力，但它知道万维网的内容规模是如此巨大，以至于它可以忽视这些复杂段落，而只从它能处理的简单段落中寻找答案。

## 信息抽取

**信息抽取的任务是：通过浏览文本获取特定类别的对象以及对象之间的关系的过程。**

### 基于有限状态自动机的信息抽取

基于属性的抽取系统假设整个文本都是关于单一对象，而系统的任务就是抽取该对象的属性。对于这类问题，我们可以针对每个需要抽取的属性定义一个模板，而模板可以用有限状态自动机定义。模板通常由三部分组成：前缀正则表达式、目标正则表达式和后缀表达式。

更进一步的是关系抽取系统，它处理多个对象以及它们之间的关系。关系抽取系统可以由一组级联有限状态转换器构成。FASTUS由以下五个阶段组成：

1. 符号分析：将字符流分割成一个个符号。
2. 处理复合词：复合词可以通过结合词典条目和有限状态语法规则进行识别。
3. 处理基本词组：将它们分成组块，以便于后续阶段的处理。
4. 将基本词组组合成复合短语。
5. 合并前一步产生的结构。

基于有限状态模板的信息抽取方法在受限领域中效果较好，因为在受限领域中可能预先确定讨论的主题及其表达方式。采用级联转换器模型，有利于对所需知识进行模块化，便于构建系统。但在格式变化较大的领域，可能涉及范围更广的主题，有限状态信息抽取方法就很难获得成功。

### 信息抽取的概率模型

**隐马尔可夫模型（HMM）**：一种处理带有隐含状态的序列的最简单的概率模型。

为了将HMM应用于信息抽取，可以为所有属性建立一个大的HMM，也可以为每个属性分别建立一个独立的HMM。本文采用第二种方法，观察值序列就是文本的单词序列，隐含状态分别表示处于属性模板的目标、前缀或后缀部分，或者是背景部分。

HMM相比FSA有两大优势：第一，HMM是概率模型，可以抗噪声；第二，HMM可以用数据训练得到，而无需构造模板的繁重工程。

### 基于条件随机场的概率模型

**条件随机场的任务是：对于给定的观察变量集合，该模型对一组目标变量的条件概率分布进行建模。**

一种常见的结构是线性链条件随机场，可以表示时间序列中变量之间的马尔可夫依赖关系。状态特征函数和转移特征函数是CRF的关键组成部分。

CRF与HMM有如下几点不同：

* HMM是生成模型，而CRF是判别模型。
* 由于独立性假设的原因，HMM不能使用重叠的特征，CRF可以。
* CRF中的特征能使用序列中的任何部分。
* CRF的特征能定义到状态的转移上。
* CRF比HMM更灵活，准确率更高。

### 从大型语料库中抽取本体

信息抽取另一种不同的应用就是建立一个大型的知识库或从语料中抽取事实性主体。

这一应用有三方面的不同：

* 它是无限制的：我们需要获取关于所有领域的事实，而不仅限于某一领域。
* 对于大型语料库，这个任务强调的是准确率而不是召回率。
* 结果是从多个来源统计汇集而成，而不仅从某一特定文本中抽取。

### 自动模板构建

给定一个匹配集合，一个简单的模板是生成框架就可以找到能解释这些匹配的模板。模板语言与匹配有紧密的映射关系，适合自动学习，并强调高准确率。

如果最初很少的几个模板中有一个不正确，错误就会迅速传播。限制这个问题的一种方法是，不接受新的示例，除非经过了多个模板验证；不接受新的模板，除非它发现了多个其他模板也发现的示例。

### 机器阅读

机器阅读的任务是：构建一个对于各种关系都无需人工输入的信息抽取系统，这个系统能够阅读并构建自己的数据库。这样的系统是关系独立的，对任何关系都适用。

TEXTRUNNER是一个代表性的机器阅读系统。它利用协同训练来提升性能，但它需要一些基础进行引导而逐步提升。

TEXTRUNNER已经从一个包含五亿网页的语料库中提取了数亿条事实信息，在大规模万维网语料上的准确率达到了88%，召回率达到了45%，由此可见，我们正逐步接近机器阅读的目标。