等价于 MAP 的 SOVA 译码方法

田志刚　　郭文彬　　杨大成
(北京邮电大学 93#信箱电信工程学院　北京　100876)

摘　要　不同于 MAP(Maximum *A Posteriori*)算法，SOVA(Soft-Output Viterbi Algorithm)算法的软输出不是真正意义上的后验概率，很少有文献给出 SOVA 算法的完整数学解释。该文给出了一种完整的 SOVA 的数学表达形式，并从 SOVA 的数学表达出发推导出了两种等价于 MAP，具有 SOVA 形式的译码方法，一种是 Li 等人(1995)给出的适用于连续传输的最佳软输出算法(Optimal Soft output Algorithm, OSA)；后一种是对 OSA 算法的改进，后者可以得到与前者等价的软输出，但是降低了运算复杂度。

关键词　软入软出译码，软输出维特比译码，最大后验概率译码，最优软输出译码，最大似然

中图分类号：TN919.3+2　　**文献标识码**：A　　**文章编号**：1009-5896(2006)07-1270-04

MAP Decoding Methods Derived from SOVA

Tian Zhi-gang　　Guo Wen-bin　　Yang Da-cheng
(*Telecommunication Engineering School, Beijing University of Posts&Telecomm,, Beijing* 100876　*China*)

Abstract　Different from MAP, the soft output of SOVA is not the exact *a posteriori* probability, and few literatures describe SOVA in mathematical form. This paper gives a new comprehensive mathematical expression of SOVA, and deduces two MAP-equivalent SISO decoding methods in the form of SOVA. The first method is just OSA algorithm given by Li et al, (1995)which is suitable for continuous decoding. The other one is the improved version of OSA，whose output is equivalent to that of OSA with the decrease of complexity.

Key words　SISO，SOVA，MAP，OSA，Maximum Likelihood (ML)

1　引言

具有软输入、软输出(Soft Input Soft Output SISO)的译码方法是在Turbo码的发展过程中获得大量的研究的。常用的SISO译码方法有两类：SOVA (Soft-Output Viterbi Algorithm)和MAP (Maximum *A Posteriori*)。MAP根据所有可能路径计算最优软信息，但是具有较高的复杂度，需要前向和后向两次迭代过程；一般意义上的SOVA主要根据幸存路径来计算软信息，是一种次优算法，具有中等的复杂度，只需要在前向译码时回溯一个固定长度。SOVA最初由Hagenauer-Hoeher[1]发表，主要用于二进制符号软信息的输出，后来Hoeher给出了一种针对非二进制(non-binary)符号的SOVA[2]，Battail对SOVA的软信息更新规则进行了修正[3]。除了Bahl给出的symbol by symbol MAP译码(亦称为BCJR译码)方法外[4]，还陆续出现了一些具有不同形式、适用于不同系统的MAP译码方法[5,6]。文献[5]中给出的(OSA Optimal Soft output Algorithm)算法适用于连续传输系统，与SOVA一样只需要一个前向迭代过程，并实时对此前的符号软信息进行更新；当输出时延等于块长度时OSA的软输出等价于BCJR算法。鉴于前述的关于SOVA的文献缺少对SOVA的完整数学解释，本文将给出一种SOVA的数学表达，并从这个数学表达出发揭示了OSA与SOVA的联系；在此基础上，推导出了一种OSA的改进算法，可以得到与OSA等价的软输出，但是降低了运算复杂度。本文只讨论码间干扰 (ISI) 信道中的信号检测的情况，但是结论具有一般意义，文献[7]证明了OSA适用于其他情形。

2　一些概率定义

假设接收机接收到一段长度为 N 的序列，序列中的每个输入符号是 M 进制的，等效离散信道的记忆深度为 L，则接收序列表达为 $\boldsymbol{Y}_{(N)}=\{y_0,\cdots,y_N\}$，信源输入序列为 $\boldsymbol{I}_{(N)}=\{I_0,\cdots,I_N\}$，其中 $I_i\in\{U_0,\cdots,U_{M-1}\}$，接收信号与输入符号之间的关系：

$$y_k=\sum_{l=0}^{L}h\cdot I_{k-l}+\eta_k \tag{1}$$

其中 h_l 是等价离散信道，η_k 是叠加在接收侧的白噪声，方差为 σ^2。在白噪声的假设下，似然函数表达为

$$\left.\begin{aligned}P\left(Y_{(N)}\middle|\boldsymbol{I}_{(N)}\right)&=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-D\left(\boldsymbol{Y}_{(N)},\boldsymbol{I}_{(N)}\right)\big/2\sigma^2\right)\\D\left(\boldsymbol{Y}_{(N)},\boldsymbol{I}_{(N)}\right)&=\sum_{n=1}^{N}\left|y_n-\sum_{l=0}^{L}h_l\square I_{n-l}\right|^2\end{aligned}\right\} \tag{2}$$

2004-11-15 收到，2005-04-28 收回

假设可能发送序列的全集 $A\left(\boldsymbol{I}_{(N)}\right)$，一个特定序列的后验概率(后面简称概率)可以表达为似然函数的表达式：

$$P\left(\boldsymbol{I}_{(N)}^{*}\middle|\boldsymbol{Y}_{(N)}\right)=\frac{P\left(\boldsymbol{I}_{(N)}^{*},\boldsymbol{Y}_{(N)}\right)}{P\left(\boldsymbol{Y}_{(N)}\right)}=\frac{P\left(\boldsymbol{Y}_{(N)}\middle|\boldsymbol{I}_{(N)}^{*}\right)\cdot P\left(\boldsymbol{I}_{(N)}^{*}\right)}{\sum\limits_{\boldsymbol{I}_{(N)}\in A\left(\boldsymbol{I}_{(N)}\right)}P\left(\boldsymbol{Y}_{(N)}\middle|\boldsymbol{I}_{(N)}\right)\cdot P\left(\boldsymbol{I}_{(N)}\right)} \tag{3}$$

MLSE 接收机的目标是：找出具有最大似然概率的可能序列，在白噪声的假设下也就是找到具有最小欧氏距离的可能序列。

$$\boldsymbol{I}_{(N)}^{*}=\underset{\boldsymbol{I}_{(N)}}{\mathrm{Max}}\left\{P\left(\boldsymbol{Y}_{(N)}\middle|\boldsymbol{I}_{(N)}\right)\right\}\Leftrightarrow\boldsymbol{I}_{(N)}^{*}=\underset{\boldsymbol{I}_{(N)}}{\mathrm{Max}}\left\{D\left(\boldsymbol{Y}_{(N)},\boldsymbol{I}_{(N)}\right)\right\} \tag{4}$$

MAP准则：根据贝叶斯准则，在给定接收序列的条件下，第k个符号是U_m的条件概率为

$$P\left(I_k=U_m\middle|\boldsymbol{Y}_{(N)}\right)=\sum_{\boldsymbol{I}_{(N)}\in A\left(\boldsymbol{I}_{(N)}\right)\cup I_k=U_m}P\left(\boldsymbol{I}_{(N)}\middle|\boldsymbol{Y}_{(N)}\right) \tag{5}$$

另外定义一些状态，设 θ_k 表示序列在第k个时刻所处的状态，总共的状态数为M^L，状态可以表征为此时刻之前L个符号的组合：$\theta_k=\left(I_{k-L},I_{k-L+1},\cdots,I_{k-1}\right)$，设 $\theta_n\left(\boldsymbol{I}_{(k)}\right)$ 表示序列在第n个时刻所处的状态。则序列在第k个时刻处于某个状态的概率：

$$P\left(\theta_k^{*}\middle|\boldsymbol{Y}_{(k)}\right)=\sum_{\theta_k\left(\boldsymbol{I}_{(k)}\right)=\theta_k^{*}}P\left(\boldsymbol{I}_{(k)}\middle|\boldsymbol{Y}_{(k)}\right) \tag{6}$$

求和号里面的概率可以根据式(3)计算。第 k 个时刻处于某个状态的所有序列中的某个序列的概率：

$$P\left(\boldsymbol{I}_{(k)}\middle|\theta_k^{*},\boldsymbol{Y}_{(k)}\right)=\frac{P\left(\boldsymbol{I}_{(k)},\boldsymbol{Y}_{(k)}\right)}{P\left(\theta_k^{*},\boldsymbol{Y}_{(k)}\right)}=\frac{P\left(\boldsymbol{I}_{(k)}\middle|\boldsymbol{Y}_{(k)}\right)}{P\left(\theta_k^{*}\middle|\boldsymbol{Y}_{(k)}\right)};\theta_k\left(\boldsymbol{I}_{(k)}\right)=\theta_k^{*} \tag{7}$$

分子、分母中的概率可以根据式(3)、式(6)计算。

3　SOVA 的数学表达

SOVA 对 VA 的改动有两点：

(1)判决所依赖度量(Metric)的改变，是Fossorier在提出SOVA与Max-Log-MAP等效时候对SOVA的改进[8]，主要是将度量从似然函数改为后验概率；如果输入符号等概并忽略噪声方差，则等同于常规的VA (Viterbi Algorithm)；

(2)需要对幸存路径上的符号的软信息进行回溯更新；有Hagenauer更新规则[1，2]和Battail更新规则[3]两种，后者更准确。

对于第(1)点，VA 的度量是似然函数：

$$P\left(\boldsymbol{Y}_{(k)}\middle|\boldsymbol{I}_{(k)}^{*}\right)=P\left(y_k\middle|\boldsymbol{I}_{(k)}^{*}\right)\bullet P\left(\boldsymbol{Y}_{(k-1)}\middle|\boldsymbol{I}_{(k)}^{*}\right)=P\left(y_k\middle|\boldsymbol{I}_{(k)}^{*}\right)\bullet P\left(\boldsymbol{Y}_{(k-1)}\middle|\boldsymbol{I}_{(k-1)}^{*}\right) \tag{8}$$

上面用到了白噪声样本的独立特性。其中 $\boldsymbol{I}_{(k-1)}^{*}$ 表示 $\boldsymbol{I}_{(k)}^{*}$ 的前 $k-1$ 个符号，也就是 $\boldsymbol{I}_{(k)}^{*}=\left(\boldsymbol{I}_{(k-1)}^{*},\boldsymbol{I}_{k}^{*}\right)$。公式中的两项都可以表达为欧氏距离的形式：

$$\begin{aligned}\ln\left(P\left(\boldsymbol{Y}_{(k)}\middle|\boldsymbol{I}_{(k)}^{*}\right)\right)&=\ln\left(P\left(y_k\middle|\boldsymbol{I}_{(k)}^{*}\right)\right)+\ln\left(P\left(\boldsymbol{Y}_{(k-1)}\middle|\boldsymbol{I}_{(k-1)}^{*}\right)\right)\\&=C+\frac{D\left(y_k,\left\{\theta_k\left(\boldsymbol{I}_{(k)}^{*}\right),\theta_{k-1}\left(\boldsymbol{I}_{(k)}^{*}\right)\right\}\right)+D\left(\boldsymbol{Y}_{(k-1)},\boldsymbol{I}_{(k-1)}^{*}\right)}{2\sigma^2}\end{aligned} \tag{9}$$

其中 C 是与 $\boldsymbol{I}_{(k)}^{*}$ 无关的常数，SOVA 中将度量修正为如同式(3)的后验概率：

$$P\left(\boldsymbol{I}_{(k)}^{*}\middle|\boldsymbol{Y}_{(k)}\right)=\frac{P\left(\boldsymbol{Y}_{(k)}\middle|\boldsymbol{I}_{(k)}^{*}\right)\cdot P\left(\boldsymbol{I}_{(k)}^{*}\right)}{P\left(\boldsymbol{Y}_{(k)}\right)} \tag{10}$$

因为 $P\left(\boldsymbol{Y}_{(k)}\right)$ 对所有可能的 $\boldsymbol{I}_{(k)}^{*}$ 是一样的，所以可以在度量的计算中忽略，考虑分子项：

$$\begin{aligned}&\ln\left(P\left(\boldsymbol{Y}_{(k)}\middle|\boldsymbol{I}_{(k)}^{*}\right)\cdot P\left(\boldsymbol{I}_{(k)}^{*}\right)\right)\\&=\ln\left(P\left(\boldsymbol{Y}_{(k-1)}\middle|\boldsymbol{I}_{(k-1)}^{*}\right)\cdot P\left(\boldsymbol{I}_{(k-1)}^{*}\right)\right)+\ln\left(P\left(y_k\middle|\boldsymbol{I}_{(k)}^{*}\right)\right)+\ln\left(I_k^{*}\right)\end{aligned} \tag{11}$$

在Turbo码的译码中，可以通过前一级译码器得到 $\ln(I_k^{*})$；一般情况下可以假设发送信源等概，$\ln(I_k^{*})$ 可以忽略。这样修正过的SOVA需要知道噪声方差，如果再省略掉噪声方差，则SOVA的度量退化为VA的度量。因为在VA过程中第 $k-1$ 个时刻经过判决，选出了M^L个幸存路径 $\boldsymbol{I}_{(k-1)}^{*}$，则第$k$个时刻的判决和软信息计算只能基于 $\boldsymbol{I}_{(k-1)}^{*}$ 派生出来的竞争路径 $\boldsymbol{I}_{(k)}^{C}=(\boldsymbol{I}_{(k-1)}^{I},I_k),I_k\in\left\{U_0,\cdots,U_{M-1}\right\}$，所以修正式(6)：

$$P'\left(\theta_k^{*}\middle|\boldsymbol{Y}_{(k)}\right)=\sum_{\theta\left(\boldsymbol{I}_{(k)}^{C}\right)=\theta_k^{*},\boldsymbol{I}_{(k)}^{C}\in\left\{\left(\boldsymbol{I}_{(k-1)}^{i},I_k\right)\right\}}P\left(\boldsymbol{I}_{(k)}\middle|\boldsymbol{Y}_{(k)}\right),i\in\left[1,M^L\right] \tag{12}$$

相应的式(7)也可以改写，第 k 个时刻进入 θ_k^{*} 的所有竞争序列中的某个序列 $\boldsymbol{I}_{(k)}^{C}:\theta_k\left(\boldsymbol{I}_{(k)}^{C}\right)=\theta_k^{*}$ 的概率：

$$\begin{aligned}P'\left(\theta_{K-1}\left(\boldsymbol{I}_{(k)}^{C}\right)\middle|\theta_k^{*},\boldsymbol{Y}_{(k)}\right)&=P'\left(\boldsymbol{I}_{(k-1)}^{i}\middle|\theta_k^{*},\boldsymbol{Y}_{(k)}\right)\\&=P'\left(\boldsymbol{I}_{(k)}^{C}\middle|\theta_k^{*},\boldsymbol{Y}_{(k)}\right)=\frac{P\left(\boldsymbol{I}_{(k)}^{C}\middle|\boldsymbol{Y}_{(k)}\right)}{P'\left(\theta_k^{*}\middle|\boldsymbol{Y}_{(k)}\right)}\end{aligned} \tag{13}$$

其中 $\boldsymbol{I}_{(k-1)}^{i}$ 表示进入状态 θ_k^{*} 的某一条竞争路径 $\boldsymbol{I}_{(k)}^{C}$ 的前 $k-1$ 个符号组成的序列，因为进入状态 θ_k^{*} 的 $\boldsymbol{I}_{(k)}^{C}$ 与它对应的 $\boldsymbol{I}_{(k-1)}^{i}$ 以及 $\theta_{k-1}\left(\boldsymbol{I}_{(k)}^{C}\right)$ 是一一对应的，所以有上面的等式。根据式(11)计算进入 θ_k^{*} 状态的 M 个竞争路径 $\boldsymbol{I}_{(k)}^{C}$ 的度量，并经过 ACS (Add-Compare-Select) 过程选择一条 θ_k^{*} 状态下的幸存路径 $\boldsymbol{I}_{(k)}^{*}=\left(\boldsymbol{I}_{(k-1)}^{*},I_k^{*}\right)$。

假设 $k-1$ 时刻的幸存路径 $\boldsymbol{I}_{(k-1)}^{i}$ 上的关于第 $k-j$ (j=0,1,⋯ δ, δ 是判决时延，典型值是 5×L)时刻的符号的概率 $P'\left(I_{k-j}\middle|\boldsymbol{I}_{(k-1)}^{i},\boldsymbol{Y}_{(k-1)}\right)$，则在第 k 个时刻的 θ_k^{*} 状态上的关于第 $k-j$ 时刻符号的概率：

$$
\begin{aligned}
&P'\left(I_{k-j},\theta_k^*,\boldsymbol{Y}_{(k)}\right)\\
&=\sum_{\boldsymbol{I}_{(k-1)}^i\to\theta_k^*}P'\left(I_{k-j},\boldsymbol{I}_{(k-1)}^i,\boldsymbol{Y}_{(k-1)}\right)\cdot P'\left(\theta_k^*,y_k\middle|\boldsymbol{I}_{(k-1)}^i\right)\\
&=\sum_{\boldsymbol{I}_{(k-1)}^i\to\theta_k^*}P'\left(I_{k-j}\middle|\boldsymbol{I}_{(k-1)}^i,\boldsymbol{Y}_{(k-1)}\right)\cdot P'\left(\boldsymbol{I}_{(k-1)}^i,\boldsymbol{Y}_{(k-1)}\right)\\
&\quad\cdot P'\left(\theta_k^*,y_k\middle|\boldsymbol{I}_{(k-1)}^i\right)\\
&=\sum_{\boldsymbol{I}_{(k-1)}^i\to\theta_k^*}P'\left(I_{k-j}\middle|\boldsymbol{I}_{(k-1)}^i,\boldsymbol{Y}_{(k-1)}\right)\cdot P'\left(\boldsymbol{I}_{(k-1)}^i,\boldsymbol{Y}_{(k-1)},\theta_k^*,y_k\right)\\
&=\sum_{\boldsymbol{I}_{(k-1)}^i\to\theta_k^*}P'\left(I_{k-j}\middle|\boldsymbol{I}_{(k-1)}^i,\boldsymbol{Y}_{(k-1)}\right)\cdot P'\left(\boldsymbol{I}_{(k-1)}^i\middle|\theta_k^*,\boldsymbol{Y}_{(k)}\right)\\
&\quad\cdot P'\left(\theta_k^*,\boldsymbol{Y}_{(k)}\right)
\end{aligned}
$$

$$
\begin{aligned}
P'\left(I_{k-j}\middle|\theta_k^*,\boldsymbol{Y}_{(k)}\right)&=\frac{P'\left(I_{k-j},\theta_k^*,\boldsymbol{Y}_{(k)}\right)}{P'\left(\theta_k^*,\boldsymbol{Y}_{(k)}\right)}\\
&=\sum_{\boldsymbol{I}_{(k-1)}^i\to\theta_k^*}P'\left(I_{k-j}\middle|\boldsymbol{I}_{(k-1)}^i,\boldsymbol{Y}_{(k-1)}\right)\cdot P'\left(\boldsymbol{I}_{(k-1)}^i\middle|\theta_k^*,\boldsymbol{Y}_{(k)}\right)
\end{aligned}\tag{14}
$$

式(14)中的 $P'\left(I_{k-j}\middle|\boldsymbol{I}_{(k-1)}^i,\boldsymbol{Y}_{(k-1)}\right)$ 是存储在 $k-1$ 时刻幸存路径上的软信息，而 $P'\left(\boldsymbol{I}_{(k-1)}^i\middle|\theta_k^*\boldsymbol{Y}_{(k)}\right)$ 可以通过式(13)，式(12)和式(3)计算。以上是针对 SOVA 中的 Battail 更新法则的解释，Hagenauer 更新法则进一步将式(14)中的竞争路径 $\boldsymbol{I}_{(k)}^C$ 上的 $P'\left(I_{k-j}\middle|\boldsymbol{I}_{(K-1)}^i,\boldsymbol{Y}_{(K-1)}\right)$ 用最终此状态 θ_k^* 上的幸存路径 $\boldsymbol{I}_k^*$ 上保存的 $P'\left(I_{k-j}\middle|\boldsymbol{I}_{k-1}^*,\boldsymbol{Y}_{k-1}\right)$ 代替。仿真表明 Battail 更新法则输出的软信息更准确。因为 k 时刻所有幸存路径在 $k-\delta$ 时刻之前已经重合，所以 Hagenauer 更新法则不对 $k-\delta$ 时刻之前的符号做更新；Battail 更新法则对不同幸存路径上的 $k-\delta$ 之前的符号可能得到不同的软信息，可以选择全局最优路径上的软信息输出，或者对所有幸存路径上的 $P\left(I_{k-\delta}\middle|\boldsymbol{I}_{(k)}^i,\boldsymbol{Y}_{(k)}\right)$ 用各自的度量 $P\left(\boldsymbol{I}_{(k)}^i\middle|\boldsymbol{Y}_{(k)}\right)$ 加权输出。而经过第 k 时刻的 ACS 过程，找到进入 θ_k^* 状态的幸存路径 $\boldsymbol{I}_{(k)}^*$，则可以将上面更新后的软信息赋给幸存路径：

$$P\left(I_{k-j}\middle|\theta_k^*,\boldsymbol{Y}_{(k)}\right)\to P\left(I_{k-j}\middle|\boldsymbol{I}_{(k)}^*,\boldsymbol{Y}_{(k)}\right)\tag{15}$$

如此则上述过程可以迭代进行下去。以上就是 SOVA 的一种数学解释，因为采用了一些近似，所以实际上 SOVA 无法得到最优软信息，其软信息也很难用数学形式来表达其真实含义。

4 等价于 MAP 的 SOVA

4.1 OSA

如果没有根据式(11)的度量选择幸存路径并消除其他竞争路径，没有式(15)的替换，经过与式(14)一样的推导，可以得到一种等价于MAP的软输出算法[5]。

$$P\left(I_{k-j}\middle|\theta_k^*,\boldsymbol{Y}_{(k)}\right)=\sum_{\theta_{k-1}^*\to\theta_k^*}P\left(I_{k-j}\middle|\theta_{k-1}^i,\boldsymbol{Y}_{(k-1)}\right)\cdot P\left(\theta_{k-1}^i\middle|\theta_k^*,\boldsymbol{Y}_{(k)}\right)$$

$$
\begin{aligned}
P\left(\theta_{k-1}^i\middle|\theta_k^*,\boldsymbol{Y}_{(k)}\right)&=\frac{P\left(\theta_{k-1}^i,\theta_k^*,\boldsymbol{Y}_{(k)}\right)}{P\left(\theta_k^*,\boldsymbol{Y}_{(k)}\right)}\\
&=\frac{P\left(y_k\middle|\theta_k^*,\theta_{k-1}^i\right)\cdot P\left(\theta_k^*\middle|\theta_{k-1}^i\right)\cdot P\left(\theta_{k-1}^i,\boldsymbol{Y}_{(k-1)}\right)}{\sum_j P\left(y_k\middle|\theta_k^*,\theta_{k-1}^j\right)\cdot P\left(\theta_k^*\middle|\theta_{k-1}^j\right)\cdot P\left(\theta_{k-1}^j,\boldsymbol{Y}_{(k-1)}\right)}
\end{aligned}\tag{16}
$$

其中 θ_{k-1}^i 是如式(13)所示的进入 θ_k^* 的竞争路径所对应的 $k-1$ 时刻的状态。上面分子和分母的第 1 项可以根据相应状态转移和接收到的信号直接计算，第 2 项可以从前一级解码器输入或者假设先验等概，第 3 项需要迭代计算。相当于SOVA所计算的状态上的度量改变为 $p\left(\theta_{(k)}^*,Y_k\right)$，而存储在相应状态上的软信息为 $p\left(I_{k-j}\middle|\theta_k^i,Y_{(k)}\right)$。如此则不需要幸存路径的判断，而且需要对$k$时刻之前的所有时刻符号的软信息进行储存和更新，而不能像SOVA一样输出δ个时刻以前的符号软信息。但是与SOVA一样只需要前向的迭代过程。文献[5]还有两点创新，一是不需要存储全部M个符号的概率，因为他们的和为 1；二是 $(k-L)\sim k$ 之间的符号的概率不需要更新，因为它们不加计算地可以根据所处状态等于 1 或者 0，只有当他们退出 θ_k^i 所能表示的范围时才更新。这里如果有判决时延δ的设定，则不是最优软信息，如果对全部以往符号更新，其输出等价于BCJR-MAP算法。有判决延时δ的设定时，对 $k-\delta$ 时刻的软信息采用加权平均的办法输出：

$$p\left(I_{k-\delta},Y_{(k)}\right)=\sum_{\mathrm{O}=1}^{M^L}p\left(I_{k-\delta}\middle|\theta_k^i,Y_{(k)}\right)\cdot p\left(\theta_k^i,Y_{(k)}\right)\tag{17}$$

OSA 适用于 Streaming 方式的数据传输，BCJR-MAP 适用于 Block 方式的数据传输。

4.2 OSA 的一种改进形式

上面的 OSA 算法，是在第 k 时刻，按照进入 θ_k^* 的竞争路径来计算 $P\left(\theta_{k-1}^i\middle|\theta_k^*,\boldsymbol{Y}_{(k)}\right)$ 并更新软信息 $P\left(I_{k-j}\middle|\theta_k^*,\boldsymbol{Y}_{(k)}\right)$ 的，可以有进一步的简化：

$$
\begin{aligned}
P\left(I_{k-j},\theta_k^*,\boldsymbol{Y}_{(k)}\right)&=\sum_{\theta_{k-1}^i\to\theta_k^i}P\left(I_{k-j},\theta_{k-1}^i,\boldsymbol{Y}_{(k-1)}\right)\cdot P\left(\theta_k^*,y_k\middle|\theta_{k-1}^i\right)\\
&=\sum_{\theta_{k-1}^i\to\theta_k^i}P\left(I_{k-j},\theta_{k-1}^i,\boldsymbol{Y}_{(k-1)}\right)\cdot P\left(y_k\middle|\theta_k^*,\theta_{k-1}^i\right)\cdot P\left(\theta_k^*\middle|\theta_{k-1}^i\right)
\end{aligned}\tag{18}
$$

每个状态上的度量仍然为 $P\left(\theta_k^i,\boldsymbol{Y}_{(k)}\right)$，每个状态上的软信息变为 $P\left(I_{k-j},\theta_k^i,\boldsymbol{Y}_k\right)$，并按照式(18)更新软信息，也可以利用文献[5]中给出的两点创新：每个状态上对某个时刻的软信息只需要存储 $M-1$ 个，因为它们的和等于该状态所存储的度量 $P\left(\theta_k^i,\boldsymbol{Y}_{(k)}\right)$；$(k-L)\sim k$之间的符号的概率不需要更新。

这种改进的 OSA 算法不需要采用式(16)来计算 $P\left(\theta_{(k-1)}^i\middle|\theta_k^*,\boldsymbol{Y}_{(k)}\right)$；而且在输出 δ 个时刻以前的符号软信息时，也不需要像 OSA 中的式(17)一样作加权平均，只需要简单的将所有 θ_k^i 状态上的软信息相加：

$$P\left(I_{k-\delta},\boldsymbol{Y}_{(k)}\right)=\sum_{i=1}^{M^L}P\left(I_{k-\delta},\theta_k^i,\boldsymbol{Y}_{(k)}\right)\tag{19}$$

从上面的数学推导可以看出，经过改进的 OSA 算法与原有的 OSA 算法的软信息输出是等价的，而且复杂度有所降低。这两种算法的等价性经过理论上严格的证明，所以不需要进行仿真验证。

5 结束语

本文给出了一种完整的 SOVA 的数学表达，并从这个数学表达出发推导出了两种可以输出最优软信息和具有 SOVA 形式的 SISO 译码算法。第 1 种就是文献中给出的 OSA 算法；第 2 种是对 OSA 的改进。这两种算法的处理过程与 SOVA 相似，只需要一次前向的迭代过程，每前进一个时刻需要对此时刻以前的一定输出时延内的软信息进行更新。第 2 种算法可以得到与 OSA 等价的软输出，并降低了运算复杂度。

参 考 文 献

[1] Hagenauer J, Hoeher P. A Viterbi algorithm with soft-decision outputs and its applications, Global Telecommunications Conference, 1989, and Exhibition. 'Communications Technology for the 1990s and Beyond'. GLOBECOM '89, IEEE, Dallas, Texas, 27-30 Nov., 1989, vol.3: 1680－1686.

[2] Hoeher P. TCM on frequency-selective fading channels: a comparison of soft-output probabilistic equalizers, Global Telecommunications Conference, 1990, and Exhibition. 'Communications: Connecting the Future', GLOBECOM '90, IEEE, San Diego, CA, 2-5 Dec., 1990, vol.1: 376－381.

[3] Berrou C, Adde P, Angui E, Faudeil S. A low complexity soft-output Viterbi decoder architecture. ICC 93. Technical Program, Conference Record, IEEE International Conference on Communications, Geneva, Switzerland, 23-26 May, 1993, Vol.2: 737－740.

[4] Bahl L, Cocke J, Jelinek F, Raviv J. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. on Information Theory*, 1974, 20(2): 284－287.

[5] Li Funxin, Vucetic B, Sato Y. Optimum soft-output detection for channels with intersymbol interference. *IEEE Trans. on Information Theory*, 1995, 41(3): 704－713.

[6] Tan J, Stuber G L. A MAP equivalent SOVA for non-binary turbo codes. IEEE International Conference on Communications 2000, New Orleans, LA, 18-22 June, 2000, vol.2: 602－606.

[7] Bai B, Ma X, Wang X, Novel algorithm for continuous decoding of turbo codes. *IEE Proc-I*, 1999, 146(5): 271－274.

[8] Fossorier M P C, Burkert F, Shu Lin, Hagenauer J. On the equivalence between SOVA and max-log-MAP decodings. *IEEE Communications Letters*, 1998, 2(5): 137－139.

田志刚： 男，1978 年生，博士生，研究方向为多用户检测、均衡技术.

郭文彬： 男，1971 年生，讲师，研究方向为信号估计与检测、Turbo 均衡技术.

杨大成： 男，1951 年生，教授，博士生导师, 研究方向为第三代移动通信系统技术、移动通信网络的规划与优化.