EE/Ma 127c Error-Correcting Codes R. J. McEliece
draft of April 25, 2001 162 Moore

The Forward–Backward Algorithm

Introduction.

In these notes I will present my somewhat idiosyncratic view of the famous *forward-backward algorithm*. The presentation will for now be restricted to a bare description of the simple mathematical underpinnings, with no discussion of the many possible generalizations and applications. In essense, the idea is simply this: If we have a set of N elements of a noncommutative ring, say $x_1, \ldots, x_N$, and are asked to compute the N products of the form

$$X_i = \prod_{\substack{j=1 \\ j \neq i}}^{N} x_j, \qquad \text{for } i = 1, 2, \ldots, N$$

it is a good idea to first compute (recursively) the "forward" and "backward" partial products

$$\alpha_i = x_1 \cdots x_i \qquad \text{for } i = 0, \ldots, N$$
$$\beta_j = x_{j+1} \cdots x_N \qquad \text{for } j = 0, \ldots, N$$

and then use the relationship

$$X_i = \alpha_{i-1}\beta_i \qquad \text{for } i = 1, 2, \ldots, N.$$

it is easy to see that this simple trick reduces the number of multiplications required to compute $X_1, \ldots, X_N$ from $O(N^2)$ to $O(N)$.

It will be good to keep this basic idea in mind as we go through the formalities. We begin in the next section with a desription of the basic problem solved by the FBA, which involves the notion of a *weighted trellis*.

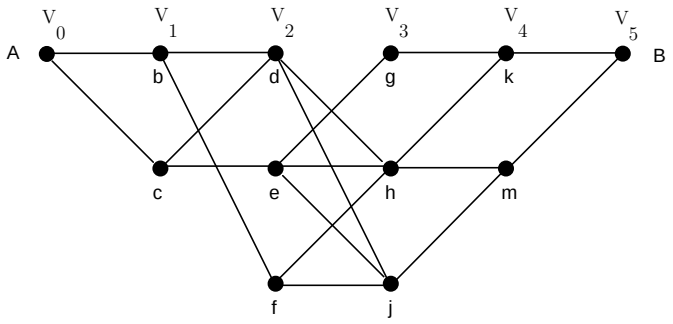

Figure 1. A trellis of rank 5.

1. The Underlying Problems.

A *trellis* is a special kind of directed graph, like the one illustrated in Figure 1. The vertices are sorted according to *rank*, and the set of vertices of rank i is denoted by V_i, for $i = 0, 1, \ldots, N$. The number N is caled the rank of the trellis. The number of vertices of rank i is denoted by q_i. There is a single vertex of rank 0 caled the *source*, and denoted by A, and a single vertex of rank N, called the *sink*, and denoted by B. The only allowed (directed) edges are between vertices whose rank differs by 1, and the set of edges joining vertices of rank $i-1$ to those of rank i is denoted by $E_{i-1,i}$. The initial vertex of an edge e is denoted by $\text{init}(e)$, and the final vertex of e is denoted by $\text{fin}(e)$. For example, In Figure 1 we see a trellis with $N = 5$ and $q_0 = 1$, $q_1 = 2$, $q_2 = 3$, $q_3 = 3$, $q_4 = 2$, and $q_5 = 1$. Also, $V_0 = \{A\}$, $V_1 = \{b, c\}$, $E_{0,1} = \{(A, b), (A, c)\}$, etc.

A *path* P in a trellis is a set of connected edges: $P = e_1 e_2 \ldots e_k$, with $\text{fin}(e_1) = \text{init}(e_2), \ldots, \text{fin}(e_{k-1}) = \text{init}(e_k)$. The number of edges in P is called the *length* of P. If a path $P = e_1 e_2 \cdots e_k$ has initial vertex $u = \text{init}(e_1)$ and final vertex $v = \text{fin}(e_k)$, we write

$$P : u \mapsto v.$$

If in addition the path P passes through the vertex w, we write

$$P : u \overset{w}{\mapsto} v.$$

Finally, if the path P contains the edge e, we write

$$P : u \overset{e}{\mapsto} v.$$

Next, we suppose that each edge e in the trellis is assigned a *weight* $w(e)$, which for now we assume is a real number.

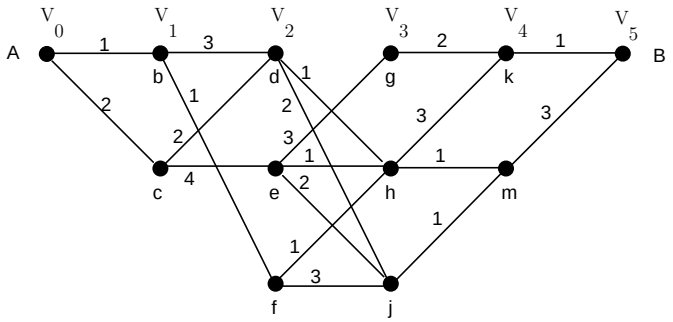

Figure 2. Weights assigned to the trellis of Figure 1.

For example, in Figure 2 we see the trellis of Figure 1 with a weight associated with each edge. If $P = e_1 e_2 \cdots e_k$ is a path of length k in T, its weight is defined as the product of the weights of the component edges:

$$w(P) = w(e_1)w(e_2)\cdots w(e_k).$$

The fundamental quantities computed by the FBA are the *flows*, defined as follows.

Definitions. *If u and v are vertices in a trellis, the flow from u to v, denoted $\mu(u, v)$, is defined as the sum of the weights of all paths from u to v:*

$$\mu(u, v) = \sum_{P: u \mapsto v} w(P).$$

(If there are no such paths, $\mu(u, v)$ is defined to be zero.) Similarly, if u, x, and v are vertices in T, the flow from u to v through x is defined as

$$\mu_x(u, v) = \sum_{P: u \overset{x}{\mapsto} v} w(P).$$

Finally, u and v are vertices and e is an edge, the flow from u to v through e is defined as

$$\mu_e(u, v) = \sum_{P: u \overset{e}{\mapsto} v} w(P).$$

The following simple result is one of the great secrets of the FBA. It shows how to use the humble distributuve law to greatly simplify the calculation of the constrained flows $\mu_x(u, v)$ and $\mu_e(u, v)$.

Theorem 1 . *We have*

$$\mu_x(u,v) = \mu(u,x)\mu(x,v). \tag{1}$$

Similarly, if $\text{init}(e) = x$ *and* $\text{fin}(e) = y$,

$$\mu_e(u,v) = \mu(u,x) \cdot w(e) \cdot \mu(y,v). \tag{2}$$

Proof: We will prove (1), leaving (2) as an exercise.

Suppose there are m paths from u to x, say $P_1, \ldots, P_m$, and n paths from x to v, say $Q_1, \ldots, Q_n$. Then there are exactly mn paths from u to v through x, namely paths of the form $P_i * Q_j$, where "$*$" denotes concatenation. Then we have

$$\begin{aligned}
\mu_x(u,v) &= \sum_{P: u \overset{x}{\mapsto} v} w(P) \\
&= \sum_{i=1}^{m} \sum_{j=1}^{n} w(P_i * Q_j) \\
&= \sum_{i=1}^{m} \sum_{j=1}^{n} w(P_i) w(Q_j) \\
&= \left(\sum_{i=1} w(P_i) \right) \left(\sum_{j=1}^{n} w(Q_j) \right) \quad \text{(The distributive law)} \\
&= \mu(u,x)\mu(x,v). \quad \blacksquare
\end{aligned}$$

The forward-backward algorithm (FBA) addresses the following three problems.

Problem 1. *Compute the flow from A to B, i.e.,*

$$\mu(A,B) = \sum_{P: A \mapsto B} w(P).$$

Since there may be as many as $q_1 q_2 \cdots q_{N-1}$ paths from A to B, the computation of $\mu(A,B)$ appears to be a formidable task. However, as we shall see, the FBA computes this flow using at most $2(q_0 q_1 + q_1 q_2 + \cdots + q_{N-1} q_N)$ arithmetic operations.

Problem 2. *For each vertex v, compute the flow from A to B through v, i.e.,*

$$\mu_v(A,B) = \sum_{P: A \overset{v}{\mapsto} B} w(P).$$

Note that we have

$$\sum_{v \in V_i} \mu_v(A, B) = \mu(A, B),$$

since each path from A to B must pass through exactly one of the vertices in V_i. Thus the ratio $\mu_v(A, B)/\mu(A, B)$ can be interpreted as a kind of probability that a randomly selected path from A to B passes through v.

Problem 3. *For each edge e, compute the flow from A to B through e, i.e.,*

$$\mu_e(A, B) = \sum_{P : A \overset{e}{\mapsto} B} w(P).$$

Note that we have

$$\sum_{e \in E_{i-1,i}} \mu_e(A, B) = \mu(A, B),$$

since each path from A to B must traverse exactly one of the edges in $E_{i-1,i}$. Thus the ratio $\mu_e(A, B)/\mu(A, B)$ can be interpreted as the probability that a randomly selected path from A to B traverses the edge e.

2. The Forward and Backward Recursions.

For each $i = 1, 2, \ldots N$, define a $q_{i-1} \times q_i$ matrix W_i, whose rows are indexed by the vertices in V_{i-1} and columns are indexed by the vertices in V_i:

$$W_i(u, v) = \begin{cases} w(e) & \text{if there is an edge } e \text{ joining } u \text{ to } v \\ 0 & \text{otherwise.} \end{cases}$$

For example, the W_i's associated with the weighted trellis of Figure 2 are as follows:

$$W_1 = \begin{array}{c} \\ A \end{array} \begin{array}{c} \begin{array}{cc} b & c \end{array} \\ \begin{pmatrix} 1 & 2 \end{pmatrix} \end{array}$$

$$W_2 = \begin{array}{c} \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} d & e & f \end{array} \\ \begin{pmatrix} 3 & 0 & 1 \\ 2 & 4 & 0 \end{pmatrix} \end{array}$$

$$W_3 = \begin{array}{c} \\ d \\ e \\ f \end{array} \begin{array}{c} \begin{array}{ccc} g & h & j \end{array} \\ \begin{pmatrix} 0 & 1 & 2 \\ 3 & 1 & 2 \\ 0 & 1 & 3 \end{pmatrix} \end{array}$$

$$W_4 = \begin{array}{c} \\ g \\ h \\ j \end{array} \begin{array}{c} \begin{array}{cc} k & m \end{array} \\ \begin{pmatrix} 2 & 0 \\ 3 & 1 \\ 0 & 1 \end{pmatrix} \end{array}$$

$$W_5 = \begin{array}{c} \\ k \\ m \end{array} \begin{array}{c} B \\ \begin{pmatrix} 1 \\ 3 \end{pmatrix} \end{array}$$

The heart of the forward-backward algorithm is the recursive computation of certain "forward" vectors $\boldsymbol{\alpha}_i$, for $i = 0, 1, \ldots, N$ and "backward" vectors $\boldsymbol{\beta}_i$, for $i = N, N-1, \ldots, 0$. Here $\boldsymbol{\alpha}_i$ is a row vector of dimension q_i, and $\boldsymbol{\beta}_i$ is a column vector of dimension q_i. Both $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$ have components indexed by V_i. If $v \in V_i$, we will denote the vth component of $\boldsymbol{\alpha}_i$ (resp. $\boldsymbol{\beta}_i$) by $\alpha_i(v)$ (resp. $\beta_i(v)$).

The $\boldsymbol{\alpha}_i$'s are defined by the following *forward* recursion:

$$\boldsymbol{\alpha}_0 = 1, \quad \boldsymbol{\alpha}_i = \boldsymbol{\alpha}_{i-1} W_i, \quad \text{for } i = 1, \ldots, N. \tag{3}$$

and the $\boldsymbol{\beta}_i$'s are defined by a corresponding *backward* recursion:

$$\boldsymbol{\beta}_N = 1, \quad \boldsymbol{\beta}_i = W_{i+1} \boldsymbol{\beta}_{i+1}, \quad \text{for } i = N-1, N-2, \ldots, 0. \tag{4}$$

Clearly we have

$$\boldsymbol{\alpha}_i = W_1 W_2 \cdots W_i, \quad \text{for } i = 1, \ldots, N. \tag{5}$$

$$\boldsymbol{\beta}_i = W_{i+1} W_{i+2} \cdots W_N, \quad \text{for } i = N-1, N-2, \ldots, 0. \tag{6}$$

For example, for the weighted trellis shown in Figure 2, we have

$$\begin{aligned}
\boldsymbol{\alpha}_0 &= (\,1\,) \\
\boldsymbol{\alpha}_1 &= \boldsymbol{\alpha}_0 W_1 = (\,1 \quad 2\,) \\
\boldsymbol{\alpha}_2 &= \boldsymbol{\alpha}_1 W_2 = (\,7 \quad 81 \quad 1\,) \\
\boldsymbol{\alpha}_3 &= \boldsymbol{\alpha}_2 W_3 = (\,24 \quad 16 \quad 33\,) \\
\boldsymbol{\alpha}_4 &= \boldsymbol{\alpha}_3 W_4 = (\,96 \quad 49\,) \\
\boldsymbol{\alpha}_5 &= \boldsymbol{\alpha}_4 W_5 = (\,243\,)
\end{aligned}$$

and

$$\begin{aligned}
\boldsymbol{\beta}_5 &= (\,1\,) \\
\boldsymbol{\beta}_4 &= W_5 \boldsymbol{\beta}_5 = (\,1 \quad 3\,)^T \\
\boldsymbol{\beta}_3 &= W_4 \boldsymbol{\beta}_4 = (\,2 \quad 6 \quad 3\,)^T \\
\boldsymbol{\beta}_2 &= W_3 \boldsymbol{\beta}_3 = (\,12 \quad 18 \quad 15\,)^T \\
\boldsymbol{\beta}_1 &= W_2 \boldsymbol{\beta}_2 = (\,51 \quad 96\,)^T \\
\boldsymbol{\beta}_0 &= W_1 \boldsymbol{\beta}_1 = (\,243\,)
\end{aligned}$$

(An alternative definition of the $\boldsymbol{\alpha}_i$s and $\boldsymbol{\beta}_i$s, which is based on geometry rather than algebra:

$$\alpha_i(u) = \sum_{\substack{e \in E_{i-1,i} \\ \mathrm{fin}(e) = u}} \alpha_{i-1}(\mathrm{init}(e)) \cdot w(e).$$

and

$$\beta_i(u) = \sum_{\substack{e \in E_{i,i+1} \\ \mathrm{init}(e) = u}} w(e) \cdot \beta_{i+1}(\mathrm{fin}(e)).$$

These definitions give a direct way to benefit from the sparseness of the trellis,, and can be used to show that the arithmetic complexity of computing the $\boldsymbol{\alpha}_i$'s and $\boldsymbol{\beta}_i$'s is $O(|E|)$, where E is the total number of edges in the trellis.)

The key property of the $\boldsymbol{\alpha}$'s and the $\boldsymbol{\beta}$'s is contained in the following Theorem.

Theorem 2 . *For any $u \in V_i$, we have*

$$\alpha_i(u) = \mu(A, u). \tag{7}$$

Similarly, for any $v \in V_i$, we have

$$\beta_i(v) = \mu(v, B). \tag{8}$$

Proof: We will prove (7); the proof of (8) follows by taking mirror images.

We prove (7) by induction on i. The case $i = 1$ says that

$$\alpha_1(u) = \mu(A, u).$$

But $\boldsymbol{\alpha}_1 = W_1$ by (3), and $W_1(A, u)$ is by definition the weight of the edge (if any) between A and u. So (7) is correct for $i = 1$.

Assuming the truth of (7) for i, we proceed as follows. Let $u \in V_{i+1}$. Then

$$\begin{aligned}
\mu(A, u) &= \sum_{x \in V_i} \mu_x(A, u) \quad \text{(conservation of flow)} \\
&= \sum_{x \in V_i} \mu(A, x)\mu(x, u) \qquad \text{(Lemma 1)} \\
&= \sum_{x \in V_i} \alpha_i(x) W_{i+1}(x, u) \qquad \text{(induction hypothesis and definiton of } W_{i+1}) \\
&= \alpha_{i+1}(u) \quad \text{(definition of } \boldsymbol{\alpha}_{i+1}) \quad \blacksquare
\end{aligned}$$

3. The Solutions.

Theorem 3 . *(Solution to Problem 1.) For any $i = 0, 1, \ldots, N$, we have*

$$\mu(A, B) = \boldsymbol{\alpha}_i \boldsymbol{\beta}_i.$$

In particular, $\mu(A, B) = \boldsymbol{\alpha}_N = \boldsymbol{\beta}_0$.

Thus for example,

$$\mu(A, B) = \boldsymbol{\beta}_0 = \boldsymbol{\alpha}_0\boldsymbol{\beta}_0 = \boldsymbol{\alpha}_1\boldsymbol{\beta}_0 = \boldsymbol{\alpha}_2\boldsymbol{\beta}_2 = \boldsymbol{\alpha}_3\boldsymbol{\beta}_3 = \boldsymbol{\alpha}_4\boldsymbol{\beta}_4 = \boldsymbol{\alpha}_5\boldsymbol{\beta}_5 = \boldsymbol{\alpha}_5 = 243.$$

In short, in Figure 2, the flow from A to B is 243.

Proof: For any index i we have

$$\begin{aligned}
\mu(A, B) &= \sum_{x \in V_i} \mu_x(A, B) \quad \text{(conservation of flow)} \\
&= \sum_{x \in V_i} \mu(A, x)\mu(x, B) \quad \text{(Lemma 1)} \\
&\sum_{x \in V_i} \alpha_i(x)\beta_i(x) \quad \text{(Theorem 2)} \\
&= \boldsymbol{\alpha}_i \boldsymbol{\beta}_i. \qquad \blacksquare
\end{aligned}$$

Corollary. *We have*

$$\mu(A, B) = W_1 W_2 \cdots W_N.$$

Thus computing flows is equivalent to matrix multiplication!

Proof: We have from (5) and (6):

$$\boldsymbol{\alpha}_i = W_1 \cdots W_i, \quad \boldsymbol{\beta}_i = W_{i+1} \cdots W_N,$$

and hence $\boldsymbol{\alpha}_i \boldsymbol{\beta}_i = W_1 W_2 \cdots W_n$. The result now follows from Theorem 3. $\blacksquare$

Theorem 4 . *(Solution to Problem 2.) If $v \in V_i$, we have*

$$\mu_v(A, B) = \alpha_i(v)\beta_i(v).$$

Thus if we use the notation $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = (x_1 y_1, \ldots, x_n y_n)$, the vertex-constrained flows through the vertices in V_i are the cmponents of the vector $\langle \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i \rangle$. In the weighted trellis of Figure 2, for example, we have

$$
\begin{aligned}
\langle \boldsymbol{\alpha}_0, \boldsymbol{\beta}_0 \rangle &= \begin{matrix} A \\ \begin{pmatrix} 243 \end{pmatrix} \end{matrix} \\
\langle \boldsymbol{\alpha}_1, \boldsymbol{\beta}_1 \rangle &= \begin{matrix} b & c \\ \begin{pmatrix} 51 & 192 \end{pmatrix} \end{matrix} \\
\langle \boldsymbol{\alpha}_2, \boldsymbol{\beta}_2 \rangle &= \begin{matrix} d & e & f \\ \begin{pmatrix} 84 & 144 & 15 \end{pmatrix} \end{matrix} \\
\langle \boldsymbol{\alpha}_3, \boldsymbol{\beta}_3 \rangle &= \begin{matrix} g & h & j \\ \begin{pmatrix} 48 & 96 & 99 \end{pmatrix} \end{matrix} \\
\langle \boldsymbol{\alpha}_4, \boldsymbol{\beta}_4 \rangle &= \begin{matrix} k & m \\ \begin{pmatrix} 96 & 147 \end{pmatrix} \end{matrix} \\
\langle \boldsymbol{\alpha}_5, \boldsymbol{\beta}_5 \rangle &= \begin{matrix} B \\ \begin{pmatrix} 243 \end{pmatrix} \end{matrix}
\end{aligned}
$$

For example, the flow from A to B through vertex h is 96.

Proof: We have

$$
\begin{aligned}
\mu_v(A, B) &= \mu(A, v)\mu(v, B) \quad \text{(Lemma 1)} \\
&= \alpha_i(v)\beta_i(v) \quad \text{(Theorem 2)}
\end{aligned}
$$

∎

Theorem 5 . *(Solution to Problem 3.) If* $\text{init}(e) = u \in V_{i-1}$ *and* $\text{fin}(e) = v \in V_i$*, we have*

$$\mu_e(A, B) = \alpha_{i-1}(u) \cdot w(e) \cdot \beta_i(v).$$

Thus if we use the notation $\langle \boldsymbol{\alpha} W \boldsymbol{\beta} \rangle_{(u,v)} = \alpha(u) W(u, v) \beta(v)$, the edge-constrained flows through the edges in $E_{i-1,i}$ are given by the entries in the matrix $\langle \boldsymbol{\alpha}_{i-1} W_i \boldsymbol{\beta}_i \rangle$. In the

trellis of Figure 2, for example, we have

$$\langle \boldsymbol{\alpha}_0 W_1 \boldsymbol{\beta}_1 \rangle = \begin{array}{c} \\ A \end{array} \begin{array}{c} \begin{array}{cc} b & c \end{array} \\ \begin{pmatrix} 51 & 192 \end{pmatrix} \end{array}$$

$$\langle \boldsymbol{\alpha}_1 W_2 \boldsymbol{\beta}_2 \rangle = \begin{array}{c} \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} d & e & f \end{array} \\ \begin{pmatrix} 36 & 0 & 15 \\ 48 & 144 & 0 \end{pmatrix} \end{array}$$

$$\langle \boldsymbol{\alpha}_2 W_3 \boldsymbol{\beta}_3 \rangle = \begin{array}{c} \\ d \\ e \\ f \end{array} \begin{array}{c} \begin{array}{ccc} g & h & j \end{array} \\ \begin{pmatrix} 0 & 42 & 42 \\ 48 & 48 & 48 \\ 0 & 6 & 9 \end{pmatrix} \end{array}$$

$$\langle \boldsymbol{\alpha}_3 W_4 \boldsymbol{\beta}_4 \rangle = \begin{array}{c} \\ g \\ h \\ j \end{array} \begin{array}{c} \begin{array}{cc} k & m \end{array} \\ \begin{pmatrix} 48 & 0 \\ 48 & 48 \\ 0 & 99 \end{pmatrix} \end{array}$$

$$\langle \boldsymbol{\alpha}_4 W_5 \boldsymbol{\beta}_5 \rangle = \begin{array}{c} \\ k \\ m \end{array} \begin{array}{c} \begin{array}{c} B \end{array} \\ \begin{pmatrix} 96 \\ 147 \end{pmatrix} \end{array}$$

For example, the flow from A to B through the edge $e \to h$ is 48.

Proof: We have

$$\begin{aligned} \mu_e(A, B) &= \mu(A, u) \cdot w(e) \cdot \mu(v, B) \quad \text{(Lemma 1)} \\ &= \alpha_{i-1}(u) \cdot w(e) \cdot \beta_i(v). \quad \text{(Theorem 2)} \end{aligned}$$

∎

References.

1. L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Inform. Theory*, vol. IT-20 (1974), pp. 284–287.