单词 bi-gram统计的一种自适应算法

应江黔　　兵藤安昭　　池田尚志

（日本岐阜大学工学部电子情报工学科）

摘要：单词bi-gram在语言中的分布频度是反映词语间搭配，共起关系的重要统计数量．我们针对怎样建立一个词类间的 bi-gram 模型，使之能够用来计算单词bi-gram 的频度这个问题，提出一种能对逐次读入的语料实时处理，不断修正模型参数的自适应算法，初步实验结果表明这一算法在实际运算中呈现的特性与理论分析相符．

An adaptive algorithm for statistical analysis of word bi-grams

Jiangqian Ying　　Yasuaki Hyodo　　Takashi Ikeda

(Department of Electronics and Computer Engineering, Gifu University, Japan)

ABSTRACT: The frequencies of word bi-grams can be used as statistical measures for the description of word association in a natural language. We consider the problem of calculating these frequencies from a statistical model which is composed of a number of word categories, the frequencies of bi-grams of categories and the probabilities of word distribution in categories. An adaptive algorithm is given for estimating the model parameters from a text data base. Preliminary experiment shows that the performance of this algorithm coincides with the theoretical analysis.

1　前　　言

近年来，可利用计算机自动处理的语言库规模越来越大，为运用统计学的方法研究自然语言提供了有利条件。用统计方法处理自然语言的一个重要问题是怎样准确描述和利用词的共起关系．解决这个问题的一个办法是将单词分类，通过词类间的关系来近似表达单词间的关系。当然，这种近似必然会伴随信息量的损失，这种损失的大小是衡量分类方法正确与否的客观标准。

同词之间的共起关系一样，类之间的共起关系亦可用 n-gram 来描述。Brown等人在文献[1]中论述了基于 bi-gram 的分类模型及相应的算法。他们的方法是统计出单词组 (bi-gram) 的频度分布，最初把每一个词都看成一个类，然后逐次把小类合并成大类。每次合并都以信息量损失最小为基准决定合并的对象。在文献 [2] 中，Pereira 等人则采用了逐次将大类分裂成小类的算法．

这些算法都是利用事先对语料库做处理后得到的统计数据来进行计算的。

本文介绍一种不需事先对整个语料库做统计处理的自适应算法。我们先确定类的数目及类别的统计模型，并适当地设定模型的初始参数。然后利用逐次读入的语料实时更新模型的参数。根据最后得到的模型参数，可按概率的大小确定每一个词该属于哪一类。下一节里我们论述模型及算法的理论依据。第 3 节里介绍一个计算实例，最后一节讨论一些相关的问题。

2 类别 bi-gram 模型及自适应算法

假定 $\{w_1,...,w_n\}$ 为单词的集合，$\{c_1,...,c_m\}$ 为类的集合。假定从类 c_i 中产生单词 w_k 的概率为 $P(w_k|c_i)$, i=1,...m, k=1,...,n. 类的 bi-gram c_ic_j 出现的概率设为 $P(c_ic_j)$ i,j=1,...,m。这些概率的数值决定我们的类别模型，我们用 θ 表示这个模型。这个模型隐含了单词的分类概念，因为任何一个单词 w_k 属于类 c_i 的概率可依下式算出

$$P(c_i|w_k)=P(w_k|c_i)P(c_i)/\sum_{j=1}^{m}P(c_j)P(w_k|c_j)$$

其中

$$P(c_i)=\sum_{h=1}^{m}1/2[P(c_ic_h)+P(c_hc_i)].$$

这种分类方法给出分属的可能性的概率数值。我们可按照贝叶斯决策的原理，将 w_k 划分为使 $P(c_i|w_k)$ 最大的类。

从这个模型，可以推算单词 bi-gram w_kw_l 在语言中出现的概率 $P(w_kw_l|\theta)$

$$P(w_kw_l|\theta)=\sum_{i,j=1}^{m}P(c_ic_j)P(w_k|c_i)P(w_l|c_j).$$

我们用 $P(w_kw_l)$ 表示在语言中出现 w_kw_l 的真实概率，从模型推算出的概率 $P(w_kw_l|\theta)$ 可看作 $P(w_kw_l)$ 的近似值。这种近似的精确程度有较客观的数值尺度来衡量。

假设 bi-gram w_kw_l 在语料库 Ω 中出现的次数为 $f(w_kw_l)$，$T=\sum_{k,l=1}^{n}f(w_kw_l)$ 为 bi-gram 出现的总数。由模型产生 Ω 的概率为 $P(\Omega|\theta)=\prod_{k,l}P(w_kw_l|\theta)^{f(w_kw_l)}$.另一方面，因为 $P(w_kw_l)$ 是 w_kw_l 出现的真实概率，Ω 出现的概率应为 $P(\Omega)=\prod_{k,l}P(w_kw_l)^{f(w_kw_l)}$

我们定义 $L(\Omega|\theta)=1/T[\log P(\Omega)-\log P(\Omega|\theta)]=\sum_{k,l}\frac{f(w_kw_l)}{T}\log\frac{P(w_kw_l)}{P(w_kw_l|\theta)}$,

这里 log 表示自然对数。

假定语料库正确反映了所代表的语言的全部统计特性，那么我们有 $P(w_kw_l)=f(w_kw_l)/T$,

所以 $$L(\Omega|\theta)=\sum_{k,l}P(w_kw_l)\log\frac{P(w_kw_l)}{P(w_kw_l|\theta)}.$$

这个量反映了用 $P(w_kw_l|\theta)$代替 $P(w_kw_l)$时产生的信息量的损失．在统计学上称作 Kullback-Leibler 距离。$L(\Omega|\theta)$越小，$P(\Omega|\theta)$越大，即由模型产生语料库的概率越大。在 $\sum_{k,l}P(w_kw_l|\theta)=1$的条件下，$L(\Omega|\theta)\geq 0$.当 $P(w_kw_l|\theta)=P(w_kw_l)$时，$L(\Omega|\theta)=0$. 我们把 $L(\Omega|\theta)$ 当作衡量模型精确程度的标准。

若事先统计出 $P(w_kw_l)$，我们可以采用各种求极值的优化算法去求最优模型的参数．文献［1］及［2］中均采用了这一类算法。

这里我们介绍一种在 $P(w_kw_l)$未知的情况下，实时处理逐次读入的语料，不断更新模型参数的自适应算法．在模式识别的领域中有类似的算法，但基本模型有较大差异（文献［3］）。

为了数学运算上的方便，我们设定下列独立变量（每个变量取值限于正数或零）：

$$x_{11},x_{12},\ldots,x_{mn};y_{11},y_{12},\ldots,y_{mm}.$$

设 $P(w_k|c_i)=x_{ik}\Big/\sum_{l=1}^{n}x_{il}$, $P(c_ic_j)=y_{ij}\Big/\sum_{g,h=1}^{m}y_{gh}$, $i,j=1,\ldots,m;k=1,\ldots,n.$

这些独立变量构成一个 m(m+n) 维的向量：$\theta=(x_{11}x_{12}\ldots x_{mn}y_{11}y_{12}\ldots y_{mm})$.

前面我们用θ表示抽象的模型，由于模型决定于参数，我们仍用θ表示这些参数。

我们的算法很单纯：当读入某个 w_kw_l时，调节 θ使得$\log P(w_kw_l|\theta)$趋于增大。

以下用∇表示对向量 θ求梯度的算子：$\nabla=\nabla_\theta=(\partial/\partial x_{11}\ldots\partial/\partial x_{mn}\partial/\partial y_{11}\ldots\partial/\partial y_{mm})$.

设 ε为一正数，A 为一m(m+n)×m(m+n) 正定 方阵。当读入某一bi-gram $w_{k_0}w_{l_0}$ 时，θ 的变化量为 $\delta\theta_0=\delta\theta(w_{k_0}w_{l_0})=\varepsilon A\nabla\log P(w_{k_0}w_{l_0}|\theta)$.

新的参数向量为 $\theta+\delta\theta_0$。下面我们说明这种简单的算法的确能使得$L(\Omega|\theta)$趋于减少。

θ增加 $\delta\theta_0$后，$\delta\log P(w_kw_l|\theta)=\delta\theta_0\bullet\nabla\log P(w_kw_l|\theta)+o(\varepsilon^2)$,

$\bullet$表示点积，$o(\varepsilon^2)$为一与ε^2成比例的数值。

$L(\Omega|\theta)$的变化量 $\delta L(\Omega|\theta)_{w_{k_0}w_{l_0}}=-\sum_{k,l}P(w_kw_l)\delta\log P(w_kw_l|\theta)$.

由于读入 $w_{k_0}w_{l_0}$的概率为$P(w_{k_0}w_{l_0})$，故$\delta L(\Omega|\theta)$的期望值为

$$\overline{\delta L(\Omega|\theta)}=\sum_{k_0l_0}P(w_{k_0}w_{l_0})\delta L(\Omega|\theta)_{w_{k_0}w_{l_0}}.$$

代入 $\delta L(\Omega|\theta)_{w_{k_0}w_{l_0}}$，容易算出 $\overline{\delta L(\Omega|\theta)} = -\nabla L(\Omega|\theta) \bullet \varepsilon A \nabla L(\Omega|\theta) + o(\varepsilon^2)$.

当 ε 足够小，$\nabla L(\Omega|\theta) \neq 0$ 时，$\overline{\delta L(\Omega|\theta)} < 0$.这就是说，平均来讲 $L(\Omega|\theta)$ 是趋于减少的。

数值 ε 及矩阵A规定了参数的变更速度。以下我们只考虑A为对角阵的情形。各个元素规定了对应的 θ 的成分的变化速度。在下一节里，我们直接指定各个成分 x_{ik}, y_{ij} 的变化速度。

3　对日语语料库的初步实验结果

我们用上述算法做了以下实验。我们选了 2 2 个日语单词作为统计分类的对象:

は、が、の、を、に；
ついて、たいして、したがって、はじめ、めどに；
作る、進める、示した、固めた、発足、明らかに；
方針、対策、計画、検討、準備、組織。

这些单词常出现在日本的有关行政问题的新闻报道中。他们的典型组合有:

対策　に　ついて　検討　を　進める　（关于对策进行商讨）

方針　に　したがって　組織　を　発足　（依然方针建立组织）

等等。这些单词的分类并没有固定的规则。如”準備”、”組織”、”発足”等，即可作名词也可作动词的词干部分。而日语助词”は”、”が”、”の”、”を”、”に”等的语法功能也各不一样，但可主观判断某一分类是否合理，例如上面按行的分类可以认为是合理的。

我们使用的语料库是日本朝日新闻 1991 年 1 月的部分文章，共 302884 个句子。我们实际利用的是上述 2 2 个单词按出现先后构成的bi-gram系列，共133890 个 bi-gram （包含重复）。

我们设定类的数目 m=6，参数的初始值 $x_{ik} = 1/n = 1/22, y_{ij} = 1/m^2 = 1/36.$

设 $\varepsilon_1 = 0.005, \varepsilon_2 = 0.05.$

当读入某一 $w_{k_0}w_{l_0}$ 时，设 w_{k_0} 出现在 bi-gram 左部的累计次数为 $g(w_{k_0})$，w_{l_0} 出现在右部的累计次数为 $h(w_{l_0})$。那么，x_{ik_0} 的更新速度设为 $\dfrac{\varepsilon_1 P(w_{k_0}w_{l_0}|\theta)}{2\log g(w_{k_0}) + \log h(w_{l_0})}$，

x_{jl_0} 的更新速度设为 $\dfrac{\varepsilon_1 P(w_{k_0}w_{l_0}|\theta)}{\log g(w_{k_0}) + 2\log h(w_{l_0})}$，$y_{ij}$ 的更新速度设为 $\dfrac{\varepsilon_2 P(w_{k_0}w_{l_0}|\theta)}{\log g(w_{k_0}) + \log h(w_{l_0})}$。

更新后的 y_{ij} 可能出现负值，设 $z = \min\{y_{ij}\}$,我们从每个 y_{ij} 减去 z。

计算结果:

据初始设定的参数算出的 Kullback-Leibler距离为 $L(\Omega|\theta) = 3.97$.经过133890次参术变更后

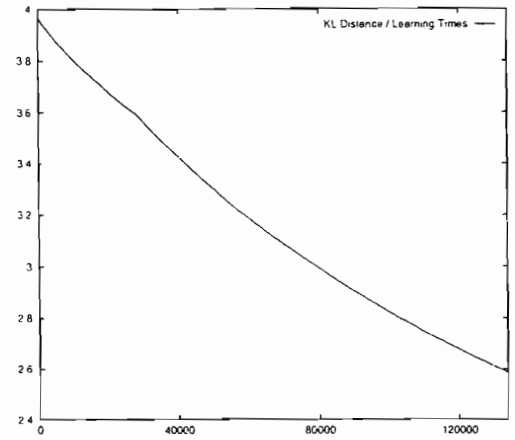


Figure1. Reduction of KL distance

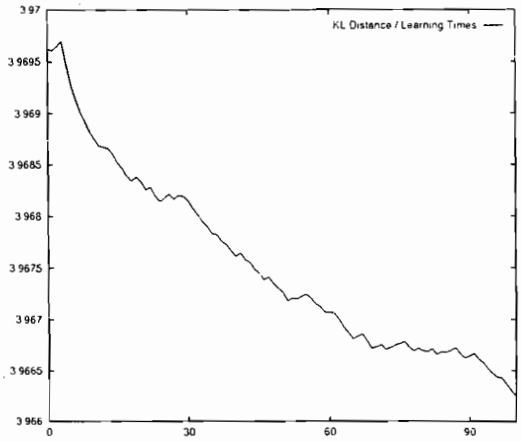
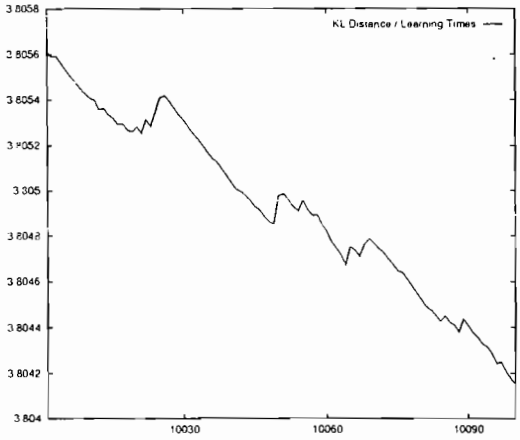


Figure2(a). Learning at the beginning



Fugure2(b). Learning in the middle

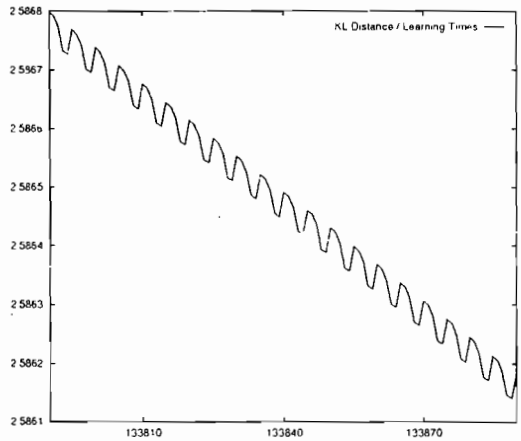


Figure2(c). Learning at the end

降为2.586. 图 1 反映了 $L(\Omega|\theta)$变化的整体过程。图 2 显示了变化的局部情况，可以看出不是每一次学习都使 $L(\Omega|\theta)$减小，但总的趋势是减小。在学习的后面阶段，$L(\Omega|\theta)$的变化速度减弱，而且局部变化呈一定的规律性，这种规律性可能是由于变更参数时的特殊处理所引起。

按第 2 节的分类原则，初始阶段所有的单词均被划为第 6 类，最后的分类结果如下:

$c_1 = \varphi$
$c_2 =$ {初め　対して}
$c_3 =$ {は}
$c_4 =$ {示した　作る　固める　明らかに　進める　組織　発足　めどに　したがって}
$c_5 =$ {方針　検討　対策　準備　計画}
$c_6 =$ {の　を　に　が　ついて}

这个结果我们认为还是反映了日语的一些重要语法特征。

4　讨论及结束语

本文论述了单词类别 bi-gram 模型的一种自适应算法。这一算法的特点是能对读入的数据实时处理，不断修正模型参数。我们的初步实验结果表明这一算法在实际运算中表现出的举动与理论分析基本相符。我们的实验限于 2 2 个单词，下一步打算对更多的单词进行类似的实验。另外，也打算用 tri-gram 做同样的实验。

用统计模型描述自然语言，是自然语言处理的一个重要方法。但对于不同的领域，语言有不同的统计特性。我们不大能够指望某一固定的模型能表达千变万化的语言现象。但可以设想有较为稳定的模型，将此模型用于某一特定的领域时，可以自动调整参数，以得到合适的模型。

对于以 bi-gram 为基础的简单模型，本文提出了一种自适应算法。对于概率语境自由文法(Probabilistic Context Free Grammar)等较复杂的模型，类似的算法是否可行，将是我们今后的课题。

参考文献

[1] P.F.Brown et al., "Class-Based n-gram Models of Natural Language", Computational Linguistics, Vol.18, No.4, p.467-479 (1992)

[2] F.Pereira et al., "Distributional Clustering of English Words", Proceedings of the 31st Annual Meeting of the ACL, p.183-190 (1993)

[3] S.Amari, "A Theory of Adaptive Pattern Classifiers", IEEE Transactions on Electronic Computer, EC-16, p.299-307 (1967)