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 essence, 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

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

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

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

and then use the relationship

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

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 description of the basic problem solved by the FBA, which involves the notion of a weighted trellis.
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 called 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 called 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 init$(e)$, and the final vertex of $e$ is denoted by 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.

fig:Trellis, Trellis2.pdf scaled 500, A trellis of rank 5.

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

$$T : u \mapsto v.$$ 

If in addition the track $T$ passes through the vertex $w$, we write

$$T : u \xrightarrow{w} v.$$ 

Finally, if the track $T$ contains the edge $e$, we write

$$T : u \xrightarrow{e} v.$$ 

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 $T = e_1e_2 \cdots e_k$ is a track of length $k$ in $T$, its weight is defined as the product of the weights of the component edges:

$$\gamma(T) = w(e_1)w(e_2) \cdots w(e_k).$$

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 tracks from $u$ to $v$:

$$\mu(u,v) = \sum_{T:u\rightarrow v} w(T).$$

(If there are no such tracks, $\mu(u,v)$ 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

$$\mu_x(u,v) = \sum_{T:u\xrightarrow{x} v} w(T).$$

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

$$\mu_e(u,v) = \sum_{T:u\xrightarrow{e} v} w(T).$$

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

**Theorem 1.** We have

$$\mu_x(u,v) = \mu(u,x)\mu(x,v).$$

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

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

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

Suppose there are $m$ tracks from $u$ to $x$, say $T_1, \ldots, T_m$, and $n$ tracks from $x$ to $v$, say $Q_1, \ldots, Q_n$. Then there are exactly $mn$ tracks from $u$ to $v$ through $x$, namely tracks of
the form $T_i * Q_j$, where “$*$” denotes concatenation. Then we have

$$
\mu_x(u,v) = \sum_{T:u \rightarrow v} w(T)
$$

$$
= \sum_{i=1}^{m} \sum_{j=1}^{n} w(T_i * Q_j)
$$

$$
= \sum_{i=1}^{m} \sum_{j=1}^{n} w(T_i)w(Q_j)
$$

$$
= \left( \sum_{i=1}^{m} w(T_i) \right) \left( \sum_{j=1}^{n} w(Q_j) \right) \quad \text{(The distributive law)}
$$

$$
= \mu(u,x)\mu(x,v).
$$

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

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

$$
\mu(A,B) = \sum_{T:A \rightarrow B} w(T).
$$

Since there may be as many as $q_1q_2 \cdots q_{N-1}$ tracks 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 $2(q_0q_1 + q_1q_2 + \cdots + q_{N-1}q_N)$ arithmetic operations.

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

$$
\mu_v(A,B) = \sum_{T:A \rightarrow B} w(T).
$$

Note that we have

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

since each track 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 track from $A$ to $B$ passes through $v$.

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

$$
\mu_e(A,B) = \sum_{T:A \rightarrow B} w(T).
$$
Note that we have

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

since each track 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 track 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:

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

$$W_2 = b \begin{pmatrix} 3 & 0 & 1 \\ 2 & 4 & 0 \end{pmatrix}$$

$$W_3 = e \begin{pmatrix} 3 & 1 & 2 \\ 0 & 1 & 3 \end{pmatrix}$$

$$W_4 = h \begin{pmatrix} 3 & 1 \\ 0 & 1 \end{pmatrix}$$

$$W_5 = k \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$

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

The \(\alpha_i\)'s are defined by the following forward recursion:
\[
(3) \quad \alpha_0 = 1, \quad \alpha_i = \alpha_{i-1}W_i, \quad \text{for } i = 1, \ldots, N.
\]
and the \(\beta_i\)'s are defined by a corresponding backward recursion:
\[
(4) \quad \beta_N = 1, \quad \beta_i = W_{i+1}\beta_{i+1}, \quad \text{for } i = N-1, N-2, \ldots, 0.
\]

Clearly we have
\[
(5) \quad \alpha_i = W_1W_2\cdots W_i, \quad \text{for } i = 1, \ldots, N.
\]
\[
(6) \quad \beta_i = W_{i+1}W_{i+2}\cdots W_N, \quad \text{for } i = N-1, N-2, \ldots, 0.
\]

For example, for the weighted trellis shown in Figure 2, we have
\[
\alpha_0 = (1) \quad \alpha_1 = \alpha_0W_1 = (1 \quad 2) \quad \alpha_2 = \alpha_1W_2 = (7 \quad 81 \quad 1) \quad \alpha_3 = \alpha_2W_3 = (24 \quad 16 \quad 33) \quad \alpha_4 = \alpha_3W_4 = (96 \quad 49) \quad \alpha_5 = \alpha_4W_5 = (243)
\]

and
\[
\beta_5 = (1) \quad \beta_4 = W_5\beta_5 = (1 \quad 3)^T \quad \beta_3 = W_4\beta_4 = (2 \quad 6 \quad 3)^T \quad \beta_2 = W_3\beta_3 = (12 \quad 18 \quad 15)^T \quad \beta_1 = W_2\beta_2 = (51 \quad 96)^T \quad \beta_0 = W_1\beta_1 = (243)
\]

(An alternative definition of the \(\alpha_i\)s and \(\beta_i\)s, which is based on geometry rather than algebra:
\[
\alpha_i(u) = \sum_{e \in E_{i-1}, \text{fin}(e) = u} \alpha_{i-1}(\text{init}(e)) \cdot w(e).
\]

and
\[
\beta_i(u) = \sum_{e \in E_{i+1}, \text{init}(e) = u} w(e) \cdot \beta_{i+1}(\text{fin}(e)).
\]

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 \(\alpha_i\)s and \(\beta_i\)s is \(O(|E|)\), where \(E\) is the total number of edges in the trellis.)

The key property of the \(\alpha\)'s and the \(\beta\)'s is contained in the following Theorem.
**Theorem 2.** For any \( u \in V_i \), we have

\[
\alpha_i(u) = \mu(A, u).
\]

Similarly, for any \( v \in V_i \), we have

\[
\beta_i(v) = \mu(v, B).
\]

**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 \( \alpha_1 = W_1 \) 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

\[
\mu(A, u) = \sum_{x \in V_i} \mu_x(A, u) \quad \text{(conservation of flow)}
\]

\[
= \sum_{x \in V_i} \mu(A, x)\mu(x, u) \quad \text{(Lemma 1)}
\]

\[
= \sum_{x \in V_i} \alpha_i(x)W_{i+1}(x, u) \quad \text{(induction hypothesis and definition of } W_{i+1})
\]

\[= \alpha_{i+1}(u) \quad \text{(definition of } \alpha_{i+1}) \]

\[\blacksquare\]
3. The Solutions.

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

\[
\mu(A, B) = \alpha_i \beta_i.
\]

In particular, \( \mu(A, B) = \alpha_N = \beta_0 \).

Thus for example,

\[
\mu(A, B) = \beta_0 = \alpha_0 \beta_0 = \alpha_1 \beta_0 = \alpha_2 \beta_2 = \alpha_3 \beta_2 = \alpha_4 \beta_4 = \alpha_5 \beta_5 = \alpha_5 = 243.
\]

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

Proof: For any index \( i \) we have

\[
\mu(A, B) = \sum_{x \in V_i} \mu_x(A, B) \quad \text{(conservation of flow)}
\]

\[
= \sum_{x \in V_i} \mu(A, x) \mu(x, B) \quad \text{(Lemma 1)}
\]

\[
= \sum_{x \in V_i} \alpha_i(x) \beta_i(x) \quad \text{(Theorem 2)}
\]

\[
= \alpha_i \beta_i. \quad \blacksquare
\]

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):

\[
\alpha_i = W_1 \cdots W_i, \quad \beta_i = W_{i+1} \cdots W_N,
\]

and hence \( \alpha_i \beta_i = W_1 W_2 \cdots W_n \). 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 $\langle x, y \rangle = (x_1y_1, \ldots, x_ny_n)$, the vertex-constrained flows through the vertices in $V_i$ are the components of the vector $\langle \alpha, \beta \rangle$. In the weighted trellis of Figure 2, for example, we have

$$
\langle \alpha_0, \beta_0 \rangle = \begin{pmatrix} A \\ b \\ c \\ \alpha_1, \beta_1 \rangle = \begin{pmatrix} 51 \\ 192 \\ d \\ e \\ f \\ \alpha_2, \beta_2 \rangle = \begin{pmatrix} 84 \\ 144 \\ 15 \\ g \\ h \\ j \\ \alpha_3, \beta_3 \rangle = \begin{pmatrix} 48 \\ 96 \\ 99 \\ k \\ m \\ \alpha_4, \beta_4 \rangle = \begin{pmatrix} 96 \\ 147 \\ B \\ \alpha_5, \beta_5 \rangle = \begin{pmatrix} 243 
\end{pmatrix}
$$

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

**Proof:** We have

$\mu_v(A, B) = \mu(A, v)\mu(v, B)$ \hspace{1cm} (Lemma 1)
$\hspace{1cm} = \alpha_i(v)\beta_i(v)$ \hspace{1cm} (Theorem 2)

**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 $\langle \alpha W \beta \rangle_{(u, v)} = \alpha(u)W(u, v)\beta(v)$, the edge-constrained flows through the edges in $E_{i-1,i}$ are given by the entries in the matrix $\langle \alpha_{i-1}W_i\beta_i \rangle$. In the
trellis of Figure 2, for example, we have

\[
\begin{align*}
\langle \alpha_0 W_1 \beta_1 \rangle &= A \begin{pmatrix} b & c \\ d & e & f \end{pmatrix} \\
\langle \alpha_1 W_2 \beta_2 \rangle &= b \begin{pmatrix} 36 & 0 & 15 \\ 48 & 144 & 0 \end{pmatrix} \\
\langle \alpha_2 W_3 \beta_3 \rangle &= e \begin{pmatrix} 48 & 48 & 48 \\ 0 & 6 & 9 \end{pmatrix} \\
\langle \alpha_3 W_4 \beta_4 \rangle &= f \begin{pmatrix} 48 & 0 \\ 48 & 48 \end{pmatrix} \\
\langle \alpha_4 W_5 \beta_5 \rangle &= g \begin{pmatrix} 96 \\ 147 \end{pmatrix}
\end{align*}
\]

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

**Proof:** We have

\[
\mu_e(A, B) = \mu(A, u) \cdot w(e) \cdot \mu(v, B) \quad \text{(Lemma 1)}
\]
\[
= \alpha_{i-1}(u) \cdot w(e) \cdot \beta_i(v). \quad \text{(Theorem 2)}
\]

Look again at Figure 1, and imagine a particle starting at point $A$ and following edges at random until it reaches point $B$. What is the probability that the particle passed through vertex $h$? Or edge $bf$? Of course we cannot answer these questions unless we know a bit more about how the particle selected its track. Let us suppose therefore that the particle was Markovian. By this we mean that each edge in the trellis is labelled with a probability $\pi(e)$, and that when the particle arrives at vertex $v$ it selects its next edge according to the probabilities $\pi(e)$, for those edges with init $e = v$. Naturally we require

$$\sum_{e: \text{init } e = v} \pi(e) = 1 \quad \text{for all } v.$$ 

Then if $T = e_1 e_2 \cdots e_n$ is a track from $A$ to $B$, the probability that the particles followed $T$ is

$$\pi(T) = \pi(e_1) \pi(e_2) \cdots \pi(e_n).$$

And now we can answer questions of the form “What is the probability that the particle passed through a given vertex or edge:

$$\Pr\{\text{Particle passed through } v\} = \sum_{T: A \xleftarrow{} v} \pi(T)$$

$$\Pr\{\text{Particle passed through } e\} = \sum_{T: A \xleftarrow{} B} \pi(T).$$

These probabilities are apparently suited for the FBA.

For example, consider the following sets of “Markov edge labels,” for the trellis of Figure 2:

---

**Figure 3.** A Markov trellis.
When we compute the $\alpha_i$, things seem to go well:

$$\alpha_0 = A \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\alpha_1 = \Pi_1 = A \begin{pmatrix} 1/3 & 2/3 \end{pmatrix}$$

$$\alpha_2 = \Pi_1 \Pi_2 = A \begin{pmatrix} 17/36 & 16/36 & 3/36 \end{pmatrix}$$

$$\alpha_3 = \Pi_1 \Pi_2 \Pi_3 = A \begin{pmatrix} 96/432 & 109/432 & 227/432 \end{pmatrix}$$

$$\alpha_4 = \Pi_1 \Pi_2 \Pi_3 \Pi_4 = A \begin{pmatrix} 79/192 & 113/192 \end{pmatrix}$$

$$\alpha_5 = \Pi_1 \Pi_2 \Pi_3 \Pi_4 \Pi_5 = A \begin{pmatrix} 1 \end{pmatrix}$$
But when we compute the $\beta$, we get a surprise:

$$\begin{align*}
\beta_1 &= \Pi_1 \Pi_2 \Pi_3 \Pi_4 \Pi_5 = A \begin{pmatrix} B \\ B \end{pmatrix} \\
\beta_2 &= \Pi_2 \Pi_3 \Pi_4 \Pi_5 = b \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
\beta_3 &= \Pi_3 \Pi_4 \Pi_5 = e \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
\beta_4 &= \Pi_4 \Pi_5 = h \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
\beta_5 &= \Pi_5 = k \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
\beta_6 &= B \begin{pmatrix} 1 \end{pmatrix}
\end{align*}$$

The explanation is . . . .

Suppose when our markovian particle executes a random trajectory

\[ T = E_1, E_2, \ldots, E_N, \]

where init \( E_1 = A, \) fin \( E_i = \text{init} E_{i+1}, \) for \( i = 1, \ldots, N - 1, \) and fin \( E_N = B, \) it produces “evidence”

\[ Y = Y_1, Y_2, \ldots, Y_N, \]

where \( Y_i \) is a random variable which is a “noisy” version of the edge \( E_i \in E_{i-1,i}. \) The evidence is stochastically described by a series of transition probability matrices \( p_i(y|e), i = 1, \ldots, N: \)

\[ \Pr\{Y_i = y|E_i = e\} = p_i(y|e). \]

Suppose the vector \( Y_1, \ldots, Y_n \) is observed to be \( (y_1^*, \ldots, y_N^*) \). We call \( (y_1^*, \ldots, y_N^*) \) the “evidence,” denoted \( E. \) Now by Bayes’s rule,

\[ \Pr\{T|E\} = \frac{\Pr\{T\} \Pr\{E|T\}}{\Pr\{E\}}, \]

\[ \Pr\{v \in T|E\} = \sum_{T:A \rightarrow B} \Pr\{T|E\} \propto \sum_{T:A \rightarrow B} \Pr\{T\} \Pr\{E|T\} \]

\[ \Pr\{e \in T|E\} = \sum_{T:A \rightarrow B} \Pr\{T|E\} \propto \sum_{T:A \rightarrow B} \Pr\{T\} \Pr\{E|T\}. \]

Now

\[ \Pr\{T\} = \prod_{i=1}^{N} \pi(e_i) \quad \text{(prior)} \]

\[ \Pr\{E|T\} = \prod_{i=1}^{N} p_i(y_i^*|e_i). \quad \text{(likelihood)} \]

So if we define the edge weights for \( e \in E_{i-1,i} \) as

\[ \gamma(e) = \pi(e)p_i(y_i^*|e) \]

\[ \gamma(T) = \prod_{i=1}^{N} \gamma(e_i). \]

we have a job for the FBA!

\[ \Pr\{v \in T|E\} \propto \sum_{T:A \rightarrow B} \gamma(T) = \alpha(v)\beta(v) \]

\[ \Pr\{e \in T|E\} \propto \sum_{T:A \rightarrow B} \gamma(T) = \alpha(\text{init } e)\gamma(e)\beta(\text{fin } e). \]
Example: Let the edge transition table be as follows: (In this example \( p(y|e) \) depends only on \( \text{init}_e \).)

\[
\begin{array}{c|cccc}
  & 0 & 1 & 2 & ? \\
\hline
0* & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \\
1* & \frac{1}{8} & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} \\
2* & \frac{1}{4} & \frac{1}{8} & \frac{1}{2} & \frac{1}{8} \\
\end{array}
\]

\[
\Pi = 1 \begin{pmatrix}
0 & \frac{1}{2} & \frac{1}{2} \\
2 & \frac{1}{2} & \frac{1}{2}
\end{pmatrix}
\]

\[
\Gamma[0] = 1 \begin{pmatrix}
0 & \frac{1}{4} & \frac{1}{4} \\
2 & \frac{1}{8} & \frac{1}{8}
\end{pmatrix}
\]

\[
\Gamma[1] = 1 \begin{pmatrix}
0 & \frac{1}{8} & \frac{1}{8} \\
2 & \frac{1}{16} & \frac{1}{16}
\end{pmatrix}
\]

\[
\Gamma[2] = 1 \begin{pmatrix}
0 & \frac{1}{16} & \frac{1}{16} \\
2 & \frac{1}{8} & \frac{1}{4}
\end{pmatrix}
\]

\[
\Gamma[?] = 1 \begin{pmatrix}
0 & \frac{1}{16} & \frac{1}{16} \\
2 & \frac{1}{16} & \frac{1}{16}
\end{pmatrix}
\]

Alternatively, \( p(y|e) \) may depend only on \( \text{fin}_e \):

\[
\begin{array}{c|cccc}
  & 0 & 1 & 2 & ? \\
\hline
*0 & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \\
*1 & \frac{1}{8} & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} \\
*2 & \frac{1}{4} & \frac{1}{8} & \frac{1}{2} & \frac{1}{8} \\
\end{array}
\]
References.