# 从第一原则推导出Softmax

**——Softmax与sigmoid函数：Softmax原理推导**

## 第一章 引言

这篇文章的最初目标是探讨softmax和sigmoid函数之间的关系。事实上，这种关系似乎一直不能达到：“一个在分子中有一个指数！一个有一个总和！一个在分母中有一个1！”当然，这两者有着不同的名字。

一旦衍生出来，受到条件概率公理本身启发，我很快就意识到，如何将这种关系推回到更一般的模型框架中。因此，这篇文章首先探讨了sigmoid是什么,它仅仅是 softmax 的一个特殊情况，以及在 Gibbs 分布，因数乘积和概率图模型中的每一个基础。接下来，我们继续展示这个框架如何进行自然扩展，去定义典型的模型类，如softmax回归，条件随机场，朴素贝叶斯和隐式马尔可夫模型。

## 第二章 目标

这是一个预测模型。它是一个接收输入并产生输出的菱形。

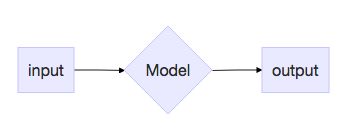


图1：预测模型

输入是一个向量 . 有3个可能的输出：.我们模型的目标是在输入条件下预测产生每个输出的概率，即：

当然，概率只是一个实数，位于封闭的区间 .

## 第三章 综述

### 3.1 输入如何影响输出？

我们的输入是4个数字的列表；每一个在不同程度上影响每一个可能的输出。我们称这种影响为“权重”。4个输入乘以3个输入等于12个不同的权重。他们可能看起来像这样：

表1：权重表

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | .1 | .4 | .3 |
|  | .2 | .3 | .4 |
|  | .3 | .2 | .1 |
|  | .4 | .1 | .2 |

### 3.2 产出一个输出

给定一个输入向量 . 我们的模型将使用上述权重为每个输出 产生一个数字。每个输入元素的效果将会被相加。之后再说明其原因。

这些总和将决定我们的模型产生什么样的结果。最大的数字获得胜利。例如，给定：

我们的模型将有最好的机会生产c。

### 3.3 转化为概率

我们之前说过，我们的目标是获得以下结果：

用粗体表示来任何输入值，鉴于我们现在具有特定的输入值，即 ，我们可以更准确地说明我们的目标：

到目前为止，我们只有 . 要将每个值转换成概率，即,在 上的一个非特殊的数字，仅仅除以总和即可。

最后，作为有效的概率分布，所有数字必须总和为1。

### 3.4 如果我们得到的值为负呢？

如果我们的初始非归一化概率之一是负的，即 .所有这些都不成立了。

不是一个有效的概率，因为它不会落入区间 .

为了确保所有非规范化概率都是正数，我们必须首先将它们传递给一个函数作为输入，这个函数将实数作为输入，并产生严格的正实数作为输出。这只是一个指数;现在我们选择欧拉数（）。 这个选择的理由将在后面解释（尽管注意到任何正指数将会满足我们所说的目的）。

我们的归一化概率，即有效概率现在看起来如下：

统一表示为：

这便是 softmax 函数。

### 3.5 与 sigmoid 函数的关系

softmax 在 个不同的输出上输出有效的概率分布，sigmoid 只是在 在情况下。因此，sigmoid 仅仅是 softmax 的一个特殊情况。通过这个定义，假设我们的模型只产生两个可能的输出 和 ，在给定输入 ，我们可以写出 sigmoid 如下所示：

但是，请注意，我们只需要计算 的概率，至于 在这个注释里，我们来重新展开表达式 ：

然后，将分子和分母除以 ：

最后，我们可以将其代回原来的补充：

我们的方程是不确定的，因为有比方程式（1个）更多的未知数（2个）。因此，我们的方程将有无数的解 。因此，我们可以直接固定这些值之一，设定

这便是 sigmoid 函数。最后：

### 3.6 为什么是非标准化的概率的总和？

我们都认为规范线性组合 的语义是理所当然的，但是我们为什么首先进行求和呢？

为了回答这个问题，我们首先重申我们的目标：在输入条件下预测产生每个输出的概率,即 , 接下来，我们将重新讨论条件概率本身的定义：

就个人而言，我觉得这有点难以解释。 让我们重新变形，以获得更直观的东西。

这可以解释为：

同时观察（给定值）A和B的概率，即，A和B的联合概率等于观察A发生的概率乘以在A发生的情况下观察B发生的概率。

例如，假设女孩出生的概率是0.55，并且一个女孩欢数学的可能性是0.88。因此，

现在，我们按照上面的定义重写我们的原始模型输出。

记住，我们指数化每一个非规范化概率 ，以将其转换为严格正数。从技术上讲，这个数字应该被称为 ,由于其值可能 ，因此不是一个有效的概率，因此，我们需要再为方程式链引入另外一个术语：

这是一个类似 的算术等式。

在等式左边：

* 分子是有效的联合概率分布。
* 分母为“观察 的任何值的概率”是1。

在等式右边：

* 分子是严格的正的非规范化概率分布。
* 分母是一个常数，可以确保

其和为1. 实际上，这个“规范化器”被称为分区函数; 我们将在下面回到这里。

考虑到这一点，让我们进一步分解我们的softmax方程的分子。

引理：鉴于我们的输出函数[1] 执行取幂，以便在可能的模型输出上获得有效的条件概率分布，因此我们对该函数[2]的输入应该是加权模型输入元素[3]的总和。

1. softmax 函数
2. 其中之一
3. 模型输入元素为 。加权模型输入元素是 。

不幸的是，如果我们首先能够把握住我们所遇到的 这样事实就好了。下面介绍 Gibbs 分布。

### 3.7 Gibbs 分布

吉布斯分布给出了一组结果的非规范联合概率分布，类似于在上面计算的 ,表示为：

其中 定义了一组因子。

#### 3.7.1 因子

一个因子是一个函数：

* 将一组随机变量列表作为输入。 该列表被称为该因子的范围。
* 返回每个 随机变量可以采用的值的 唯一组合 的值，即对于其范围内，交叉积空间中的每个条目。

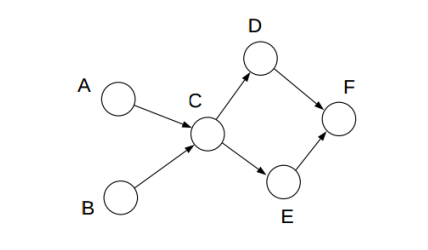
例如，一个范围为 的因子可能看起来像：

|  |  |  |
| --- | --- | --- |
| A | B |  |
|  |  | 20 |
|  |  | 25 |
|  |  | 15 |
|  |  | 4 |

#### 3.7.2 概率图模型 推测复杂系统的行为（通常）会计算其可能结果的联合概率分布。 例如，假设我们有一个商业问题，其中：

* 星期几（A）和营销渠道（B）影响客户注册的概率（C）。
* 顾客注册会影响我们的年度经常性收入（D）和年终雇佣预测（E）。
* 我们的ARR和雇佣预测会影响我们将为节日派对（F）订购多少蛋糕。

我们可以这样绘制我们的系统：



我们的目标是计算 。在大多数情况下，我们只会拥有我们系统小的子集数据；例如，我们曾经运行过一个受控实验来调查 其间的关系，或者问卷调查，他们喜欢在圣诞节吃多少蛋糕。如果完全不合理，获得一个相当复杂系统的联合概率分布，是非常少见的。

为了计算这个分布，我们把它分成几部分。每个部分都是描述系统某些子集细节行为的一个因子。（例如，一个因子可能会给出你在给定的一天观察到的次数（））。为此，我们说：

如果存在一组因子 ，则在图 上所需的非归一化概率分布 因子分解为：

其中 是 的推理图。

这个引理的前半部分只不过是重新定义非规范化的吉布斯分布。 下半部分进行扩展，我们注意到：

由一系列因子得到的推理图是一个相当完美的图，其中我们在因子域超集中的每个变量周围绘制一个圆圈，并在给定因子域中同时出现的变量之间绘制线条。

, 这两个因子的“因子域超集”为 。推理图将有用线连接的三个圆圈， 连接 ， 连接 。

最后，它接着是：

1. 给定带有变量的商业问题 --我们可以画该问题的一幅图。

2. 我们可以建立描述这个问题子集行为的因子。 实际上，这些只是小的子集。

3. 如果由我们的因子推理的图形看起来像我们所绘制的图形，我们可以将我们的系统表示为一个因子积。

不幸的是，在我们最初的模型中所产生的因子积结果 仍然是非规范化的，像 一样。

#### 3.7.3 配分函数

配分函数是分母，即 “归一化”，是我们的softmax函数。它用于将非规范化概率分布转换为归一化（即有效）概率分布。 真正的吉布斯分布如下：

其中 是配分函数。

要计算此函数，我们只需将非标准化表中的所有值相加。 给定 ：

|  |  |  |
| --- | --- | --- |
| A | B |  |
|  |  | 20 |
|  |  | 25 |
|  |  | 15 |
|  |  | 4 |

我们的有效概率分布变为：

|  |  |  |
| --- | --- | --- |
| A | B |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

这是softmax函数的分母。

### 3.8 Softmax 回归

再次申明，我们的模型的目标是在输入条件下预测产生每个输出的概率，即：

在机器学习训练数据中，我们给出了联合概率分布的构建模块。例如，观察共同投入和产出的分类帐。我们推测每个输入元素以不同的程度会影响每个可能的输出，即我们将它乘以一个权重。接下来，我们对每个结果 进行取幂，即因子, 然后乘以结果（或者，我们可以指数化因子的线性组合，即机器学习中的特征）：这给我们在所有（输入和输出）变量上的非规范化联合概率分布。

我们想要的是在输入条件下，在可能输出上一个有效的概率分布，即 。而且，我们的输出是一个单一的，在 中 “1-of-k”变量（与一系列变量相反）。这是softmax回归的几乎逐字定义。

Softmax回归也称为多项回归，或多类逻辑回归。二进制逻辑回归是softmax回归的一个特殊情况，与Sigmoid是softmax的特殊情况相同。

为了计算我们的条件概率分布，我们得到等式（1）：

换句话说，在输入条件下，产生每个输出的概率相当于：

1. softmax函数

2. 由配分函数归一化的输入元素的乘积的一个指数化因子。

#### 3.8.1我们的配分函数取决于

为了计算在条件 下，在 上的分布，我们的配分函数变成依赖 。换句话说，对于给定的输入 ，我们的模型计算条件概率 . 如果对softmax函数进行迂腐的重述，虽然这可能看起来是一个微不足道的事情，但是要注意的是，我们的模型是有效地计算一系列条件分布——每个对应独特的输入 。

### 3.9 条件随机场

以这种方式构建我们的模型，使我们自然地扩展到其他类别的问题。 想象一下，我们正在尝试为给定会话中的每个单词分配一个标签，标签可能包括：“neutral”，“offering an olive branch”和“them is fighting words”。现在我们的问题与原来的模型在一个关键的方式有所不同，和另一个可能的关键方式：

1. 我们的结果现在是一系列的标签。不再是“1-of-k”。在会话“hey there jerk shit roar”中可能的标签序列为：“neutral”，“neutral”，“them is fighting words”，“them is fighting words”，“them is fighting words”。

2. 单词之间可能存在关系，这可能会影响最终输出标签序列。例如，对于每一个单词来说，当叙述这个词时，这个人举起他的拳头，是一个在前还是另一个在前？换句话说，我们构建表明我们输入元素之间的空间关系的因子（即特征）。我们这样做是因为我们认为这些关系可能影响最终输出（当我们说我们的模型“假设特征之间的依赖关系”，这就是我们所要表达的）。

条件随机场输出函数是一个像softmax一样的函数。换句话说，如果我们为我们的会话分类任务构建一个softmax回归，其中：

1. 我们的输出是一系列标签。

2. 我们的特征是一堆（空间启发）的交互的特征，一个“sklearn.preprocessing.PolynomialFeatures”

我们基本上只是建立一个条件随机场。

当然，在输入条件下建立输出的完整分布的模型，其中我们的输出再次是一系列标签，导致组合爆炸非常快（例如，5个字的语句已经有 个可能的输出）。为此，我们使用一些动态编程技巧来确保我们在合理的时间内计算 .

### 3.10 隐式马尔科夫模型及其以外

最后，隐式马尔可夫模型是朴素贝叶斯条件随机场的 softmax 回归：每一对中的前者通过对一系列标签进行建模而建立在后者上。这个图1可以更深入地了解这些关系：

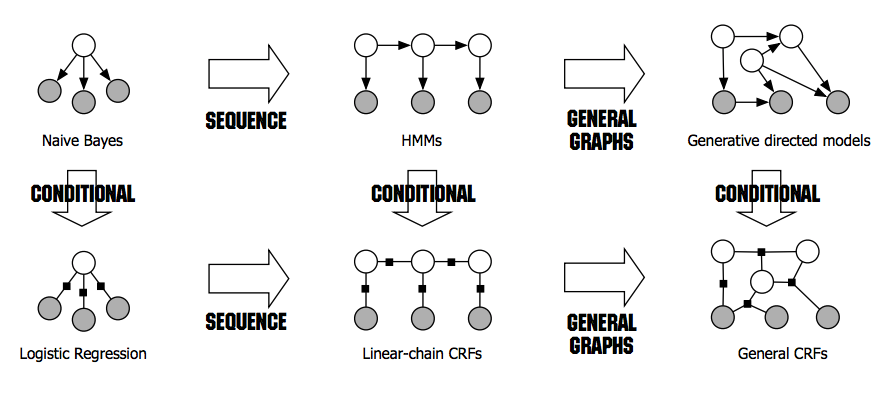


图1

### 3.11 从哪里来？

方程式（2）表明，softmax的分子，即输入元素的指数化线性组合，

## 第四章 总结

这次学习使典范机器学习模型，激活函数和条件概率的基本公理之间的关系变得更加清晰。 有关更多信息，请参考以下资源，特别是Daphne Koller的关于概率图形模型的材料。 非常感谢您阅读这篇文章。