# 最大熵模型的树-栅格最优 N 解码算法*)

冯　冲[1,2]　陈肇雄[2]　黄河燕[2]　王江伟[2,3]
(中国科大计算机系 合肥 230027)[1]　(中科院计算机语言信息工程中心　北京 100083)[2]
(南京理工大学计算机系　南京 210094)[3]

**摘　要**　最大熵模型已被广泛应用于多种自然语言处理任务，但一些现有研究工作在解码算法上存在有待改进的地方。本文提出了一个最大熵模型的树-栅格最优 N 解码算法，并对算法性能进行了分析和比较。算法的另一优点在于可以在解码过程中检测并控制潜在的标注冲突。

**关键词**　树-栅格算法，最大熵模型，解码

## A Tree-Trellis N-Best Algorithm for Decoding in Maximum Entropy Models

FENG Chong[1,2]　CHEN Zhao-Xiong[2]　HUANG He-Yan[2]　Wang Jiang-Wei[2,3]
(Dept. of CS, USTC, Hefei 230027)[1]　(CCLIE, CAS, Beijing 100083)[2]　(Dept. of CS, NJUST, Nanjing 430074)[3]

**Abstract**　Maximum entropy models have been widely adapted in various natural language processing tasks. But there are some deficiencies in the decoding algorithm used by many previous researches. A n-best tree trellis algorithm is proposed for decoding in maximum entropy models. The performance analysis and comparison with other decoding algorithms are also presented. Another advantage of our method is that the possible collision in action sequences can be detected and eliminated.

**Keywords**　Tree trellis algorithm, Maximum entropy models, Decoding

## 1　引言

最大熵模型表现出很强的知识表达能力，已逐渐成为自然语言处理领域建立统计语言模型的有效方法之一。A. L. Berger 等人比较详细地介绍了最大熵的理论框架，并讨论了其在基于统计的机器翻译领域的一些应用；Borthwick 研究了基于最大熵模型的命名实体的识别；Ratnaparkhi 应用最大熵进行了句子边界识别、词性标注、Chunk 等浅层句法分析的研究。在中文领域，周雅倩等利用最大熵模型研究了名词短语识别，李素建等对中文组块的识别和划分进行了研究，并都取得了很好的效果。

最大熵模型的应用由建模、训练和解码三个环节构成。如上面介绍的，此领域研究工作主要集中在参数训练算法、特征选择算法，以及最大熵模型在特定自然语言处理任务中的使用等研究。解码环节一直被较少关注。但我们发现，一些工作在解码环节上存在着可以进一步改进之处[1,2]。而解码算法能否给出全局最优解，能否给出最优 $N$ 解，直接影响着最大熵模型的性能。为此，本文研究并给出了一种计算最大熵模型最优 $N$ 解的树-栅格解码算法。算法时间复杂度对文本长度的依赖是线性的。其另一优点在于能够通过控制标注序列中的冲突改善系统性能。算法应用在我们的命名实体识别系统 MulNERec 中，取得了良好的实验结果[3]。

## 2　问题描述

根据最大熵原理，在所有符合已知事实（训练样本）的概率分布中，应当选择熵最大的那个分布。即选择概率分布 $p^*$，使得 $p^*=\underset{p\in P}{\operatorname{argmax}}H(p)$，其中 $H(p)$ 为模型 $p$ 的熵。而 $p^*$ 的解的数学形式为

$$\begin{cases} p^*(a|b)=\dfrac{1}{Z(b)}\prod_{j=1}^{k}\alpha_j^{f_j(a,b)} \\ Z(b)=\sum_{a\in A}\prod_{j=1}^{k}\alpha_j^{f_j(a,b)} \end{cases} \tag{1}$$

其中的 $f_j(a,b)=\begin{cases}1,\text{如果 } a,b \text{ 满足某种条件}\\0,\text{否则;}\end{cases}$ $\alpha_j$ 为每个特征函数的参数，代表每个特征的重要性；$Z(b)$ 是归一化因子。更详细的数学推导参见文[4]。

学习器的训练过程中的两个主要计算问题是参数估计和特征选择。参数估计可以使用 GIS 算法、IIS 算法、L-BFGS 算法等，特征选择可以通过简单的设置特征函数频度阈值或通过递增学习不断选取最有区分度的特征集来实现。在训练结束后，就可以根据式(1)进行解码。

学习器的解码过程可以看作是一个标注问题。即在给定输入序列 $W=\{w_1,w_2,\cdots,w_n\}$ 的情况下，依据训练得到的特征函数集 $f_i(a,b)$ 及其参数 $\alpha_i$，求解行动序列 $A=\{a_1,a_2,\cdots,a_n\}$，其中 $a_i$ 的值域为类集或标注集 $C=\{c_1,c_2,\cdots,c_{|C|}\}$，$b_i$ 为输入序列中每一个元素 $w_i$ 的上下文环境信息。也可以把这一过程看作在候选标记序列空间内搜索具有最大概率的标注结果序列，这个序列的概率为

$$\hat{A}=\underset{A}{\operatorname{argmax}}P(a_1\cdots a_n|w_1\cdots w_n)=\underset{A}{\operatorname{argmax}}\prod_{i=1}^{n}p(a_i|b_i)=\underset{A}{\operatorname{argmin}}\{-\log P(a_1\cdots a_n|w_1\cdots w_n)\}=\underset{A}{\arg\min}\{\sum_{i=1}^{n}(-\log p(a_i|b_i))\} \tag{2}$$

其中，当前元素的各种行动的概率 $p^*(a_{ij}|b_i)$，$a_{ij}\in A$，可以通过式(1)计算出来。在具体实现算法时，为了避免计算过程中的浮点溢出，缩短计算时间，对式(2)取负对数处理。

*)受国家自然科学基金(编号 60272088)资助。冯　冲　博士研究生，主要研究方向为统计方法的信息抽取和机器翻译。陈肇雄　研究员，主要研究方向为机器翻译。黄海燕　研究员，主要研究方向为机器翻译。王江伟　硕士生，主要研究方向为信息抽取。

## 3 树-栅格最优 N 解码算法

一种最常见的算法是如文[1,2]中采用的对输入序列 $W=\{w_1,w_2,\cdots,w_n\}$ 从左向右地进行标注的算法。即对其中的每一元素 $w_i$，把前面的标注结果填充到当前词上下文特征中，直接取 $a_i^*=\arg\max_{1\leqslant j\leqslant|C|}p^*(a_{ij}|b_i)$ 作为当前位置的解码结果，这样依次处理就完成了解码输出序列的计算。

如果相邻位置的元素的解码结果相互没有影响，从每个局部最优解得到的行动序列也就是式(2)的全局最优解。但在很多自然语言处理问题中，前一元素的行动 $a_i^*$ 是后继元素的上下文 $b_{i+1}$ 的组成部分，因此影响着后继元素的 $p^*(a_{(i+1)j}|b_{i+1})$ 的计算。换言之，在 $p^*(a_{ij}|b_i)$ 取得最大的情况下选择 $p^*(a_{(i+1)j}|b_{i+1})$ 中的最大者，并不能保证得到一个使全局目标函数式(2)最大的行动序列。

为了克服上述算法的不足，我们设计了一种基于 Viterbi 算法思想的动态规划解码算法。定义局部最优变量 $\vec{\delta}(i)=(\delta_{c_1}(i),\delta_{c_2}(i),\cdots,\delta_{c_{|C|}}(i))$，向量中的每一项表示在位置 $i$ 时把 $w_i$ 标注为 $c_1,c_2,\cdots,c_{|C|}$ 的最大概率。$\vec{\delta}(i)$ 满足如下的递归关系，使得我们能够应用动态规划：

$$\delta_k(i+1)=\max_{k,l\in C}\{\delta_l(i)\times p(k|b_{i+1},l)\} \tag{3}$$

式中，$l,k\in C$，分别表示位置 $i$ 和位置 $i+1$ 上的标注结果。定义与局部最优变量相对应的回退指针变量 $\vec{E}(i)=(E_{c_1}(i),E_{c_2}(i),\cdots,E_{c_{|C|}}(i))$，它记录当 $\delta_k(i+1)$ 为当前位置 $i+1$ 的最大值时，前一位置 $i$ 上的标注结果，即式(3)中的 $l$。具体过程如下：

1)初始化：$\delta_k(1)=\max_{k\in C}\{p(k|b_1)\}$，$E_l(1)=l$；

2)从左向右，$i=1,\cdots,n-1$，递推地计算具有最大概率的标注结果序列：

$$\delta_k(i+1)=\max_{k,l\in C}\{\delta_1(i)\times p(k|b_{i+1},l)\};E_k(i+1)=l$$

3)递推过程的终结：

$$\delta_k(n)=\max_{k,l\in C}\{\delta_l(i)\times p(k|b_{i+1},l)\};E_k(n)=l;$$

$$a_n=\arg\max_q\{\delta_q(n)\};$$

4)序列回溯：

根据最后的标注结果 $a_n$ 和回溯指针向量 $\vec{E}(i)$ $(1\leqslant i\leqslant n)$，回溯，得到最大概率的标注结果序列。

这样，算法通过引入具有递归性质的局部最优变量，把后继元素对前一元素的标注结果的依赖记录在回溯指针变量中，从而实现了全局最优序列的求解。

上面的动态规划算法可以得到全局最优解。但一些实验结果表明，保留最优 $N$ 结果而不仅仅是最优解，将更有利于提高最大熵模型的性能[5]。我们对前面的动态规划算法加以进一步改进，设计了最大熵模型的树-栅格解码算法。

算法由同步的前向栅格搜索和异步的后向树搜索两部分构成。其中的前向栅格搜索是在前面给出的动态规划算法的基础上得到的。和前述算法的不同之处在于，搜索过程中把所有局部路径的分值记录在数据结构中。然后使用后向的树搜索对局部路径加以扩展。数据结构设计如下：

用 *maxent* 数据类型中存储各元素的信息以及到该元素为止的局部最优路径。*maxent.start* 和 *maxent.end* 分别指向该元素的起始位置和结束位置。*maxent.tag* 存放元素的行动(即标注结果)，其值域随不同任务而变化，例如在命名实体任务中可采用 BIO 方式的标记体系。*maxent.n_state* 记录包括当前标注结果在内的最末两个或三个标注结果(在考虑当前元素的标注受前一元素或前两元素标注结果影响的情况下)。*maxent.prob_current* 存储从第一个元素到该元素的最优局部路径的分值。*maxent.previous* 指向前一局部最优节点，此信息可用于反向选取最优路径，但在引入了后向的最优 $N$ 搜索后并不是必须的。

*maxent_list* 是一个线性表。该表以每个节点的 *maxent.end* 为索引项，其中的各个 *maxent* 数据类型的节点构成了到当前结束位置的最优局部路径。函数 *set_maxent_list* 和 *get_maxent_list* 分别用于生成和读取当前结束位置的最优局部路径。

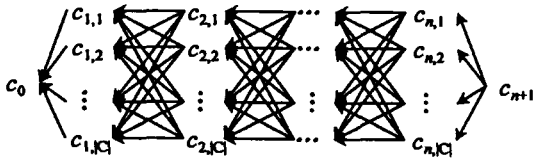


图 1 最大熵解码过程中的启发式后向树搜索

后向的树搜索使用启发式搜索算法，如图 1 所示(对启发式搜索和 $A^*$ 算法的一般性介绍可参考文[6])。每个 *maxent* 结构作为启发式搜索中的一个状态。反向搜索过程从输入序列的最末端元素开始，使用 *maxent_list* 每步扩展反向的局部路径，一直回溯到起始元素，这样就顺序得到最优 $N$ 标注结果。

我们采用的启发式估计函数的形式为 $f(i,j)=g(i,j)+h(i,j)$，$(1\leqslant i\leqslant n,1\leqslant j\leqslant n)$。式中，$g(i,j)$ 表示目前已知的由最末节点回溯到当前节点的最短路径 $g^*(i,j)$ 的估计；$h(i,j)$ 表示当前节点到起始节点的最终路径长度 $h^*(i,j)$ 的启发式估计值。两函数的计算式如下：

$$g(i,j)=\hat{g}(i+1)+e_B(i,j)$$

$$h(i,j)=\min_k(e_F(i,j,k)+h(i-1,k)) \tag{4}$$

其中，$\hat{g}(i+1)$ 的计算由式(5)得到，它对应从实际的最末节点到 $i+1$ 节点的部分最优路径。由于输入边界的存在，$g(n+1)=h(0)=0$。

$$\hat{g}=\min_j g(i,j)=\hat{g}(i+1)+\min_j e_B(i,j) \tag{5}$$

耗费函数 $e_B(i,j)$ 和 $e_F(i,j,k)$ 的计算见式(6)，分别表示后向搜索和前向搜索的相应的父子节点间的代价。

$$e_B(i,j)=-\log P(a_{ij}|b_i) \tag{6}$$

$$e_F(i,j,k)=-\log P(a_{i-1,k}|b_{i-1})$$

根据式(5)可知，对于任意相邻反向节点对 $((i,j)(i-1,k))$，不等式 $h(i,j)\leqslant h(i-1,k)+\phi_F(i,j,k)$，且有 $h(0)=0$，因此以上启发式函数满足 A* 单调性限制条件，在搜索到 $w_i$ 时就得到最有可能的行动，避免了重复搜索，即 $\hat{a}_i=\arg\min_{j=1,\cdots,|C|}f(i,j)$。

## 4 算法分析

### 4.1 时间性能分析

算法的运行时间由前向栅格搜索(动态规划算法)和异步的后向树搜索(启发式算法)两部分构成。前者取决于递推的次数，即输入序列的长度 $n$，和每步递推计算所耗的时间。在每步递推计算中，由于要在所有合法的相邻序列中搜索 $\delta_l(i)\times p(k|b_{i+1},l)$ 的最优解，因此每步递推计算的时间开销受 $|C|^2$ 制约。具体解码任务中，标记集合的大小 $|C|$ 通常是一个不大的常量。后者的时间开销为 $O(n|C|)$。这样，算法总体时间复杂度为 $O(n|C|^2)$，即主要由动态规划算法决定。

### 4.2 算法特点

本文算法除了可以解出最大熵模型的全局最优序列之外，还有两个特点。

第一，可以在不降低系统时间性能的条件下得到最优 $N$ 结果。文[5]等研究了保留最优 $N$ 结果对于基于最大熵模型的句法分析的影响，实验结果表明，当 $N$ 增大时句法分析器的性能会不断改善。

第二，可以简便地解决相邻标注结果的冲突问题。在每步计算式(3)中的 $p(k|b_{i+1},l)$ 时，过滤掉相互冲突的 $k$ 和 $l$，就可以保证当前的所有候选序列中不含冲突结果。这样最后通过回溯找到的结果就是所有无冲突候选序列中的最大概率标注序列。

## 5 实验

我们在基于最大熵模型的组织机构名识别系统中进行了实验[3]。输入长度为 $n$ 的含分词和词性标注信息的文本 $W=\{w_1,w_2,\cdots,w_n\}$，并把输入序列中每一个元素 $w_i$ 的上下文环境信息记作 $b_i$。$b_i$ 包括当前词左右各两个词及其词性，以及前一个词的命名实体标注结果。标记集合采用 $C=\{B,I,O\}$。组织机构名标注问题也就是根据特征函数集 $f_i(a,b)$ 及其参数 $\alpha_i$ 的情况，搜寻使得 $P(E|W,F)$ 最大的标注结果序列 $A^*=\{a_1,a_2,\cdots,a_n\}$，其中 $a_i\in C$，即 $A^*=\arg\max_{a_i\in C}\{\prod_{i=1}^{n}P(a_i|b_i)\}$。

实验数据由来自两个语料库的含“[]$nt$”标记的句子构成。一部分从北大-富士通的 1 个月的人民日报标注语料(199801)中选出；另一部分选自兰开斯特语料库的新闻报道(LCMC_A)、新闻社论(LCMC_B)和新闻评论(LCMC_C)部分，详见表 1。

表 1 实验数据

| 语料来源 | 语料规模 | $C_1$—训练用 | | $C_2$—测试用 | |
|---|---|---|---|---|---|
| | | 比例 | 样本个数 | 比例 | 样本个数 |
| 1998_01 | 8419 | 90% | 7579 | 10% | 840 |
| LCMC_ABC | 2386 | 90% | 2146 | 10% | 240 |
| | 10805 | 90% | 9725 | 10% | 1080 |

模型训练过程采用设置频数门限的方法进行特征选择，采用 L-BFGS 算法进行参数估计。

实验考察的性能指标沿用了 MUC 和 MET 中定义的度量方式：查准率 $p=N_3/N_2$、查全率 $r=N_3/N_1$，以及二者的调和均值：$F=(2\times p\times r)/(p+r)$。其中 $N_1$ 为人工标注结果中组织机构名的总数，$N_2$ 为系统切分结果中组织机构名的总数，$N_3$ 为系统切分结果中的正确切分出的组织机构名(与人工标注结果的标注结果完全相同的词)的总数。

实验对比考察了局部最优解码算法、动态规划解码算法和树-栅格最优 $N$ 解码算法在同一最大熵模型上得到的标注结果，如表 2 所示。

实验系统的语料库规模较小，也没用采用其他技术手段，整体性能不高。但它已能满足对比分析解码算法的要求。实验结果表明，解码算法的改进对实验效果的影响是显著的。

表 2 解码算法的对比实验结果

| 解码算法 | $P$% | $r$% | $F$% |
|---|---|---|---|
| 局部最优解码 | 36.8 | 66.2 | 47.3 |
| 动态规划解码 | 38.4 | 67.9 | 49.1 |
| 树栅格最优 $N$ 解码($N$=5) | 44.2 | 73.6 | 55.2 |

## 6 与相关研究的比较

近年来，国内研究人员已经在基于最大熵模型的中文信息处理方面作了很多有价值的工作。如文[1]利用最大熵模型研究了名词短语识别，文[2]对中文组块的识别和划分进行了研究，分别取得了很好的成果。但是，第一，所采用的解码算法都是从左到右依次标注每一个词。每步计算 $p^*(a_{ij}|b_i)$ 得到的标注结果是局部最优的，但不能保证得到 $\Pi_{1\cdots n}(p^*(a_{ij}|b_i))$ 的最大值及其对应的标注序列。第二，局部最优算法只保留当前位置的最大概率及其行动，不能给出最优 $N$ 结果。第三，没有讨论标注序列中可能存在的冲突，而例如类似于“…$OI$…”这样的错误标注序列是应当被过滤的。

国外文献中，文[5]研究了采用最大熵模型的句法分析器，和本文的基本假设不同，他们提出了一种基于广度优先搜索的解码算法。在输入序列长度为 $n$，标记集合大小为 $|C|$，保留最优 $N$ 结果的情况下，算法的时间复杂度为 $O(nN|C|\log(N|C|))$。文[7]在用决策树方法处理日文命名实体时提到了采用 Viterbi 算法处理标记序列中的冲突，但并未详细讨论。

结论 本文讨论最大熵模型的解码问题。针对现有算法的不足之处，设计了一种用于最大熵模型解码的动态规划算法，然后在此算法基础上进一步改进，提出了最大熵模型的树-栅格最优 $N$ 解码算法，并给出了实验结果和对比分析。算法的优点在于，它不仅可以在随文本长度线性增长的时间复杂度内得到全局最优解，而且可以得到最优 $N$ 解。此外它能够判断相邻状态是否合法，解决了行动序列(标注结果)中潜在的冲突问题。

算法也存在着一些需要进一步改进的地方。其一是算法实现会较为复杂。当特征函数中的 $b_i$ 受 $a_i,a_{i-1}$ 两个状态的影响时，算法的每步递推计算需要处理的计算规模变为 $|C|^3$，时间性能会下降为 $O(n|C|^3)$。

最优 $N$ 解码算法为我们研究组织机构名的主动学习提供了必要的支持。相关工作在文[3]中介绍。进一步的工作将集中在把算法改进后用于 IHSMTS[8] 的最长名词短语识别模块。

致谢 模型训练部分使用了东北大学张乐提供的 L-BFGS 参数估计算法模块，在此表示感谢。

## 参考文献


1 周雅倩，郭以昆，黄萱菁，吴立德. 基于最大熵方法的中英文基本名词短语识别[J]. 计算机研究与发展，2003，40(3)：440～446

2 李素建，刘群，杨志峰. 基于最大熵模型的组块分析[J]. 计算机学报，2003，26(12)：1722～1727

3 冯冲，陈肇雄，黄河燕. 采用主动学习策略的组织机构名识别. 小型微型计算机系统，2004(已录用)

4 Berger A L，Pietra B，Pietra V. A Maximum Entropy Approach to Natrual Language Processing [J]. Computational Linguistics，1996，22 (1)：39～71

5 Ratnaparkhi A. Maximum Entropy Models For Natural Language Ambiguity Resolution：[Ph. D Thesis]. University Of Pennsylvania，1998

6 Nilsson N J. Artificial Intelligence，A New Synthesis，Morgan Kaufmann，New York，1998

7 Sekine S. NYU system for Japanese NE - MET2 [C]. In：Proc. of the 7th Message Understanding Conference (MUC-7)，1998. 114～120

8 黄河燕，陈肇雄. 基于多策略的交互式智能辅助翻译平台总体设计. 计算机研究与发展，2004，41(7)：1266～1272