消除溢出问题的精确 Baum-Welch 算法*

贾宾[1] 朱小燕[2,3] 罗予频[1] 胡东成[1]

[1](清华大学自动化系 北京 100084)

[2](智能技术与系统国家重点实验室 北京 100084)

[3](清华大学计算机科学与技术系 北京 100084)

E-mail: luo@tsinghua.edu.cn

摘要 Baum-Welch 算法是在语音领域中用于 HMM(hidden Markov model)模型参数训练的最基本方法之一.但它在多样本训练时存在着严重的上、下溢问题,需要不断地人工介入来调整中间参数.该文提出了一种新的能消除上、下溢问题的 Baum-Welch 改进算法.该算法不但摆脱了人工介入,保证了计算的精度,而且不会带来过大的计算和存储要求.实验结果表明了这种新算法的有效性.

关键词 隐马尔可夫模型,Baum-Welch 算法,溢出,语音识别.

中图法分类号 TP391

隐马尔可夫模型(hidden Markov model,简称 HMM)[1]是现代语音识别主流技术的基础,被广泛地应用于识别系统的声学底层,在上层的语言模型层也常常依靠它.训练 HMM 最可靠的算法是 Baum-Welch 算法[2],因此,对它的研究具有重要意义.

虽然 Baum-Welch 算法的使用很普遍,但它在实际应用中还存在着严重的上、下溢问题.这个问题起因于计算机所能表达的浮点数范围的有限性:在训练中要进行对状态输出概率密度的大量积累计算,状态输出概率密度常常取比较大或比较小的值,使积累值可能超出计算机浮点数范围,从而造成上、下溢问题.对于用单样本进行训练,这个问题已被解决[3],不需要人的参与就可以自动进行.而实际上,为了充分训练 HMM 模型参数,每个模型常常用多个样本进行训练.对于此时的上、下溢问题,传统上常用人工介入的办法[3],在训练过程中要求人不断地重复调整参数,才能避免出现上、下溢现象.但这需要大量的人力以及调整者的经验.本文从分析 Baum-Welch 算法出现上、下溢的原因出发,提出了相应的改进算法,从而在理论上彻底解决了多样本的上、下溢问题.

1 Baum-Welch 算法中的上、下溢分析

为了后面论述的方便,先对本文中 HMM 的参数代号进行规定:一个 HMM 模型共有 N 个状态(S_1,S_2,...,S_N),状态 S_i 的初始概率分布为 a_i,状态 S_i 到 S_j 的状态转移概率为 A_{ij},状态 S_j 输出观测矢量 y 的条件概率为 $p_{S_j}(y|S_j)$.另外,某观测矢量序列 $Y^{(k)}$ 共有 $T^{(k)}$ 帧($Y_1^{(k)}$,$Y_2^{(k)}$,...,$Y_{T^{(k)}}^{(k)}$).

根据文献[3],在对多样本进行训练时,参数估算公式可以是

$$\hat{\lambda}=\frac{\sum_{k=1}^{K}\left[\sum_{t}\alpha_t^{(k)}(i)\beta_t^{(k)}(j)V^{(k)}(i,j,t)\right]}{\sum_{k=1}^{K}\left[\sum_{t}\alpha_t^{(k)}(i)\beta_t^{(k)}(j)W^{(k)}(i,j,t)\right]},\tag{1}$$

* 本文研究得到国家自然科学基金(No.69982005)资助.作者贾宾,1971 年生,博士,助教,主要研究领域为语音识别,数字信号处理,人工情感.朱小燕,女,1957 年生,博士,副教授,主要研究领域为模式识别,信号处理.罗予频,1959 年生,博士,副教授,主要研究领域为多智能体并行处理,模式识别,图论.胡东成,1946 年生,教授,博士生导师,主要研究领域为自动测试,故障诊断,可靠性.

本文通讯联系人:罗予频,北京 100084,清华大学自动化系

本文 1999-03-25 收到原稿,1999-06-01 收到修改稿

其中 $\hat{\lambda}$ 为迭代计算模型参数的估算值,它可以是 HMM 的 a_{ij},A_{ij}或者是 $p_{S_j}(y|S_j)$的均值或方差;K 为训练样本总数;右边各元素右上角的(k)代表该元素对应于第 k 个训练样本;$V^{(k)}(i,j,t)$和 $W^{(k)}(i,j,t)$都是前次迭代值 λ 的函数;$B^{(k)}$和 $D^{(k)}$分别代表相应的求和空间,且

$$\alpha_t^{(k)}(j)=\sum_{i=1}^{N}A_{ij}p_{x_j\cdot S_j}(y_t^{(k)}\ S_j)\alpha_{t-1}^{(k)}(i),\ j=1,\ldots,N,\ t=2,\ldots,T^{(k)},$$

$$\alpha_1^{(k)}(j)=a_j p_{x_1\cdot S_j}(y_1^{(k)}|S_j),$$

$$\beta_t^{(k)}(i)=\sum_{j=1}^{N}A_{ij}p_{x_{t+1}^{(k)}\cdot S_j}(y_{t+1}^{(k)}|S_j)\beta_{t+1}^{(k)}(j),\ j=1,\ldots,N,\ t=1,\ldots,T^{(k)}-1,$$

$$\beta_{T^{(k)}}^{(k)}(j)=1,j=1,\ldots,N.$$

显而易见,

$$\alpha_{(k,t)}(j)=\sum_{x_0^{(k)},x_1^{(k)},\ldots,x_t^{(k)}|x_t^{(k)}=S_j}\alpha_0^{(k)}(x_0^{(k)})\Big[\prod_{\tau=1}^{t}A_{x_\tau^{(k)}\cdot x_\tau^{(k)}}\Big]\Big[\prod_{\tau=1}^{t}p(y_\tau^{(k)}|x_\tau^{(k)})\Big],\tag{2}$$

$$\beta_t^{(k)}(i)=\sum_{x_t^{(k)},x_{t+1}^{(k)},\ldots,x_{T^{(k)}-1}^{(k)},x_{T^{(k)}}^{(k)}=S_j}\Big[\prod_{\tau=t+1}^{T^{(k)}}A_{x_{\tau-1}^{(k)}\cdot x_\tau^{(k)}}\Big]\Big[\prod_{\tau=t+1}^{T^{(k)}}p(y_\tau^{(k)}|x_\tau^{(k)})\Big].\tag{3}$$

状态输出概率密度函数 $p_{S_j}^{(k)}(y|S_j)$的值有很大的变化范围,当训练样本的帧数 $T^{(k)}$超过一定范围(经验值为 200 帧左右)时,$\alpha_t^{(k)}(j)$,$\beta_t^{(k)}(i)$的取值可能会很小或很大,超过计算机所能表示的最大或最小浮点数的范围,从而出现上、下溢问题.另外,当 $\alpha_t^{(k)}(j)$,$\beta_t^{(k)}(i)$的取值小到一定程度时,式(1)的分母会因丧失精度而趋近 0,使式(1)计算值很大,超出计算机的最大浮点数,因而造成上溢.

以前人们采用的方法是在计算 $\alpha_t^{(k)}(j)$,$\beta_t^{(k)}(i)$的过程中共同乘以与各训练样本无关的比例因子[3],其缺点是需要人工调整;或者是乘以与各训练样本有关的比例因子[3],摆脱人工干预,但因额外加入的比例因子无法消去而造成精确度下降.

2 改进算法

改进算法的思想是自动计算与各训练样本无关的比例因子或规整系数.这些规整系数不但在计算中调整了累积概率密度的数值大小,防止上、下溢问题的出现,而且在式(1)中被消去,从而保证了计算精度.实际上,各个训练样本的长度往往不等,本节将给出采用不等长多训练样本集(variable duration set,简称 VDS)的改进算法.

样本长度不相同的正规化方法的基本思想是在全体训练样本范围内确定规整系数.其关键是如何处理样本帧数不等的问题.为了描述其算法,将式(1)写成

$$\hat{\lambda}=\frac{\sum_{k=1}^{K}v^{(k)}(i,j)}{\sum_{k=1}^{K}v^{(k)}(i,j)},\tag{4}$$

其中

$$v^{(k)}(i,j)=\sum_{B^{(k)}}\alpha_{t-1}^{(k)}(i)\beta_t^{(k)}(j)V^{(k)}(i,j,t)=\sum_{B_1^{(k)}}\sum_{t=1}^{T^{(k)}}\alpha_{t-1}^{(k)}(i)\beta_t^{(k)}(j)V^{(k)}(i,j,t),$$

$$v^{(k)}(i,j)=\sum_{D^{(k)}}\alpha_{t-1}^{(k)}(i)\beta_t^{(k)}(j)W^{(k)}(i,j,t)=\sum_{D_1^{(k)}}\sum_{t=1}^{T^{(k)}}\alpha_{t-1}^{(k)}(i)\beta_t^{(k)}(j)W^{(k)}(i,j,t),$$

$$B_1^{(k)}=B^{(k)}-\{t|1\leqslant t\leqslant T^{(k)}\},\quad D_1^{(k)}=D^{(k)}-\{t|1\leqslant t\leqslant T^{(k)}\}.$$

为了防止上、下溢现象的出现,在计算规整系数时,要考虑所有将被规整的量.规整系数 φ 的计算式为

$$\varphi=\Big[\sum_{i,j=1}^{N}v(i,j)+\sum_{i,j=1}^{N}v(i,j)+\sum_{w\in H}\sum_{i=1}^{N}\tilde{\alpha}_t^{(w)}(i)\Big]^{-1},\quad H\in\{w|1\leqslant w\leqslant K,T^{(w)}>t\},\tag{5}$$

其中 $v(i,j)$和 $v(i,j)$分别用 φ_{t-1}正规化 $\tilde{v}(i,j)$和 $\tilde{v}(i,j)$得到.

$$v(i,j)=\varphi_t\tilde{v}(i,j)=\varphi_t\sum_{T^{(k)}<t}\hat{v}^{(k)}(i,j), \tag{6}$$

$$v(i,j)=\varphi_t\tilde{v}(i,j)=\varphi_t\sum_{T^{(k)}<t}\hat{v}^{(k)}(i,j), \tag{7}$$

其中 $\hat{v}(i,j)$ 和 $\hat{v}(i,j)$ 分别相当于用 $\varphi_f,f=1,\ldots,t-2$ 正规化后的 $v^{(k)}(i,j)$ 和 $v^{(k)}(i,j)$. 在每次计算完式(5)后,不仅用 φ_t 规整 $\tilde{\alpha}(\cdot)$,同时也应规整 $v(i,j)$ 和 $v(i,j)$.

算法如下:

初始化:$v(i,j)\leftarrow 0,v(i,j)\leftarrow 0,i,j=1,\ldots,N$

Loop I. 循环变量为帧号 t,从 1 依次递增到最长的第 M 个样本帧数 $T^{(M)}$

Step 1. 计算各样本的 $\alpha_t^{(k)}(i),T^{(k)}\geqslant t$

Step 2. 计算规整系数 φ_t,规整 $\alpha_t^{(k)}(i),v(i,j),v(i,j)$

Step 3. Loop II. 循环变量为样本号 k,依次处理 $T^{(k)}-t$ 的样本

Step 1. 计算 $\beta_f^{(k)}(i),\hat{\beta}_f^{(k)}(i),f=t,\ldots,1$ 以及 $\hat{v}^{(k)}(i,j)$ 和 $\hat{v}^{(k)}(i,j)$

Step 2. $v(i,j)\leftarrow v(i,j)+\tilde{v}^{(k)}(i,j),v(i,j)\leftarrow v(i,j)+\hat{v}^{(k)}(i,j),i,j=1,\ldots,N$

Loop II control

Loop I control

计算 $\hat{\lambda}=\dfrac{v(i,j)}{v(i,j)}$.

可见,新算法引入了累积变量 $v(i,j)$ 和 $v(i,j)$ 来存储帧长小于当前处理帧数的样本的 $\hat{v}^{(k)}(i,j)$ 和 $\hat{v}^{(k)}(i,j)$,$v(i,j)$ 和 $v(i,j)$ 中所包含的短样本继续与其他未被包含的长样本一起参与正规系数的计算,并且一起被同样地正规化.

3 实 验

本文进行了样本长度不相同的正规化算法的实验,分别与手工调整参数法以及较流行的 Viterbi 训练算法进行了对比.

汉语数字识别,训练样本为 10 个男子,每人分别说孤立的数字 0~9 个一次,测试样本是另外的 10 个人. 结果见表 1.

Table 1 Recognition of digital

表 1 数字识别

	VDS normalization①	Adjusting parameters by hand②
Error rate③(%)	1	1.5

①VDS 正规化方法,②手工调整参数法,③错误率.

由上表可见,样本长度不相同的正规化方法与手工调整参数法在数字识别上效果相当. 但样本长度不相同的正规化方法摆脱了繁重的手工参数调整工作,实际上提高了训练速度. 另外,用原始的 Baum-Welch 方法训练,因溢出而无法进行.

在应用中,人们也常采用 Viterbi 训练算法[4]. Viterbi 训练算法的特点是较为简单,且不会产生溢出. 本文对样本长度不相同的正规化算法与传统的 Viterbi 训练算法也进行了对比. 训练样本为 30 人提供的连续语音,测试样本为另外 10 人,进行 412 个汉语音素的训练和测试. 结果见表 2.

Table 2 Recognition of syllable

表 2 音素识别

	VDS normalization①	Viterbi method②
Training time(h)③	10.3	8.4
Error rate④(%)	5.5	16

①VDS 正规化方法,②Viterbi 方法,③训练时间(小时),④错误率.

由上表可见,虽然正规化方法比 Viterbi 方法更耗时,但错误率却大大降低.这是因为 Viterbi 训练算法是先将训练语音样本分段,然后再对每段特征矢量串进行聚类,在这个过程中分别会产生两次误差.首先,Viterbi 方法分段的精度比较差,分段后可能将对应于不同状态的矢量作为同一个状态分到一起(这种问题对长训练样本尤其突出);其次,聚类时初始值对聚类结果的影响常常很大,当初始值选择得不好时,聚类效果就很差,而初始值的选取具有很大的偶然性.

4 结　论

本文提出了 Baum-Welch 改进算法,新算法彻底克服了 Baum-Welch 算法在多训练样本训练时频繁出现的上、下溢问题.它跳出了传统方法中的在单个样本中确定调整系数的局限,在全局上用所有的样本确定共同的系数.改进算法最终使人不必参与训练过程,并且此算法不需要过大的计算量和存储空间,在进行长样本训练时,这种新算法也因不存在 Viterbi 训练算法所特有的分段和聚类误差,而优于 Viterbi 训练算法.

参考文献

1 Rabiner L. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 1989,77(2):257~286

2 Huang X D, Jack M. Semi-Continuous hidden Markov models for speech signals. Computer Speech and Language, 1989,3(3):239~251

3 Yang Xing-jun, Chi Hui-sheng. Speech Digital Signal Processing. Beijing: Electronic Industry Press, 1995
(杨行峻,迟惠生. 语音信号数字处理. 北京:电子工业出版社,1995)

Accurate Baum-Welch Algorithm Free from Overflow

JIA Bin[1]　ZHU Xiao-yan[2,3]　LUO Yu-pin[1]　HU Dong-cheng[1]

[1](*Department of Automation　Tsinghua University　Beijing*　100084)

[2](*State Key Laboratory of Intelligent Technology and Systems　Beijing*　100084)

[3](*Department of Computer Science and Technology　Tsinghua University　Beijing*　100084)

Abstract　Baum-Welch algorithm, which is often troubled with overflow, is one of the basic methods in the field of speech signal processing. People have to adjust the inner parameters constantly. So in this paper, a modified Baum-Welch algorithm is presented to avoid the overflow completely. With this algorithm manual adjustment is not needed and the computation accuracy is guaranteed. There is no significant extra cost of computation and storage. The feasibility of the new algorithm is shown in the experiment.

Key words　HMM (hidden Markov model), Baum Welch algorithm, overflow, speech recognition.