前言

译自“An Introduction to Conditional Random Fields" --Charles Sutton, Andrew McCallum。我也在学习之中，必有错漏之处，希望能依靠大家的力量，共同进步。

目前数学公式显示总出问题，很多更新在gitbook上编译不通过。可以到github上下载pdf和all.md：
https://github.com/cottageLamp/CRFIntroduction_Chinese

总体感觉原文并不是很好理解，翻译之后也不好理解。好比一本介绍“降龙十八掌”的入门书，却时不时要求读者参考一下“九阴白骨爪”、“凌波微步”、“易筋经”……，岂不要命？

争取在翻译完成后，写一篇条理清晰的总结在后面，包括一些原文中没有的内容。

摘要

许多任务要对大量的变量进行预测。这些变量相互关联，且依赖于另外的已被观测量。结构化预测方法实质上是分类器与图模型的结合。图模型能够紧凑地对多变量数据建模，而分类器能够利用大规模的输入特征完成预测。本文描述了条件随机场，一种流行的、用于结构化预测的概率方法。CRFs 已在广泛的领域中获得大量应用，包括自然语言处理， 机器视觉以及生物信息学。 我们将描述CRFs的推断方法和训练方法，包括在实现大规模CRFs时的问题。不要求读者具有图模型的知识，希望能对广大的实践者们有用。

1介绍

对很多应用来说，至关重要的是预测互相关多变量的能力。这些应用广泛分布于图片分割及分类、围棋胜负概率的预测、在DNA序列中分离基因组，以及对自然文本进行语法分割。在这些应用中，我们想基于一组观测值$\boldsymbol{x}$，来预测一个随机输出向量$\boldsymbol{y} = y_0, y_1, \ldots, y_T$。一个相对简单的例子是对自然语言进行词性标注。其中，每个$\boldsymbol{y_s}$对应着s位置的单词的词性，而输入$\boldsymbol{x}$被分解成多个输入特征向量$\{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$。每个$\boldsymbol{x_s}$ 包含着s位置单词的多种信息，如它自身、它的前后缀、它在词典中的身份，以及来自语义数据库的信息（如WordNet）。（专业词汇有问题）

一种办法是为每个位置s训练位置无关的分类器$\boldsymbol{x} \to y_s$，尤其是当我们要最大化$\boldsymbol{y_s}$的正确率时。然而，困难在于输入变量$\boldsymbol{y_s}$之间存在复杂的依赖性。如在英语中，形容词不常接名词。又如在计算机视觉中，临近区域趋向属于相近的类。另一个难点在于，输出变量常常表现出一种复杂的结构，如语法树。那么，在树的顶端附近选择怎样的语法规则会对整个树有极大的影响。

图模型是一种表达互相关变量的自然的方法。图模型包括：贝叶斯网络，神经网络，因子图，马尔科夫随机场，伊辛模型（Ising model）等等。它们把一个复杂的概率分布分解成许多局部**因子(factor)**相乘，而这些因子各自对应着变量的一部分。我们有可能描述，按照一组条件独立关系对概率密度进行的分解，能在多大程度上满足着该分布。这种对应关系，使得建模更加容易，因为我们的经验知识常常提供了合理的条件独立假设，而这决定了我们如何进行分解。

关于图模型的工作，特别是自然语言处理相关的，大量地关注了**生成模型（generative models）**。生成模型显式地建立对所有输入和输出的联合分布$p(\boldsymbol{y}, \boldsymbol{x})$。尽管这有一些好处，但存在着重要的局限。不仅是因为输入$\boldsymbol{x}$的维度可能非常大，还因为输入$\boldsymbol{x}$内在的复杂的相关性。对它们进行建模是困难的。对输入的相关性进行建模，会导致难以驾驭的模型，而忽略它们却会降低系统的性能。

一种解决办法是**判别**方法，正如在逻辑回归分类器中的做法。这里，我们直接对$p(\boldsymbol{y}|\boldsymbol{x})$建模，因为这是完成分类所需的全部。这正是条件随机场（CRFs）所采用的方法。CRFs结合了判别分类器与图模型的优点。一方面能够紧凑地对多变量输出$\boldsymbol{y}$进行建模，一方面能够应付数量庞大的输入特征$\boldsymbol{x}$，以用于预测。条件模型的优势在于，它忽略了那些仅仅存在于$\boldsymbol{x}$内在变量之间的相关性。因此，条件模型要比联合模型具有简单得多的结构。生成模型和CRFs之间的差别，正如朴素贝叶斯分类器与逻辑回归分类器之间的差别。实质上，多元逻辑回归模型可以被看成一种最简单的CRF，因它只有一个输出。

本文描述了CRFs的 建模、推断（前向计算）和参数估计方法。读者不用具有图模型的知识，因而本文希望能对广大的实践者有用。我们从介绍CRFs建模的一些问题开始（第二章），包括线性CRFs通用结构的CRFs，以及包含潜藏变量的隐CRFs（hidden crfs）。我们将说明，为何CRFs既是著名的逻辑回归的扩展，有是判别式的隐马尔科夫模型。

在接下来的两章，我们描述了推断（第4章）和学习（第5章）。**推断**既指计算$p(\boldsymbol{y}|\boldsymbol{x})$的边缘分布，也指计算极大似然$\boldsymbol{y}^* = argmax_y p(\boldsymbol{y}|\boldsymbol{x})$。**学习**是指参数估计过程，就是找到$p(\boldsymbol{y}|\boldsymbol{x})$的参数，使其最大限度地符合一组训练样本$\{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\}_{i=1}^N$。推断和学习过程往往密切地组合在一起，因为学习过程需要推断做为子过程。

最后，我们讨论了CRFs与其他类模型的关系，包括结构化预测模型，神经网络和最大熵马尔科夫模型（第6章）。

1.1动手方面的细节

本文努力指出动手实现方面的细节，而这常常被学术文献所忽略。例如，我们讨论了特征工程（feature engineering 输入设计？）（第2.5节），在推断中避免数值溢出（第4.3节），CRF在一些基准问题上训练时的伸缩性。

因为这是我们关于实现细节的第一个章节，应该提一提可供使用的一些CRFs平台。在写作本文时，一些流行的平台包括：

CRF++	**http://crfpp.sourceforge.net/**
MALLET	http://mallet.cs.umass.edu/
GRMM	http://mallet.cs.umass.edu/grmm/
CRFSuite	http://www.chokkan.org/software/crfs
FACTORIE	http://www/factorie.cc

除此之外，用于马尔科夫逻辑网络的软件（如Alchemy：http://alchemy.cs.washington.edu/)也可用于构建CRF模型。 据我们所知，Alchemy, GRMM 和 FACTORIE 是仅有的、能够处理任意的图模型的工具。

2 建模

本章，我们从建模的角度来描述CRFs，阐述了CRF是如何把机构化的输出表示成高维输入向量的分布。可以把CRFs理解成，将逻辑回归分类器扩展到任意的图模型，也可以被理解成生成模型（如隐马尔科夫模型）的判别对应物。译注：判别和生成模型是两种在理论上等价（可互相推导得到对方），但建模思路相反的模型。

我们从对图模型的简单介绍（第2.1节），以及对NLP中的生成和判别模型的介绍（第2.2节）开始。然后，我们可以给出了CRF的正式定义，包括常用的线性链（linear chains）（第2.3节），以及通用图结构（第2.4节）。因为CRF的准确性严重依赖于所使用的特征，我们也描述了特征工程常用的一些技巧（第2.5节）。最后，我们提供两个CRF应用的例子（第2.6节），以及一个宽泛的、关于CRFs应用领域的报告。

2.1 图模型

图模型是表达和推断多元概率分布的强大框架。它已经在统计模型的许多领域被证明有用，包括编码理论（coding theory），计算机视觉，知识表达（knowledge representation），贝叶斯统计（Bayesian statistics），以及自然语言处理（广告语也太多了吧）。

直接描述包含许多变量的分布，其代价是昂贵的。假如我们用表（table）来描述n个二值变量的联合分布，需要$O(2^n)$个浮点数（建议读者理解一下：每个变量有2种可能的取值，而总共有n个变量，那么总共有2^n种可能的取值。它这里的意思是：给每种取值赋予一个浮点数，表示其概率）。从图模型的角度看，认为一个分布尽管建立在许多变量之上，但常常可以表示成一些局部方程（local functions）的乘积，而这些方程只依赖于少量的变量。这种分解实际上与变量间的某些条件独立性密切相关——两种信息被轻易地用途来概括。实质上，分解、条件独立与图的结构，这三者构成了图模型框架力量的来源：条件独立性视角主要用于设计模型，而分解视角主要用于设计推断算法。

在本节的余下部分，我们从以上两个视角来介绍图模型，关注那些建立在无向图（undirected graphs)之上的模型。关于更详细、更现代的图模型及其推断算法，可参考Koller 和 Friedman 【57】的教材。

2.1.1 无向图

我们考虑随机变量集合Y上的概率分布。我们通过整数$s \in 1, 2, ... |Y|$来对变量进行索引。每个变量$Y_s \in Y$的取值范围都是集合$\mathcal{Y}$。本文我们只考虑离散的$\mathcal{Y}$，尽管它也可以是连续的。Y的一次特定的取值记做$\boldsymbol{y}_s$。对于Y中的特定变量Y_s，$\boldsymbol{y}_s$包含了对它的赋值，记做y_s。记号$\mathbf{1}_{\{y=y'\}}$表示一个函数，在$y = y'$时取1，而在其他时候取0。我们还需要边缘分布的记号。对于某个固定的取值y_s，我们用求和符号$\sum_{\boldsymbol{y} \backslash y_s}$来表示：在$\boldsymbol{y}$的全部取值中，那些$Y_s = y_s$的取值的概率的和。

假定，我们相信一个概率分布p可以表示成一组因子，记做$\Psi(\boldsymbol{y}_a)$的连乘。其中，a是一个整数索引（下标），从1变化到A，而A就是因子的个数。每个因子$\Psi(\boldsymbol{y}_a)$只依赖于部分变量$Y_a \in Y$。$\Psi(\boldsymbol{y}_a)$是一个非负数，可以被看成$\boldsymbol{y}_a$的自洽性的度量。自洽性高的取值，其发生的概率就高。这种分解让我们更高效地表示分布p，因为集合Y_a要比完整的集合Y小得多。

一个无向图模型是这样一种概率分布，它根据一组给定的因子来分解模型。正式地，给定Y的子集$\{Y_a\}_{a=1}^{A}$的集合，一个无向图模型是所有可以写成下式的分布：

$$p(\boldsymbol{y}) = \frac{1}{Z} \prod_{a=1}^{A} \Psi(\boldsymbol{y}_a) \text{（2.1）}$$

其中，对于任意的因子$\mathcal{F} = \{\Psi(\boldsymbol{y}_a)\}$，及其对应的所有可能的$\boldsymbol{y}_a$，都有$\Psi(\boldsymbol{y}_a) \geq 0$。（这些因子又被称作**局部函数**或**自洽性函数**。）我们将用**随机场**来表示由某个无向图定义的特定分布。常数Z是一个归一化因子，保证分布p的和为1。它定义如下：

$$Z = \sum_{y} \prod_{a=1}^{A} \Psi(\boldsymbol{y}_a). (2.2)$$

Z的值，考虑成因子集合$\mathcal{F}$的函数的话，也被称作**配分函数（partition function）**。注意，式(2.2)中的求和，需要在爆炸式的$\boldsymbol{y}$的所有可能取值上进行。因此，计算Z通常是不可行的，但是有很多关于估计它的研究（见第4章）。

术语“图模型”的来由，在于式（2.1）所表示的因子分解，可以建紧凑地表示成一张图。**因子图【58】**提供了一个特别自然的构图方法。一个因子图是一个两两连接图$G = (V, F, E)$。其中，节点的集合$V = \{1, 2, ..., |Y|\}$索引了模型中的全部随机变量，另一组节点的集合$F = \{1, 2, ..., A\}$索引了所有的因子。对图的理解是：如果一个变量节点s连接到一个因子节点a，那么在模型中，变量Y_s就是因子Ψ_a的一个参数。所以，因子图直接描述了，一个分布是如何被分解成多一个局部函数的乘积的。

我们正式地定义——一个因子图是否“描述”了一个分布？记$N(a)$包含了所有连接到因子节点a上的变量节点，那么：

定义2.1 仅当存在一组局部方程$\Psi(\boldsymbol{y}_a)$，使得p可以写成：

$p(\boldsymbol{y}) = Z^{-1}\prod_{a\in F}\Psi(\boldsymbol{y}_{N(a)})$（2.3）

时，一个分布$p(y)$根据因子图G分解了。

一组子集描述了无向模型，而一个因子图同样如此。在式（2.1）中，取子集为节点的邻居$\{Y_N(a)|\forall a\in F\}$。根据式（2.1）定义的无向图模型，对应着所有根据G进行分解所得的分布。

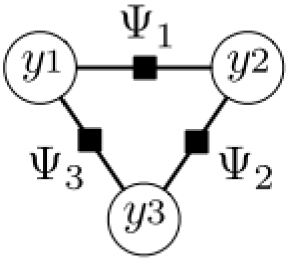

图2.1 带3个变量的因子图

图2.1展示了一个带有3个随机变量的因子图，图中，圆圈是变量节点，而灰色方块是因子节点。我们根据节点的索引进行了标注。这个因子图能够描述所有的带3个变量的分布，前提是对于任意的$\boldsymbol{y}=(y_1,y_2,y_3)$，该分布能够写成$p(y_1,y_2,y_3)=\Psi_1(y_1,y_2)\Psi_2(y_2,y_3)\Psi_3(y_1,y_3)$的形式。

图模型的因子分解与变量间（在其取值范围里）的条件独立性密切相关。这种联系可通过另一种无向图来理解——马尔科夫网。它直接描述了多元分布的条件独立关系。马尔科夫网只是随机变量的图，不包括因子。现记G为整数序列$V=\{1,2,\ldots,|Y|\}$上的无向图，而V仍是随机变量的索引。对于某一个索引s，记$N(S)$为它的邻居。那么我们称p是关于G的马尔科夫网，仅当它满足局部的马尔科夫特性：对于任意的两个变量$Y_s,Y_t\in Y$，Y_s关于它的邻居独立于Y_t。

把所有连接到同一个因子的变量都两两连接起来，可将如式(2.1)的分布，变成其对应的马尔科夫网。这很显然，因为由式（2.1）而来的条件分布$p(y_s|\boldsymbol{y}_{N(S)})$仅仅是那些马尔科夫毯中的变量的函数。

从因子分解的角度看，马尔科夫网存在着不好的歧义性。考虑图2.2（左）的3变量马尔科夫网。任何按照$p(y_1,y_2,y_3)\propto f(y_1,y_2,y_3)$分解的分布，都可能与它对应。然而，我们希望使用更严格的参数化——$p(y_1,y_2,y_3)=f(y_1,y_2)g(y_2,y_3)h(y_1,y_3)$。后面这组模型簇是前面的严格子集，且需要更少的数据来获得准确的分布估计译注：参数估计？。然而，马尔科夫网不能区分这两种参数化。相反，因子图无歧义地描述了模型的因子分解。

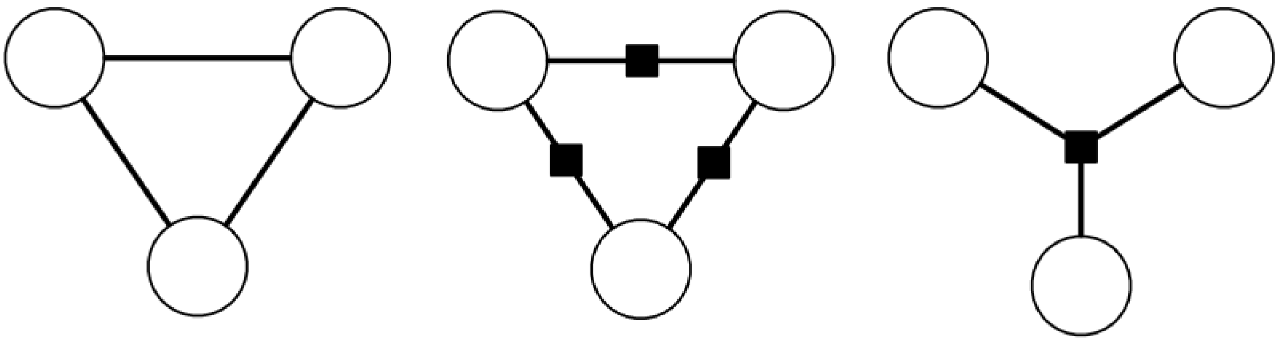

图2.2带有歧义的马尔科夫网（左）。右边的两种分解都有可能与左图对应。

2.1.2有向图

无向模型中的局部函数无需带有方向性的概率表达，有向图模型却把分布分解成局部的条件概率分布。记G为有向无环图，$\pi(s)$为Y_s的所有父节点的序号集合。一个有向图模型是一簇按照如下分解的分布：

$$p(\boldsymbol{y}) = \prod_{s=1}^{S} p(y_s|\boldsymbol{y}_{\pi(s)}). (2.4)$$

我们称$p(y_s|\boldsymbol{y}_{\pi(s)})$为**局部条件分布（localconditionaldistributions）**。注意，对于没有父节点的变量，$\pi(s)$可以是空的。这时，$p(y_s|\boldsymbol{y}_{\pi(s)})$可被理解为$p(y_s)$。可以推断$p$是合理归一化的。可以这样来理解有向模型——其每个因子都在局部完成了特殊的归一化，使得（1）因子相当于局部变量上的条件分布，且（2）归一化常数$Z=1$。有向模型常常用于生成模型，我们将在第2.2.3节讲述这一点。有向模型的一个例子是贝叶斯模型（2.7），被描述在图2.3（左）了。在这些图中，灰节点表示了某些数据集上观测的变量。贯穿本文，我们都将采用这一习惯。

2.2生成与判别模型

本节我们探讨几个已被用于自然语言处理的简单图模型。虽然它们已被熟知，但它们一方面可以澄清前文提到的诸多概念，另一方面也可以说明某些今后讨论CRFs时会遇到的议题。我们尤其关注隐马尔科夫模型（HMM），因为它与线性链条件随机场密切相关。

本节的主要目的是对比生成与判别模型。将会提到的模型，包括两个生成模型（朴树贝叶斯和HMM），一个判别模型（逻辑回归模型）。**生成模型**描述了，一个输出向量$\boldsymbol{y}$以怎样的概率“生成”输入特征$\boldsymbol{x}$。**判别模型**从相反的方向工作，直接描述了如何利用输入特征$\boldsymbol{x}$来给输出$\boldsymbol{y}$赋值。一般来说，这两者可根据贝叶斯法则互相转化。但在实践中却相去甚远，各自隐藏着一些优点（将在2.2.3节讲述）。

2.2.1 分类

我们首先讨论**分类**问题——根据给定的一个向量$\boldsymbol{x}=(x_1,x_2,\ldots,x_K)$，来预测单一的$y$变量的离散值（类别标签）。一个简单的方法是，假定当类别标签已知时，所有的特征是独立的。结果是所谓的朴素贝叶斯分类器。它基于如下的联合概率模型：

$$p(y,\boldsymbol{x}) = p(y)\prod_{k=1}^{K} p(x_k|y). (2.7)$$

这个模型可以描述为图2.3（左）的有向模型。为每个特征x_k定义因子$\Psi(y)=p(y)$，以及因子$\Psi_k(y,x_k)=p(x_k|y)$,我们也可以写成因子图。这样的因子图如图2.3（右）所示。

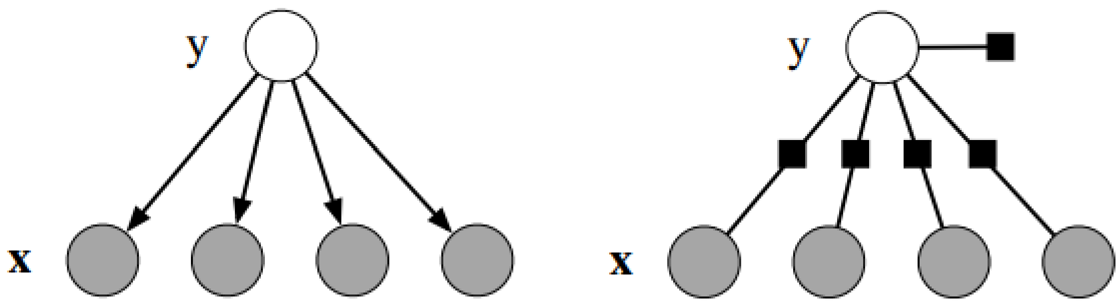

图2.3 朴素贝叶斯分类器，被当成有向模型（左），或因子图（右）

逻辑回归（有时在NLP圈子里叫做**最大熵分类器**）是另一个知名的，且很自然地表达为图模型的分类器。该分类器源于将每个类的逻辑概率，$\log p(y|\boldsymbol{x})$，假设为$\boldsymbol{x}$的线性函数，以及一个归一化常数。这导致了如下的条件概率：

$$p(y|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} exp\{\theta_y + \sum_{j=1}^{K}\theta_{y,j}x_j\}, (2.8)$$

其中$Z(\boldsymbol{x}) = \sum_y exp\{\theta_y + \sum_{j=1}^K \theta_{y,j} x_j\}$，是归一化常数。而$\theta_y$是偏置量，相当于朴素贝叶斯里面的log$p(y)$。与其像式（2.8）那样为每一个类制定一个权重向量，我们不如采用被所有类共享的一组权重的记号。这一技巧通过定义一组**特征函数(feature functions)**来实现，而这些特征只对某一类时非零。为了达到这个目的，特征权重的特征函数被定义为$f_{y',j}(y, \boldsymbol{x}) = \mathbf{1}_{\{y'=y\}} x_j$，而把偏置权重的特征函数定义为$f_{y'}(y, \boldsymbol{x}) = \mathbf{1}_{\{y'=y\}}$。现在我们可以用$f_k$来遍历每个特征函数$f_{y',j}$，用$\theta_k$来索引对应的权重$\theta_{y',j}$。利用这一符号技巧，逻辑回归模型变成了：

$$p(y|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} exp\{\sum_{k=1}^K \theta_k f_k(y, \boldsymbol{x})\}. (2.9)$$

我们之所以引入这样的记号，是因为它简化了下文介绍CRFs时的记号。译注：（2.8）中的θ_y好像丢失了？

2.2.2 序列模型

分类器只对单一变量做预测，但图模型的真正用处在于对大量互相关变量的建模能力。本节，我们讨论了可能是最简单的相关性——图模型中的输出变量被排列成一个序列。为了展示该模型的好处，我们讨论一个自然语言处理中的应用——**命名实体识别（named-entity recognition,NER）**。NER是在文本中识别并分类命名实体，包括地点（如China），人（如George Bush）和组织（如United Nations)。给定一个句子，命名实体识别任务是把其中的单词切分成几段，每一段对应一个实体，然后对该实体进行分类（类别包括人，组织，地点等等）。该问题的挑战性在于，很多实体的字符串很少见，哪怕在一个很大的训练集上。于是，我们只能根据上下文来识别它们。

一种办法是独立地对每个单词进行分类，看它是一个人、地点、组织或者其他（既不是一个实体）。这种办法的缺点在于：给定输入之后，它假定所有的命名实体标签是独立的。实际上，临近单词的标签是相关的。例如，New York是一个地点，Now York Times却是一个组织。一种缓解这种无关性假设的方法，是把输出变量安排到一个线性链中。这是隐马尔科夫模型（HMM）【111】的方法。一个HMM通过假定一个潜在的**状态**序列$Y = \{y_t\}_{t=1}^T$，来对一序列的观测$X = \{x_t\}_{t=1}^T$ 建模。记S为可能状态的有限集，O为可能观测的有限集，即是说，对于任何的t，$x_t \in O, y_t \in S$译注：S包含了所有可能的输出值，O包含了所有可能的输入值。在命名实体例子中，t位置的单词就是观测x_t，而y_t 是该位置的标签。

为了可行地对联合分布$p(\boldsymbol{y}, \boldsymbol{x})$ 建模，一个HMM做了两个无关性假设。第一，它假设每个状态只依赖于它的前一个状态，即给定y_{t-1} 之后，y_t 于$y_1, y_2, \ldots, y_{t-1}$ 都无关了。第二，它假定每个观测变量x_t 只与对应的状态y_t 有关。基于这些假设，我们可用三个概率分布来指明一个HMM。第一个，初始状态的概率布$p(y1)$；第二个，转移概率$p(y_t|y_{t-1})$;最后，观测概率$p(x_t|y_t)$。总之，状态序列$\boldsymbol{y}$于观测序列$\boldsymbol{x}$的联合分布被分解为：

$$p(\boldsymbol{y}, \boldsymbol{x}) = \prod_{t=1}^T p * y_t|y_{t-1})p(x_t|y_t). (2.10)$$

为了简化上式的符号，我们创造了“虚拟”初始状态y_0，它总是0，并是所有状态序列的起点。这让我们把创始状态概率$p(y_1)$写成$p(y_1|y_0)$。

HMMs已在自然语言处理中用于很多序列标注任务，如part-of-speech tagging, 命名实体识别和信息提取。

2.2.3比较

生成模型和判别模型都描述了$(\boldsymbol{y}, \boldsymbol{x})$的分布，却是从不同的方向。生成模型，如朴素贝叶斯分类器和HMM，是一簇按照$p(\boldsymbol{y}, \boldsymbol{x}) = p(\boldsymbol{y})p(\boldsymbol{x}|\boldsymbol{y})$进行分解的联合分布。也就是说，它描述了如何根据标签采样或”生成“特征。判别模型，如逻辑回归模型，是一簇条件分布$p(\boldsymbol{y}|\boldsymbol{x})$。也就是说，直接对分类规则建模。原理上，利用输入的边缘分布$p(\boldsymbol{x})$，一个判别模型可以被转化成联合分布$p(\boldsymbol{y}, \boldsymbol{x})$，然而很少需要这么做。

判别和生成模型在概念上的主要区别，就是条件分布$p(\boldsymbol{y}|\boldsymbol{x})$没有包含$p(\boldsymbol{x})$的模型，而它对分类并没有用。对$p(\boldsymbol{x})$建模的困难性在于，它包含了很多高度相关的特征，而这是很难建模的。如在命名实体识别中，朴素的HMM只依赖于单一的特征——单词本身。然而许多单词，特别是专有名词，却从未在训练集中出现过，因而以单词本身作为特征是缺乏足够的信息的。为了对全新单词进行标注，我们想要利用其它的特征，如它的大小写、它的临近单词、它的前后缀、它在预先确定的一组人或地方中的身份（its membership in predetermined lists of people and locations???)，等等。

判别模型的主要优势在于它适合包含丰富的、重叠的特征。为了理解这一点，考虑一簇朴素贝叶斯分布(2.7)。这簇联合分布的条件部分均采用了"逻辑回归的形式"（2.9）。然而还有很多其他的联合模型，有些带有$\boldsymbol{x}$之间的复杂的依赖，而条件分布也采用了（2.9）的形式。为了直接对条件分布建模，我们仍然可以认为$p(\boldsymbol{x})$是不可知的。判别模型，如CRF，仅对$\boldsymbol{y}$的条件独立性做假设，以及$\boldsymbol{y}$如何依赖于$\boldsymbol{x}$，但是不对$\boldsymbol{x}$之间的条件独立性做假设。 这一点也可以通过图形的方式来理解。假定我们有关于联合分布$p(\boldsymbol{y},\boldsymbol{x})$的因子图，现在要构建条件分布$p(\boldsymbol{y}|\boldsymbol{x})$的因子图，那么，所有只与$\boldsymbol{x}$有关的因子都可以消失了。它们与条件部分无关，因为它们关于$\boldsymbol{y}$是常数。

为了在生成模型中包含互相关的特征，我们有两个选择。一是增强模型以表达输入间的相关性，如在每个$\boldsymbol{x}_t$之间增加连接。然而很难可操作地这样做。例如，很难想象如何对单词的大小写以及前后缀之间的相关性建模。亦或者，我们也不想去做这个件事，因为我们总是看得到输入的句子。

第二个办法是只做一些简单的相关性假设，如朴素贝叶斯假设。例如，带有朴素贝叶斯假设的HMM采用了$p(\boldsymbol{x},\boldsymbol{y}) = \prod_{t=1}^{T} p(y_t|y_{t-1}) \prod_{k=1}^{K} p(x_{tk}|y_t)$的形式。这一思路有时很凑效，但也可能很有问题，因为这一独立性假设会影响性能。例如，虽然朴素贝叶斯分类器在文档分类方面表现优秀，它在许多应用中的平均表现要比逻辑回归差【19】。

而且，朴素贝叶斯可以产生差的概率估计。作为说明的例子，想象朴素贝叶斯在一个二分类问题上训练。现在，我们把输入特征向量$\boldsymbol{x} = (x_1, x_2, \ldots, x_K)$重复一下，变换成$\boldsymbol{x}' = (x_1, x_1, x_2, x_2, \ldots, x_K, x_k)$，然后运行朴素贝叶斯分类器。虽然没有任何新的信息被加入到数据中，这一变换却增加了概率估计的信心。就是说，朴素贝叶斯对$p(y|\boldsymbol{x}')$的估计，相比于$p(y|\boldsymbol{x})$，更倾向远离0.5。

当我们扩展到序列模型的时候，想朴素贝叶斯那样的假设尤其有问题，因为推断过程需要综合模型不同部分的证据。如果序列的每个位置的标签，其概率估计都偏大，那么很难合理地把它们综合起来。

朴素贝叶斯和逻辑回归之间的差别，正是前者是生成的，而后者是判别的。在输入为离散时，这两个分类器在其他方面完全一致。朴素贝叶斯和逻辑回归考虑了相同的假设空间，因为在相同的决策范围里，任何逻辑回归分类器都可以转变成朴素贝叶斯分类器，反之亦然。再者，朴素贝叶斯模型(2.7)与逻辑回归模型（2.9）定义了相同的分布簇。我们可以生成式地表示（2.7）如下：

$$p(y,\boldsymbol{x}) = \frac{exp\{\sum_k \theta_k f_k(y,\boldsymbol{x})\}}{\sum_{\hat{y},\hat{\boldsymbol{x}}} \theta_k f_k(\hat{y},\hat{\boldsymbol{x}})}. \quad (2.11)$$

这意味着，如果朴素贝叶斯(2.7)按照极大条件似然来训练，我们会获得与逻辑回归一样的分类器。相反，如果按照生成方法来表示逻辑回归，如（2.11），并按照最大化联合似然$p(y,\boldsymbol{x})$来训练，我们会得到与朴素贝叶斯同样的分类器。按照Ng和Jordan【98】的说法，朴素贝叶斯和逻辑回归构成了**生成-判别对（generative-discriminative pair)**。关于最新的生成与判别模型的理论视角，请参考Liang和Jordan【72】。

原理上，我们可能不清楚这两种方案如此不同的原因，毕竟它们之间可通过贝叶斯法则互相转化。如在朴素贝叶斯模型中，是很容易把联合分布$p(\boldsymbol{y})p(\boldsymbol{x}|\boldsymbol{y})$转化成条件分布$p(\boldsymbol{y}|\boldsymbol{x})$的。 实际上，该条件分布与逻辑回归模型（2.9）的形式是一样的。另外如果我们想获得关于数据的"真实"生成模型，即真正把数据产生出来的分布$p^*(\boldsymbol{y},\boldsymbol{x}) = p^*(\boldsymbol{y})p^*(\boldsymbol{x}|\boldsymbol{y})$，那么我们只需简单地计算真实的$p^*(\boldsymbol{y}|\boldsymbol{x})$，而这正是判别方法的目标。然而正是因为我们无法准确地获得真实的分布，造成这两种方案在实践中是不同的。先估计$p(\boldsymbol{y})p(\boldsymbol{x}|\boldsymbol{y})$，然后计算$p(\boldsymbol{y}|\boldsymbol{x})$（生成方案），会产生与直接估计$p(\boldsymbol{y}|\boldsymbol{x})$不同的结果。也就是说，生成与判别模型的目标都是估计$p(\boldsymbol{y}|\boldsymbol{x})$，却是通过不同的路径达到的。

我们关于生成与判别之间差异的深入观点，来自Minka【93】。假如我们拥有一个生成模型p_g，其参数为θ。根据定义，其形式为：

$$p_g(\boldsymbol{y},\boldsymbol{x};\theta)=p_g(\boldsymbol{y};\theta)p_g(\boldsymbol{x}|\boldsymbol{y};\theta).(2.12)$$

但是我们也可以按照概率的链式法则重写p_g如下：

$$p_g(\boldsymbol{y},\boldsymbol{x};\theta)=p_g(\boldsymbol{x};\theta)p_g(\boldsymbol{y}|\boldsymbol{x};\theta),(2.13)$$

其中，$p_g(\boldsymbol{x};\theta)$和$p_g(\boldsymbol{y}|\boldsymbol{x};\theta)$是通过推断来计算的，即$p_g(\boldsymbol{x};\theta)=\sum_{\boldsymbol{y}}p_g(\boldsymbol{y},\boldsymbol{x};\theta)$以及$p_g(\boldsymbol{y}|\boldsymbol{x};\theta)=p_g(\boldsymbol{y},\boldsymbol{x};\theta)/p_g(\boldsymbol{x};\theta)$。

现在要在同样的联合分布簇上，把这个生成模型与判别模型做比较。为了这么做，我们定义一个关于输入的先验概率$p(\boldsymbol{x})$，使得$p(\boldsymbol{x})$可以从p_g的某个参数配置中产生。就是说，$p(\boldsymbol{x})=p_c(\boldsymbol{x};\theta')=\sum_{\boldsymbol{y}}p_g(\boldsymbol{y},\boldsymbol{x};\theta')$译注：原文是$p(\boldsymbol{x})=p_c(\boldsymbol{x};\theta')=\sum_{\boldsymbol{y}}p_g(\boldsymbol{y},\boldsymbol{x}|\theta')$，其中$\theta'$往往与（2.13）中的$\theta$不同。把这与同样从$p_g$中产生的条件分布$p_c(\boldsymbol{y}|\boldsymbol{x};\theta)$组合，即$p_c(\boldsymbol{y}|\boldsymbol{x};\theta)=p_g(\boldsymbol{y},\boldsymbol{x};\theta)/p_g(\boldsymbol{x};\theta)$。那么结果分布是：

$$p_c(\boldsymbol{y},\boldsymbol{x})=p_c(\boldsymbol{x};\theta')p_c(\boldsymbol{y}|\boldsymbol{x};\theta).(2.14)$$

通过比较(2.13)和（2.14），可以看到条件方案具有更大的灵活性来拟合数据，因为它不要求$\theta'=\theta$。直观地，因为（2.13）中的参数θ被同时用于输入的分布和条件部分。那么一组参数需要在两方面都表现良好。潜在地，需要损失我们所关心的$p(\boldsymbol{y}|\boldsymbol{x})$的准确性，来弥补我们不怎么关心的$p(\boldsymbol{x})$的准确性。另一方面，引入了更多的自由度，增加了过拟合的风险，降低了泛化到新数据的能力。

尽管到目前为止我们一直在批判生成模型，它们也有自己的优势。第一，生成模型可以更自然地处理隐藏变量，半标注数据以及未标注数据。在更极端的例子中，当整个数据都未被标注时，生成模型可以按照非监督模式使用。相反，非监督学习在判别模型中不够自然，且扔是一个活跃的研究领域。

第二，在某些例子中生成模型表现得比判别模型好，直观上是因为输入模型$p(\boldsymbol{x})$对条件分布的影响是光滑的（smoothing）。Ng和Jordan【98】争辩道，这一作用在小数据机上尤其显著。对于任何特定的数据集，我们不可能知道谁更有优势。总之，要么问题本身需要一个自然的生成模型，要么需要同时预测输入与输出译注：一般应用假定输入为已知，而只需预测输出，都会使生成模型更被青睐。

因为生成模型的形式为$p(\boldsymbol{y},\boldsymbol{x})=p(\boldsymbol{y})p(\boldsymbol{x}|\boldsymbol{y})$，使得通过有向图来表示它更自然。其中在拓扑意义上，输出$\boldsymbol{y}$要在输入之前。相似地，我们将会看到，用无向图来表示判别模型更自然。然而，并非总是如此。无向的生成模型，如马尔科夫随机场（2.32），以及有向的判别模型，如MEMM（6.2），有时也会被采用。有时用有向图来表示判别模型也会有用，其中$\boldsymbol{x}$在$\boldsymbol{y}$之前。

朴素贝叶斯与逻辑回归之间的关系，正如HMMs和线性链CRFs。正如朴素贝叶斯与逻辑回归是生成-判别对，也存在着HMMs的判别对应物。这一对应物是一种特殊的CRF。我们将在接下来一章中介绍。朴素贝叶斯、逻辑回归、生成模型和CRFs之间的类比，如图2.4所示。

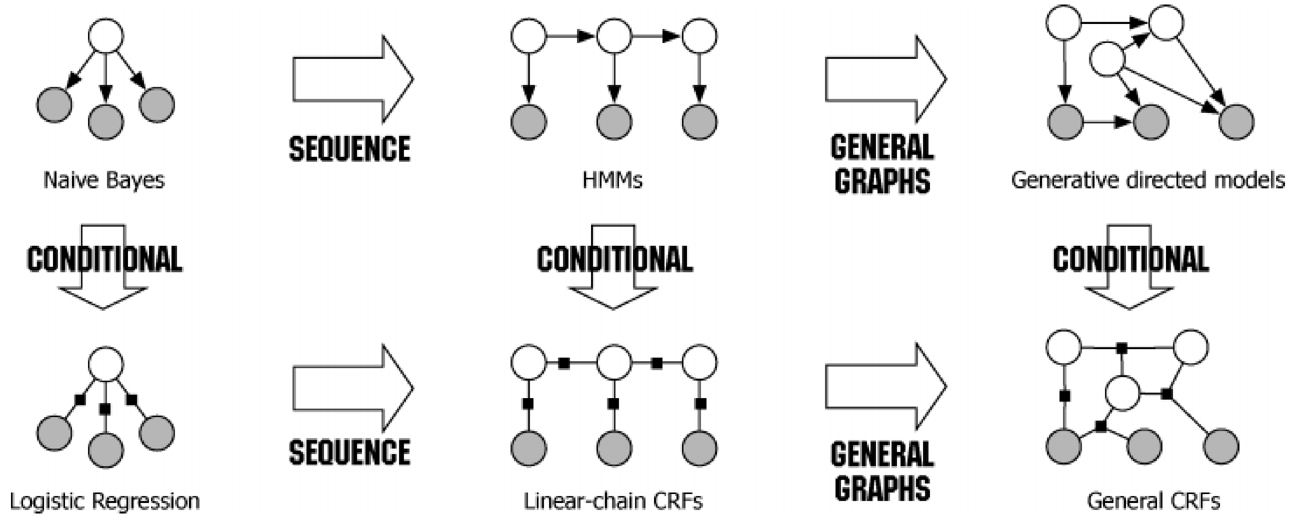

图2.4 朴素贝叶斯、逻辑回归、HMMS、线性链CRFs、生成模型和广义CRFs之间的关系图

2.3 线性链CRFs

为了引出线性链CRFs，我们考虑从HMM的联合分布$p(\boldsymbol{y},\boldsymbol{x})$引出的条件分布$p(\boldsymbol{y}|\boldsymbol{x})$。关键点在于，这一条件分布是一种具有特殊的特征方程的CRF。

首先，我们来重写HMM的联合分布(2.10)，使其更利于扩展，即：

$$p(\boldsymbol{y},\boldsymbol{x})=\frac{1}{Z}\prod_{t=1}^{T}exp\left\{\sum_{i,j\in S}\theta_{ij}\mathbf{1}_{\{y_t=i\}}\mathbf{1}_{\{y_{t-1}=j\}}+\sum_{i\in S}\sum_{o\in O}\mu_{oi}\mathbf{1}_{\{y_t=i\}}\mathbf{1}_{\{x_t=o\}}\right\},(2.15)$$

其中，$\theta=\{\theta_{ij},\mu_{oi}\}$是分布的实值参数，Z是归一化常数，能使分布的和为1。如果我们不在（2.15）中添加Z，那么参数θ有可能带来不合理的关于$(\boldsymbol{y},\boldsymbol{x})$的分布，如当所有参数都是1时。

现在有意思的是，（2.15）（几乎）确切地描述了（2.10）一类的HMMs。每个同类的HMM都可通过如下设置，写成（2.15）的形式：

$$\begin{gathered}\theta_{ij}=\log p(y'=i|y=j)\\ \mu_{oi}=\log p(x=o|y=i)\\ Z=1\end{gathered}$$

反过来也是正确的，即是说，每个按照（2.15）分解的分布都是HMM。（利用4.1节介绍的前向-反向算法，可构造对应的HMM，从而证明这一点）。因而尽管在参数中增加了灵活性，我们却没有扩大分布簇。

通过使用**特征函数feature functions**，我们可以把（2.15）弄得更紧凑，正如我们在（2.9）的逻辑回归那里一样。每个特征函数都具有形式$f_k(y_t,y_{t-1},x_t)$。对于（2.15），我们需要给每个转移(i,j)一个特征$f_{ij}(y,y',x)=\mathbf{1}_{\{y=i\}}\mathbf{1}_{\{y'=j\}}$，以及给每个“状态-特征对”$(i,o)$一个特征$f_{io}(y,y',x)=\mathbf{1}_{\{y=i\}}\mathbf{1}_{\{x=o\}}$。我们泛泛地用$f_k$来引用一个特征，其中$f_k$涵盖了全部都的$f_{ij}$和全部的$f_{io}$。于是，我们可以重写HMM如下：

$$p(\boldsymbol{y},\boldsymbol{x})=\frac{1}{Z}\prod_{t=1}^{T}\exp\left\{\sum_{k=1}^{K}\theta_k f_k(y_t,y_{t-1},x_t)\right\}.(2.16)$$

再一次，方程（2.16）定义了与（2.15）完全一样的分布簇，从而也与最初的HMM方程（2.10）一样。

最后一步，是把来自HMM（2.16）的条件分布$p(\boldsymbol{y}|\boldsymbol{x})$写出来，即：

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{p(\boldsymbol{y},\boldsymbol{x})}{\sum_{\boldsymbol{y}'} p(\boldsymbol{y}',\boldsymbol{x})} = \frac{\prod_{t=1}^{T} exp\left\{\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t)\right\}}{\sum_{\boldsymbol{y}'} \prod_{t=1}^{T} exp\left\{\sum_{k=1}^{K} \theta_k f_k(y'_t, y'_{t-1}, x_t)\right\}}. (2.17)$$

（2.17）所描述的条件分布，是线性链CRF的一种特例，即那种只包含当前单词作为特征的。然而，很多线性链CRF使用更为丰富的特征，如前后缀等等。幸运的是，将我们现有的记号扩展并非难事。我们只需简单地允许特征函数包含更多的输入。这导致了我们关于线性链CRFs的一般定义

定义2.2 记Y, X是随机向量，$\theta = \{\theta_k\} \in \mathcal{R}^K$是一个参数向量，$\mathcal{F} = \{f_k(y, y', \boldsymbol{x}_t)\}_{k=1}^K$为一组实值特征函数。那么**线性链条件随机场**是如下形式的分布$p(\boldsymbol{y}|\boldsymbol{x})$：

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z} \prod_{t=1}^{T} exp\left\{\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, \boldsymbol{x}_t)\right\}, (2.18)$$

其中，$Z(\boldsymbol{x})$是依赖于输入的归一化函数：

$$Z(\boldsymbol{x}) = \sum_{y} \prod_{t=1}^{T} exp\left\{\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, \boldsymbol{x}_t)\right\}. (2.19)$$

译注：线性链条件随机场，好像是一类随机场，实际是一个随机场——结构是定死的。我觉得这是条件随机场最非常核心的问题，本文却并没有阐明。当然，它对输入的引用还是很灵活的。

注意，线性链CRF可以用$\boldsymbol{x}$和$\boldsymbol{y}$上的因子图来描述，即

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{t=1}^{T} \Psi_t(y_t, y_{t-1}, \boldsymbol{x}_t) (2.20)$$

其中，局部函数$\boldsymbol{\Psi_t}$具有一种特殊的 log-linear形式：

$$\Psi_t(y_t, y_{t-1}, \boldsymbol{x}_t) = exp\left\{\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, \boldsymbol{x}_t)\right\}. （2.21）$$

当我们在下一节进入一般意义CRF的时候，这会很有用。

一般来说，我们将从数据中学得参数θ。这将在第5节讲述。

之前我们已看到，如果一个联合分布$p(\boldsymbol{y},\boldsymbol{x})$像HMM一样分解了，那么对应的条件分布$p(\boldsymbol{y}|\boldsymbol{x})$是一个线性链CRF。 这一很像HMM的CRF如图2.5所示。然而，其他类型的线性链CRFs也是有用的。例如，在一个HMM中，状态i到j的转移概率与输入无关，都是$\log p(y_t = j|y_{t-1} = i)$。在CRF中，我们可以让转移概率$(i, j)$依赖于当前的观测向量，这只需添加特征$\mathbf{1}_{\{y_t=j\}}\mathbf{1}_{\{y_{t-1}=j\}}\mathbf{1}_{\{x_t=o\}}$。 具有这一转移特征的CRF常常被用于文本处理，如图2.6所示。

实际上，因为CRFs不在乎输入变量$\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T$之间的关系，我们可以让因子$\boldsymbol{\Psi_t}$依赖于所有的输入$\boldsymbol{x}$。这不会大破线性图的结构——允许我们把$\boldsymbol{x}$当成单一的整体变量。结果，特征函数可以写成$f_k(y_t, y_{t-1}, \boldsymbol{x})$，从而可以把全部的输入变量$\boldsymbol{x}$一块考虑。这一事实对CRFs都适用，而不只是对线性链。具有这一结构的线性链如图2.7所示。途中，我们把$\boldsymbol{x} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T)$画成一个巨大的观测节点，冰杯所有的因子依赖，而不是把$\boldsymbol{x}_1, \cdot, \boldsymbol{x}_T$画成独立的节点。

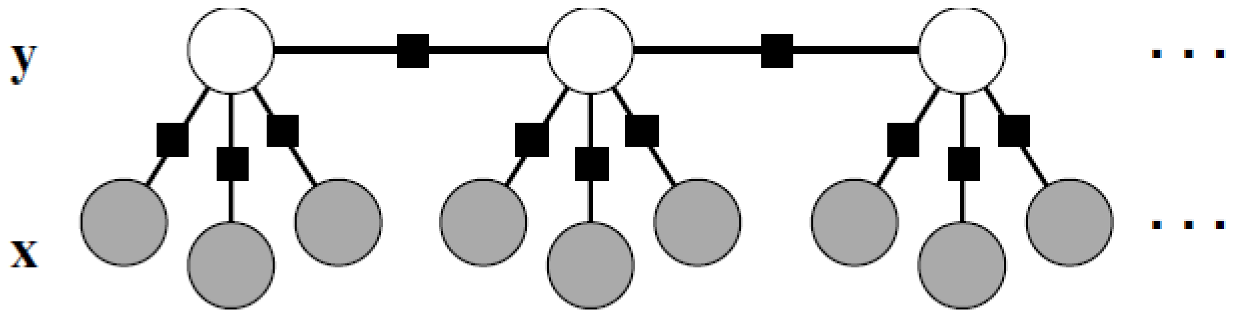

图2.5 来自式（2.17）的类HMM的线性链CRF

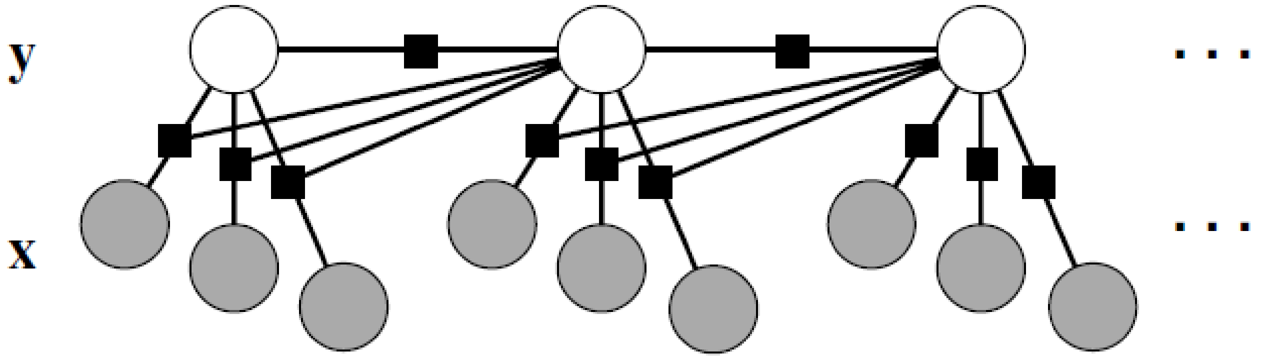

图2.6 转移因子依赖于当前输入的线性链CRF

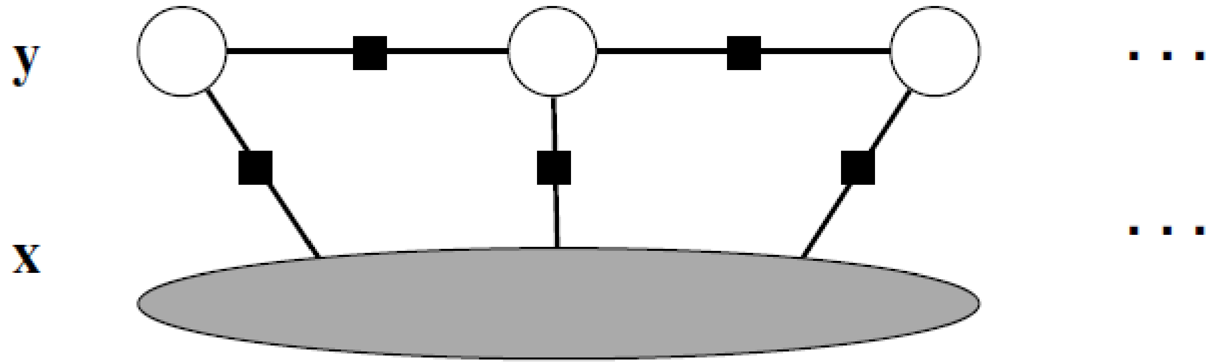

图2.7 转移因子依赖于全部输入的线性链CRF

需支出，在我们关于线性链CRF的定义中，特征函数可以从任意时刻依赖于输入，把f_k关于输入的参数写成了$\boldsymbol{x}_t$。$\boldsymbol{x}_t$应当被理解成——计算t时刻特征所需的全部输入译注：而不是t时刻的输入。 例如，如果CRF需要下一时刻的单词$\boldsymbol{x}_{t+1}$，那么$\boldsymbol{x}_t$应当包含了$\boldsymbol{x}_{t+1}$。

最后，归一化常数$Z(\boldsymbol{x})$需要在全部 可能的输出序列上求和，包含有爆炸式的大量的项。然而，它可以被前向-反向算法有效地解，正如我们在第4.1节所揭示的。

2.4 通用CRFs

现在，我们将刚刚探讨的线性链扩展到通用图，以与Lafferty在【63】中对CFR的定义相匹配。概念上，这一扩展是显而易见的。我们只需简单地把线性链因子图变成通用因子图。

定义2.3 记G是在X,Y上的因子图。如果对于X中任意的值$\boldsymbol{x}$，分布$p(\boldsymbol{y}|\boldsymbol{x})$是根据G来分解的，那么$(X,Y)$是一个**条件随机场conditional random field**。

那么，每个条件分布$p(\boldsymbol{y}|\boldsymbol{x})$都是某些因子图的CRF，包括是平凡的。如果$F \in \{\Psi_a\}$是G中的因子的集合，那么一个CRF的条件分布为：

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{X})} \prod_{a=1}^{A} \Psi_a(\boldsymbol{y}_a, \boldsymbol{x}_a). (2.22)$$

本定义相比一般无向图的定义（2.1），差别在于归一化常数$Z(\boldsymbol{x})$现在变成了关于输入$\boldsymbol{x}$的函数。因为条件性趋向于简化图模型，$Z(\boldsymbol{x})$ 有可能被计算，而Z却不是。

正如我们在HMMs和线性链CRFs中的做法，让Ψ_a是一组特征的线性函数是有用的，即：

$$\Psi_a(\boldsymbol{y}_a, \boldsymbol{x}_a) = exp\left\{\sum_{k=1}^{K(A)} \theta_{ak} f_{ak}(\boldsymbol{y}_a, \boldsymbol{x}_a)\right\}, (2.23)$$

其中特征函数f_{ak}和权重θ_{ak}都使用了因子的下标a，这是为了强调每个因子都有自己的权重集。一般来说，每个因子也可以拥有自己的特征函数。注意，如果$\boldsymbol{x}$和$\boldsymbol{y}$是离散的，那么（2.23）中的log-线性假设并没有带来额外的局限，因为我们可以给$(\boldsymbol{y}_a, \boldsymbol{x}_a)$的每一个值安排一个指示函数$f_{ak}$，类似于我们把HMMs转变成线性链CRF时的做法。

综合（2.22）和（2.23），可以把log-线性因子CRF的条件分布写成

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{\Psi_A \in F} exp\left\{\sum_{k=1}^{K(A)} \theta_{ak} f_{ak}(\boldsymbol{y}_a, \boldsymbol{x}_a)\right\}. (2.24)$$

另外，许多应用模型常常需要参数绑定。以线性链为例，每一时刻的因子$\Psi_t(y_t, y_{t-1}, \boldsymbol{x}_t)$常常使用相同的权重。为了表示这一情况，我们把G的因子划分成$\mathcal{C} = \{C_1, C_2, \cdots, C_P\}$，其中每个$C_P$是一个**团模板clique template**，是一组共享了特征函数$\{f_{pk}(\boldsymbol{x}_c, \boldsymbol{y}_c)\}_{k=1}^{K(p)}$和参数$\theta_p \in \mathcal{R}^{K(p)}$的因子。一个使用了团模板的CRF可以写成

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{C_p \in \mathcal{C}} \prod_{\Psi_c \in C_p} \Psi_c(\boldsymbol{x}_c, \boldsymbol{y}_c; \theta_p). (2.27)$$

其中每个模板因子是这样参数化的

$$\Psi_c(\boldsymbol{x}_c, \boldsymbol{y}_c; \theta_p) = exp\left\{\sum_{k=1}^{K(p)} \theta_{pk} f_{pk}(\boldsymbol{x}_c, \boldsymbol{y}_c\right\}, (2.26)$$

而归一化函数为

$$Z(\boldsymbol{x}) = \sum_{y} \prod_{C_p \in \mathcal{C}} \prod_{\Psi_c \in C_p} \Psi_c(\boldsymbol{x}_c, \boldsymbol{y}_c). (2.27)$$

这一团模板的记号方法即指明了结构重复，也指明了参数绑定。以线性链CRF为例，典型的团模板 $C_0 = \{\Psi_t(y_t, y_{t-1}, \boldsymbol{x}_t)\}_{t=1}^{T}$ 倍整个网络使用，因而$\mathcal{C} = \{C_0\}$是元素单一的集合。如果相反地，我们希望给每个因子Ψ_t分配独立的参数，就像非齐次HMM，那么需要T个模板，即$\mathcal{C} = \{C_t\}_{t=1}^{T}, C_t = \{\Psi_t(y_t, y_{t-1}, \boldsymbol{x}_t)\}$。

定义通用CRF时，如何给出重复的结构以及参数绑定，是属于最需要考虑的问题。人们推荐了一系列的规范，用于指定团模板，而我们仅仅在这里简单的罗列一下。例如，**动态条件随机场dynamic conditional random field**[140]是一些序列模型，允许在每个时刻拥有多个标签译注：不是指有多个类别，而是有多个变量，而不只是单一的标签，很像动态贝叶斯网络。第二，**关系马尔科夫网relational Markov networks**【142】，是一种用类SQL的语法来指明图结构和参数绑定的通用CRF。**马尔科夫逻辑网Markov logic networks**【113,128】用逻辑式子（logic formulae）来给出无向图的局部函数的分数。实质上，知识库中的每条一阶规则都存在一组参数。MLN的逻辑部分，本质上，可以被看成一种编码惯例，用来指明无向图中的重复结构以及参数绑定。Imperatively define factor graphs【87】使用了完整表达的Turing-complete函数来定义团模板，即给出了模型的结构，也给出了充分统计量f_{pk}。这些函数灵活

地采用了先进的编程思想，包括递归、任意搜索（arbitrary search）、惰性计算以及记忆化。本文采用的团模板的记号，来自于Taskar et al.[142]， Sutton et al. [140]，Richardson 和 Domingos [113]，以及McCallum et al.[87]

2.5特征工程

这一节，我们讲述一些特征工程中的技巧。虽然主要用于语言处理，它们还是很通用的。最主要的权衡很典型——大的特征集可以提高预测的精度，因为决策便捷更加灵活，但却需要更大的内存来保存参数，且可能因为过拟合而降低预测精度。

标签-观测特征Label-observation features.首先，当标签是离散变量，那么团模板C_p的特征f_{pk}常常采用如下的特定形式：

$$f_{pk}(\boldsymbol{y}_c, \boldsymbol{x}_c) = \mathbf{1}_{\{y_c=\tilde{y}_c\}} q_{pk}(\boldsymbol{x}_c). (2.28)$$

也就是说，一个特征只在输出正好为$\tilde{\boldsymbol{y}}_c$时才非零，而一旦如此，便只与输入有关。 我们把具有这种形式的特征称为标签-观测特征。本质上可以这么来理解：特征只依赖于输入$\boldsymbol{x}_c$，但每一种输出都有自己的一组权重。这一特征表示法的计算效率也很高，因为计算每个q_{pk}都可能涉及文本或图片处理，而只需要处理一次，就可用于每一个用到它的特征。为了避免混淆，我们把函数$q_{pk}(\boldsymbol{x}_c)$叫做观测函数，而不是特征。观测函数的例子有“单词$\boldsymbol{x}_t$是大写的”或“单词$\boldsymbol{x}_t$以ing结尾”。

Unsupported Features.使用标签-观测特征可能会带来数量庞大的参数。例如在CRFs的第一个大规模应用中，Sha和Pereira【125】在他们的最佳模型中，使用了3百8十万个参数。其中的很多特城从未在训练数据中出现过——它们总是0。原因在于，许多观测函数只与一小部分的标签相对应。例如在命名实体识别任务中，“单词$\boldsymbol{x}_t$是with，而标签y_t是CITY-NAME”，似乎永远不可能在训练集中为真。我们把它们称为unsupported features。可能很意外，这些特征也可能有用，因为可以给它们赋予负的权重，从而防止给错的标签以高的概率。（降低那些从未出现过的标签序列的分数，将会增加那些出现过的标签序列的概率，所以在后文我们描述的参数估计方法中，会给这些特征以负的权重）。包含unsupported features常常带来精度的少量提升，并以巨大的参数数量为代价。

我们曾利用一个特别的技术，来选择unsupported features的一小部分。这可以看成是使用更少内存来利用的unsupported feature的一次简单探索，可以被称为“unsuported features trick”。它认为许多unsupported features是无用的，因为模型不太可能因为它们的激活而犯错。例如，那个“with”特征不太可能有用，因为with是一个常见的单词，且总是属于OTHER标签（即它不是一个名词）。为了减少参数的数量，我们只保留那些有可能剔除错误的unsupported features。一个简单的方法是：首先训练一个不带unsupported feature的CRF，并在几次迭代后就停下来，使得模型并没有完全训练好。然后考虑那些模型未能给正确答案以高概率的团，给它们增加unsupported features。在上面这个例子中，如果我们发现训练集中有一个样本i，其t位置的序列$\boldsymbol{x}_t^{(i)}$是with，而$\boldsymbol{y}_t^{(i)}$不是“CITY-NAME”原文是$\boldsymbol{y}_t^{(i)}$ is not CITY-NAME。我认为应去掉not。译文则保留了这个not，并且$p(y_t = CITY-NAME|\boldsymbol{x}_T^{(i)}) > \epsilon$时（$\epsilon$时一个阈值），我们增加"with"这一特征。

连线-观测特征和节点-观测特征Edge-Observation and Node-Observation Features.为了减少模型中的特征数量，我们可以只在某些团使用标签-观测特征，而不是全部。最常见的两种标签-观测特征是*连线-观测特征*和*节点-观测特征*。考虑一个具有M个观测函数$\{q_m(\boldsymbol{x})\}, m\in\{1,2,\cdots,M\}$的线性链CRF。如果使用了连线-观测特征，那么每个局部函数可以依赖于全部的观测函数。那么，我们可以使用这样的特征：单词$\boldsymbol{x}_t$是New，y_t是LOCATION 且y_{t-1}也是LOCATION。这会导致模型拥有大量的参数，带来内存消耗和过拟合的缺点。一种解决办法是采用节点-观测特征。使用这一类型的特征，转移因子就是局部函数吧?不在依赖于观测函数。于是我们可以使用类似“y_t是LOCATION，且y_{t-1}是LOCATION”，以及“$\boldsymbol{x}_t$是NEW，且y_t是LOCATION”的特征，而不能使用那种一次把$\boldsymbol{x}_t, y_t, y_{t-1}$都依赖上的特征。连线-观测特征和节点特征都正式地在表2.1中给出了。一般来说，以上两种特征的选择，需要根据具体的问题来定，如需要考虑观测函数的数量，以及数据集的大小。

Table 2.1. Edge-observation features versus node-observation features.

Edge-observation features:

$$f(y_t, y_{t-1}, \mathbf{x}_t) = q_m(\mathbf{x}_t)\mathbf{1}_{\{y_t=y\}}\mathbf{1}_{\{y_{t-1}=y'\}} \qquad \forall y, y' \in \mathcal{Y}, \forall m$$

$$f(y_t, \mathbf{x}_t) = q_m(\mathbf{x}_t)\mathbf{1}_{\{\mathbf{y}_t=y\}} \qquad \forall y \in \mathcal{Y}, \forall m$$

Node-observation features:

$$f(y_t, y_{t-1}, \mathbf{x}_t) = \mathbf{1}_{\{y_t=y\}}\mathbf{1}_{\{y_{t-1}=y'\}} \qquad \forall y, y' \in \mathcal{Y}$$

$$f(y_t, \mathbf{x}_t) = q_m(\mathbf{x}_t)\mathbf{1}_{\{y_t=y\}} \qquad \forall y \in \mathcal{Y}, \forall m$$

Boundary Labels.最后一个问题是如何在边缘上取标签，例如一个序列的开始和结尾，或一张画的边缘。有时，边缘上的标签与其他标签不同。例如，大写字母在一个句子的中间意味着是专有名词，但如果是在句子的开始却没有这样的意味。一个简单的办法，是在标签序列的前面加一个特殊的标签——START。这允许模型学习得到关于边缘的特性。例如，如果连线-观测特征也被使用了，那么像“$y_{t-1} = START$且$y_t = PERSON$且x_t大写”这样的特征，可以表示，大写这一特征在句子的开始时并不是有效的。