Longest common subsequences and the Bernoulli matching model: numerical work and analyses of the r-reach simplification

Jonah Blasiak

February 1, 2008

Abstract

The expected length of longest common subsequences is a problem that has been in the literature for at least twenty five years. Determining the limiting constants γ_k appears to be quite difficult, and the current best bounds leave much room for improvement. Boutet de Monvel explores an independent version of the problem he calls the Bernoulli Matching model. He explores this problem and its relation to the longest common subsequence problem. This paper continues this pursuit by focusing on a simplification we term r-reach. For the string model, $\mathbf{L}_r(u, v)$ is the longest common subsequence of u and v given that each matched pair of letters is no more than r letters apart.

1 Introduction

In our technology oriented society fast processing of digital data is becoming increasingly important. String comparison is a kind of data processing that has applications in a wide range of fields including molecular biology, human speech recognition, computer spelling correction, and gas chromatography [4]. A robust, extensively studied, method for comparing two strings, u and v say, is to compute the length of one of their longest common subsequences (denote this length by $\mathbf{L}(u, v)$). A subsequence of a string u is a string obtained by deleting some elements of u. For example, *netra* is a subsequence of *cinematography*. A longest common subsequence of two strings u and v is a subsequence of u and v of maximum length. For example, *netra* is an longest common subsequence of *cinematography* and *neurotransmitter* because there is no longer string that is a subsequence of both strings.

1.1 The Random String model

The following notation will be useful for working with strings:

k	lower bound	approximation	upper bound	k	upper bound	approximation	lower bound
2	.77391	.8123	.83763	9	.40321	.4936	.55394
3	.63376	.7176	.76581	10	.38656	.4747	.53486
4	.55282	.6544	.70824	11	.37196	.4580	.51785
5	.50952	.6075	.66443	12	.35899	.4432	.50260
6	.47169	.5707	.62932	13	.34737	.4297	.48880
7	.44502	.5405	.60019	14	.33687	.4176	.47620
8	.42237	.5152	.57541	15	.32732	.4066	.46462

Figure 1: Current best bounds and Monte Carlo approximations of γ_k. Lower bounds are from [3] and [1]. Upper bounds are from [3]. Approximations are from [2] and were computed using Monte Carlo simulations extrapolated to large n using $\frac{\mathbf{EL}n}{n} = \gamma_k + \frac{A_k}{\sqrt{n}\ln n} + \frac{C_k}{n \ln n}$, for real numbers A_k, C_k that don't depend on n.

Definition. Define an alphabet Σ of size k to be $\{0, 1, \ldots, k-1\}$. Let Σ^n be the set of all sequences of length n on alphabet Σ.

Definition. If $u = u_1 u_2 \ldots u_n$ and $u_i \in \Sigma$, define $u(i \ldots j)$ to be the substring $u_i u_{i+1} \ldots u_j$.

A very interesting and difficult problem is to compute the average length of longest common subsequences over all possible pairs of strings. Or more precisely, define

$$\mathbf{EL}_n^{(k)} = \frac{1}{k^{2n}} \sum_{u,v \in \Sigma^n} \mathbf{L}(u, v)$$

An open problem is to compute the following limit:

$$\gamma_k = \lim_{n \to \infty} \frac{\mathbf{EL}_n^{(k)}}{n}$$

Klarner and Rivest established that $\mathbf{EL}_n$ is superadditive–$\mathbf{EL}_{n+m} \geq \mathbf{EL}_n + \mathbf{EL}_m$– and from this it can be shown that the above limit exists (see e.g., [1]).

The current best lower and upper bounds as well as Monte Carlo approximations of γ_k are shown in Figure (1).

Longest common subsequence computations can also be formulated as a dynamic programming algorithm or as a directed time passage percolation model (see e.g. [3],[2]). In the directed time passage percolation model, we work with the two dimensional lattice in the first quadrant: vertices exist at the points (i, j) for $i, j \in \{0, 1, 2, ...\}$. On each vertex (i, j) $\mathbf{D}_{i,j}$ will is an integer, and $\mathbf{D}_{i,0}$ and $\mathbf{D}_{0,i}$ are initialized to 0. Given two strings u and v, $\mathbf{L}(u, v)$ is computed by preserving $\mathbf{D}_{i,j} = \mathbf{L}(u(1 \ldots i), v(1 \ldots j))$. The necessary recurrence is

$$\mathbf{D}_{i,j} = \begin{cases} \mathbf{D}_{i-1,j-1} + 1 & \text{if } \delta_{u(i),v(j)} = 1 \\ max\{\mathbf{D}_{i,j-1}, \mathbf{D}_{i-1,j}\} & \text{if } \delta_{u(i),v(j)} = 0 \end{cases}$$

Where $\delta_{u(i),v(j)}$ is the Kronecker delta (the motivations for this notation will become clear in the next section). Another way of looking at this recurrence is

to make bonds between adjacent vertices in the lattice directed in the positive x and y directions. A diagonal bond from $(i-1, j-1)$ to (i, j) is added if and only if $\delta_{u(i),v(j)} = 1$. If the horizontal and vertical bonds are given weight 0, and the diagonal bonds are given weight 1, $\mathbf{L}(u, v)$ is the weight of a maximum weight path from $(0, 0)$ to $(|u|, |v|)$.

1.2 The Bernoulli Matching model

A related problem called the Bernoulli Matching model is named and well explored by Boutet de Monvel in [2]. It is most readily seen as a modification of the directed time passage percolation model. Instead of placing diagonal bonds based on a match in a pair of strings, diagonal bonds are placed independently at each location with probability $1/k$. In the random string model, the probability of a bond between $(i-1, j-1)$ and (i, j) is $1/k$, but these probabilities are not independent. The recurrence for the Bernoulli Matching model is

$$\mathbf{D}_{i,j} = \begin{cases} \mathbf{D}_{i-1,j-1} + 1 & \text{if } \epsilon_{ij} = 1 \\ max\{\mathbf{D}_{i,j-1}, \mathbf{D}_{i-1,j}\} & \text{if } \epsilon_{ij} = 0 \end{cases}$$

where the ϵ_{ij} are independent random variables with $\Pr(\epsilon_{ij} = 1) = 1/k$ and $\Pr(\epsilon_{ij} = 0) = 1 - 1/k$. Let $\mathbf{EL}_n^{B(k)}$ be the expected value of $\mathbf{D}_{n,n}$ given this model. $\mathbf{EL}_n^{B(k)}$, like $\mathbf{EL}_n^{(k)}$, is superadditive [2] and therefore the following limit exists:

$$\gamma_k^B = \lim_{n\to\infty} \frac{\mathbf{EL}_n^{B(k)}}{n}$$

Boutet de Monvel [2] has conjectured that $\gamma_k^B = \frac{2}{1+\sqrt{k}}$ and gives a more general conjecture for the off diagonal lattice positions (Steele conjectured this for the Random String model in 1982, Boutet de Monvel refined it in 1999). He also presents a nice derivation of this result based on cavity methods typically used for the mean field theory of disordered systems, which he does not try to justify rigorously. Though not yet a proof, the method appears to solve the problem quite elegantly and agrees well with numerical approximations.

1.3 The r-reach simplification

A straight-forward way of obtaining a lower bound for $\mathbf{EL}_n^{(k)}$ is to only consider common subsequences that do not match letters "too far" from each other. This is equivalent to restricting the lattice to a diagonal band of fixed width with center line $x = y$. More precisely, let $\mathbf{L}_r(u, v)$ be the length of a common subsequence of u and v as long as possible given that if $u(i) = v(j)$ are paired by the subsequence, then $|i - j| \leq r$. We will use $\mathbf{R}$ instead of $\mathbf{D}$ when working with r-reach. The recurrence is modified as follows ($\mathbf{R}_{i,0}$, and $\mathbf{R}_{0,i}$ are initialized to 0 as before):

$$\mathbf{R}_{i,j} = \begin{cases} \mathbf{R}_{i-1,j-1} + 1 & \text{if } \delta_{u(i),v(j)}, (\epsilon_{ij}) = 1 & \\ \max\{\mathbf{R}_{i,j-1}, \mathbf{R}_{i-1,j}\} & \text{if } \delta_{u(i),v(j)}, (\epsilon_{ij}) = 0 & \text{and } |i-j| < r \\ \mathbf{R}_{i,j-1} & \text{if } \delta_{u(i),v(j)}, (\epsilon_{ij}) = 0 & \text{and } j - i \geq r \\ \mathbf{R}_{i-1,j} & \text{if } \delta_{u(i),v(j)}, (\epsilon_{ij}) = 0 & \text{and } i - j \geq r \end{cases}$$

Let $\mathbf{EL}_{n,k,r}$, $\left(\mathbf{EL}^B_{n,k,r}\right)$ be the expected value of $\mathbf{R}_{n,n}$ given this model. Superadditivity still holds in this model (i.e. $\mathbf{EL}_{n,k,r} + \mathbf{EL}_{m,k,r} \leq \mathbf{EL}_{(n+m),k,r}$) because a maximum weight path from $(0,0)$ to $(n+m, n+m)$ has weight at least as large as (weight of maximum weight path from $(0,0)$ to (n,n))+(weight of maximum weight path from (n,n) to $(n+m, n+m)$). The same argument applies to $\mathbf{EL}^B_{n,k,r}$. Now define

$$\gamma_{k,r} = \lim_{n \longrightarrow \infty} \frac{\mathbf{EL}_{n,k,r}}{n}, \; \gamma^B_{k,r} = \lim_{n \longrightarrow \infty} \frac{\mathbf{EL}^B_{n,k,r}}{n}$$

A simple but quite interesting fact is

Claim 1

$$\lim_{r \longrightarrow \infty} \gamma^B_{k,r} = \gamma^B_k \; \textit{and} \; \lim_{r \longrightarrow \infty} \gamma_{k,r} = \gamma_k$$

Proof. r-reach effectively reduces the allowable paths. It is easy to see that for fixed values of ϵ_{ij}, $\mathbf{D}_{n,n} \geqslant \mathbf{R}_{n,n}$, and therefore

$$\mathbf{EL}^B_{n,k,r} \leq \mathbf{EL}^{B(k)}_n \Longrightarrow \gamma^B_{k,r} \leq \gamma^B_k \Longrightarrow \lim_{r \longrightarrow \infty} \gamma^B_{k,r} \leq \gamma^B_k$$

Next apply superadditivity and $\mathbf{EL}^B_{r,k,r} = \mathbf{EL}^{B(k)}_r$ to show

$$\gamma^B_{k,r} = \lim_{n \longrightarrow \infty} \frac{\mathbf{EL}^B_{n,k,r}}{n} \geqslant \frac{\mathbf{EL}^B_{r,k,r}}{r} = \frac{\mathbf{EL}^{B(k)}_r}{r}.$$

Taking the limit of both sides yields

$$\lim_{r \longrightarrow \infty} \gamma^B_{k,r} \geqslant \lim_{r \longrightarrow \infty} \frac{\mathbf{EL}^{B(k)}_r}{r} = \gamma^B_k$$

The analogous result for the Random String model is proved the same way. ■

2 Solutions to Bernoulli Matching model r-reach for small r

For small r, the percolation problem can be dissected in full detail. The approach used is fairly straight-foward and computationally intensive. Unfortunately it appears that the r-reach problem is not as elegant as the original–possibly because of the "discontinuous" boundary effects at the displaced diagonals $(i, i+r)$ and $(i+r, i)$. There are several reasons this problem is worth studying, however. First of all it gives lower bounds for the original problem. Also, it is an interesting setting to compare the Random String model with the Bernoulli Matching model. The methods outlined below seem very difficult to use to solve the problem for general r, however they provide foundations for numerical work on large r.

The basic idea of the following analyses is to break the lattice into sections consisting of the $2r+1$ vertices $(n-r, n), (n-r+1, n), ...(n, n), (n, n-1), ...(n, n-r)$

and then compute probabilities that $\mathbf{R}$ takes on specific values at these vertices. We only need to know the distribution of the n^{th} section to compute the distribution of the $(n+1)^{st}$ section. More formally, let $P_n(z)$ be the probability that $\mathbf{R}_{n,n} = z$. For notational convenience let $x_0 = y_0 = z$. For $(n \geqslant r)$ let $R_n(z, x_1, y_1, x_2, y_2, ..., x_r, y_r)$ be the event that ($\mathbf{R}_{n-i,n} = x_i$ and $\mathbf{R}_{n,n-i} = y_i$ $\forall i \in \{0, 1, ..., r\}$). Also define

$$P_n(z, x_1, y_1, x_2, y_2, ..., x_r, y_r) = \Pr(R_n(z, x_1, y_1, x_2, y_2, ..., x_r, y_r)).$$

Let $\overrightarrow{P_n(z)}$ be a row vector of length 2^{2r} whose set of components is

$$\{P_n(z, x_1, y_1, x_2, y_2, ..., x_r, y_r) : \forall i \in \{1, 2, ..., r\},$$
$$x_i = x_{i-1} - d_i^x \text{ and } y_i = y_{i-1} - d_i^y \text{ for some } d_i^x, d_i^y \in \{0, 1\}\}.$$

The order of these components in the vector is not important; we will need to pick an order later to do matrix multiplication, but for now we will leave this unspecified. The values of $\mathbf{R}$ at adjacent lattice points can only differ by 1 or 0 so the vector $\overrightarrow{P_n(z)}$ contains all possible values for vertices in the same section as (n, n). Thus

$$P_n(z) = \sum_{i=1}^{2^{2r}} \overrightarrow{P_n(z)}_i = \overrightarrow{P_n(z)}\mathbf{1}$$

where $\mathbf{1}$ is the column vector $(1, ..., 1)'$.

Now we look at the relationship between $\overrightarrow{P_n(z)}$ and $\overrightarrow{P_{n-1}(z)}$. Let $x'_0 = y'_0 = z'$. If $\overrightarrow{P_n(z)}_j = P_n(z, x_1, y_1, x_2, y_2, ..., x_r, y_r)$ and $\overrightarrow{P_{n-1}(z')}_i = P_{n-1}(z', x'_1, y'_1, x'_2, y'_2, ..., x'_r, y'_r)$ and $z' \in \{z,\, z-1\}$ define

$$\Pr(R_n(z, x_1, y_1, x_2, y_2, ..., x_r, y_r) \text{ and } R_{n-1}(z', x'_1, y'_1, x'_2, y'_2, ..., x'_r, y'_r)) = \begin{cases} \mathbf{M}_{ij} & \text{if } z' = z \\ \mathbf{N}_{ij} & \text{if } z' = z-1 \end{cases} \tag{1}$$

It sufficed to define this only for $z' = z$ or $z-1$ because otherwise the probability is 0. Therefore summing over all possibilites for $R_{n-1}()$ in the above expression gives us $\overrightarrow{P_n(z)}_j$:

$$\sum_{i=1} \overrightarrow{P_{n-1}(z)}_i \mathbf{M}_{ij} + \sum_{i=1} \overrightarrow{P_{n-1}(z-1)}_i \mathbf{N}_{ij} = \Pr(R_n(z, x_1, y_1, x_2, y_2, ..., x_r, y_r)) = \overrightarrow{P_n(z)}_j$$

Taking the convention that $\overrightarrow{P_n(z)}$ is the zero vector for $n < r$, this yields the recurrence that is true for all $n \neq r$:

$$\overrightarrow{P_n(z)} = \overrightarrow{P_{n-1}(z)}\mathbf{M} + \overrightarrow{P_{n-1}(z-1)}\mathbf{N} \ (n \neq r) \tag{2}$$

Now we will construct some generating functions. The convention made above allows the generating function variables n and z to extend over all integers. We will work with the two different generating functions $\overrightarrow{H_n(b)} = \sum_z \overrightarrow{P_n(z)} b^z$ and $\overrightarrow{G(a, b)} = \sum_{n,z} \overrightarrow{P_n(z)} a^n b^z$.

2.1 The generating function $\overrightarrow{G(a,b)}$

Multiplying (2) by $a^n b^z$ and summing over all $n \neq r$ and all z yields

$$\sum_{n\neq r,z} \overrightarrow{P_n(z)} a^n b^z = \sum_{n\neq r,z} (\overrightarrow{P_{n-1}(z)}\mathbf{M} a^n b^z) + \sum_{n\neq r,z} (\overrightarrow{P_{n-1}(z-1)}\mathbf{N} a^n b^z)$$

Add $a^r \overrightarrow{H_r(b)}$ to both sides to obtain

$$\sum_{n,z} \overrightarrow{P_n(z)} a^n b^z = \left(\sum_{n\neq r,z} \overrightarrow{P_{n-1}(z)} a^n b^z\right)\mathbf{M} + \left(\sum_{n\neq r,z} \overrightarrow{P_{n-1}(z-1)} a^n b^z\right)\mathbf{N} + a^r \overrightarrow{H_r(b)}$$

Since $\overrightarrow{P_{r-1}(z)}$ is the zero vector, this becomes

$$\overrightarrow{G(a,b)} = a\overrightarrow{G(a,b)}\mathbf{M} + ab\overrightarrow{G(a,b)}\mathbf{N} + a^r \overrightarrow{H_r(b)}.$$

Then

$$\overrightarrow{G(a,b)}(\mathbf{I} - a\mathbf{M} - ab\mathbf{N}) = a^r \overrightarrow{H_r(b)}. \tag{3}$$

2.2 The generating function $\overrightarrow{H_n(b)}$

.We can also multiply (2) by b^z and sum over all z to obtain

$$\begin{aligned} \sum_z \overrightarrow{P_n(z)} b^z &= \left(\sum_z \overrightarrow{P_{n-1}(z)} b^z\right)\mathbf{M} + \left(\sum_z \overrightarrow{P_{n-1}(z-1)} b^z\right)\mathbf{N} \quad (n \neq r) \implies \\ \overrightarrow{H_n(b)} &= \overrightarrow{H_{n-1}(b)}\mathbf{M} + b\overrightarrow{H_{n-1}(b)}\mathbf{N} \quad (n \neq r) \end{aligned}$$

This shows we can obtain $\overrightarrow{H_n(b)}$ by successive multiplications by $\mathbf{M}+b\mathbf{N}$; that is, let $\mathbf{T}(b) = \mathbf{M} + b\mathbf{N}$.

$$\overrightarrow{H_n(b)} = \overrightarrow{H_r(b)}\mathbf{T}(b)^{n-r}$$

To obtain the behavior of $\mathbf{T}(b)^{n-r}$ as $n \longrightarrow \infty$ we assume from now on $b \geqslant 0$. We can then apply results about positive matrices (see e.g. [5]). Let $\det(\mathbf{T}(b) - \lambda\mathbf{I}) = g(\lambda, b)$, a polynomial in λ and b. $g(\lambda, b) = (\lambda - f_1(b))(\lambda - f_2(b))...(\lambda - f_{2^{2r}}(b))$. Let $\mathbf{e}(b) = (e_1(b), ..., e_r(b))' > \mathbf{0}$ be s.t. $\mathbf{T}(b)\mathbf{e}(b) = \mathbf{e}(b)f_1(b)$ and let $\mathbf{e}^*(b) = (e_1^*(b), ..., e_r^*(b)) > \mathbf{0}'$ s.t. $\mathbf{e}^*(b)f_1(b) = \mathbf{e}^*(b)\mathbf{T}(b)$. Normalize $\mathbf{e}(b)$ and $\mathbf{e}^*(b)$ so that $\mathbf{e}(b)\mathbf{1} = 1, \mathbf{e}^*(b)\mathbf{1} = 1$. Applying results for positive matrices,

$$\lim_{n\longrightarrow\infty} \frac{\mathbf{T}(b)^n}{f_1(b)^n} = \mathbf{e}(b)\mathbf{e}^*(b) \Rightarrow \lim_{n\longrightarrow\infty} \frac{(\mathbf{T}(b)^n)_{ij}}{nf_1(b)^n} = 0 \tag{4}$$

When $b = 1$, this becomes

$$\lim_{n\longrightarrow\infty} \mathbf{T}(1)^n = \mathbf{1}\mathbf{e}^*(1) \tag{5}$$

since $\mathbf{T}(1)$ is the transition matrix between probability distributions $\overrightarrow{H_{n-1}(1)}$ and $\overrightarrow{H_n(1)}$.

Let $h_n(b) = \frac{(\mathbf{T}(b)^n)_{ij}}{nf_1(b)^n}$. We need the following limit result to complete the analysis. It appears that it should follow from (4), but a proof eludes us. For now, we will assume it to complete the analysis.

Claim 2

$$\lim_{n\longrightarrow\infty} \frac{dh_n(b)}{db} = 0$$

The next step is

$$\frac{dh_n(b)}{db} = \frac{1}{nf_1(b)^n}\frac{d(\mathbf{T}(b)^n)_{ij}}{db} - \frac{(\mathbf{T}(b)^n)_{ij}}{f_1(b)^{n+1}}\frac{df_1(b)}{db} \Rightarrow$$

$$\left.\frac{dh_n(b)}{db}\right|_{b=1} = \left.\frac{1}{n}\frac{d(\mathbf{T}(b)^n)_{ij}}{db}\right|_{b=1} - (\mathbf{T}(1)^n)_{ij}\left.\frac{df_1(b)}{db}\right|_{b=1} \Rightarrow$$

$$\lim_{n\longrightarrow\infty}\left.\frac{1}{n}\frac{d(\mathbf{T}(b)^n)}{db}\right|_{b=1} = \lim_{n\longrightarrow\infty}(\mathbf{T}(1)^n)\left.\frac{df_1(b)}{db}\right|_{b=1} = \mathbf{1}\mathbf{e}^*(1)\left.\frac{df_1(b)}{db}\right|_{b=1} \quad (6)$$

Where the last implication follows from the unproven claim and (5). Now we can apply this result to find $\mathbf{E}\overrightarrow{P_n(z)}$ which is defined below

$$\mathbf{E}\overrightarrow{P_n(z)} \equiv \sum_z z\overrightarrow{P_n(z)} = \left.\frac{d\overrightarrow{H_n(b)}}{db}\right|_{b=1}$$

Dividing by n and taking the limit of both sides yields

$$\lim_{n\longrightarrow\infty}\frac{\mathbf{E}\overrightarrow{P_n(z)}}{n} = \lim_{n\longrightarrow\infty}\frac{1}{n}\left.\frac{d\overrightarrow{H_n(b)}}{db}\right|_{b=1} = \lim_{n\longrightarrow\infty}\frac{1}{n}\left.\frac{d(\overrightarrow{H_r(b)}\mathbf{T}(b)^{n-r})}{db}\right|_{b=1} =$$

$$\lim_{n\longrightarrow\infty}\left(\frac{1}{n}\mathbf{T}(1)^{n-r}\left.\frac{d(\overrightarrow{H_r(b)})}{db}\right|_{b=1} + \frac{1}{n}\overrightarrow{H_r(1)}\left.\frac{d(\mathbf{T}(b)^{n-r})}{db}\right|_{b=1}\right) =$$

$$\lim_{n\longrightarrow\infty}\left(\frac{1}{n}\mathbf{T}(1)^{n-r}\mathbf{E}\overrightarrow{P_r(z)} + \frac{1}{n}\overrightarrow{H_r(1)}\left.\frac{d(\mathbf{T}(b)^{n-r})}{db}\right|_{b=1}\right) =$$

$$\overrightarrow{H_r(1)}\lim_{n\longrightarrow\infty}\left(\frac{1}{n}\left.\frac{d(\mathbf{T}(b)^{n-r})}{db}\right|_{b=1}\right) = \overrightarrow{H_r(1)}\mathbf{1}\mathbf{e}^*(1)\left.\frac{df_1(b)}{db}\right|_{b=1} = \mathbf{e}^*(1)\left.\frac{df_1(b)}{db}\right|_{b=1}$$

This last line uses (6) and $\overrightarrow{H_n(1)}\mathbf{1} = \sum_z \overrightarrow{P_n(z)}\mathbf{1} = \sum_z P_n(z) = 1$. The equality above and the equation obtained by multiplying it by $\mathbf{1}$ are stated below; they will be useful later.

$$\lim_{n\longrightarrow\infty}\frac{\mathbf{E}\overrightarrow{P_n(z)}}{n} = \mathbf{e}^*(1)\left.\frac{df_1(b)}{db}\right|_{b=1} \text{ and } \lim_{n\longrightarrow\infty}\frac{\mathbf{E}\mathbf{L}^B_{n,k,r}}{n} = \gamma^B_{k,r} = \left.\frac{df_1(b)}{db}\right|_{b=1} \quad (7)$$

The following claim makes computing $\left.\frac{df_1(b)}{db}\right|_{b=1}$ easier.

Claim 3 *Let $f_1(b)$ be the root of g with $f_1(1) = 1$. Then* $\frac{df_1(b)}{db}\big|_{b=1} = -\frac{dg(1,b)}{db}\big|_{b=1} \left(\frac{\lambda-1}{g(\lambda,1)}\right)\Big|_{\lambda=1}$

Proof.

$$\frac{dg(1,b)}{db} = \frac{d(1-f_1(b))}{db}(1-f_2(b))...(1-f_{2^{2r}}(b))+(1-f_1(b))\frac{d((1-f_2(b))...(1-f_{2^{2r}}(b)))}{db}$$

evaluating at $b = 1$ yields

$$-\frac{df_1(b)}{db}\Big|_{b=1}(1-f_2(1))...(1-f_{2^{2r}}(1)).$$

$g(\lambda,1) = (\lambda-1)(\lambda-f_2(1))...(\lambda-f_{2^{2r}}(1))$ so $\lambda-1$ divides $g(\lambda,1)$. $g(\lambda,1)$ has only one root at $\lambda = 1$ because this root corresponds to the eigenvector $\mathbf{1}$ of $\mathbf{T}(1); \mathbf{1}$ is the unique positive eigenvector of $\mathbf{T}(1)$ (see e.g. [5]). Thus $\frac{\lambda-1}{g(\lambda,1)}$ is defined at $\lambda = 1$ and equals

$$\frac{1}{(1-f_2(1))...(1-f_{2^{2r}}(1))}$$

from which the claim follows directly. ■

2.3 Detailed analysis of 1-reach

When $r = 1$, $\overrightarrow{P_n(z)} = (P_n(z,z,z), P_n(z,z,z-1), P_n(z,z-1,z), P_n(z,z-1,z-1))$. The matrices $\mathbf{M}$ and $\mathbf{N}$ are not difficult to compute by hand; they are

$$\begin{array}{ll} P_{n-1}(z,z,z) & \left[\begin{array}{cccc} \frac{(k-1)^3}{k^3} & 0 & 0 & 0 \\ \frac{(k-1)^2}{k^2} & 0 & 0 & 0 \\ \frac{(k-1)^2}{k^2} & 0 & 0 & 0 \\ \frac{k-1}{k} & 0 & 0 & 0 \end{array}\right] \\ P_{n-1}(z,z,z-1) & \\ P_{n-1}(z,z-1,z) & \\ P_{n-1}(z,z-1,z-1) & \end{array} = \mathbf{M}$$

$$\begin{array}{ll} P_{n-1}(z-1,z-1,z-1) & \left[\begin{array}{cccc} \frac{1}{k^2} & \frac{k-1}{k^2} & \frac{k-1}{k^2} & \frac{(k-1)^2}{k^3} \\ 0 & \frac{1}{k} & 0 & \frac{k-1}{k^2} \\ 0 & 0 & \frac{1}{k} & \frac{k-1}{k^2} \\ 0 & 0 & 0 & \frac{1}{k} \end{array}\right] \\ P_{n-1}(z-1,z-1,z-2) & \\ P_{n-1}(z-1,z-2,z-1) & \\ P_{n-1}(z-1,z-2,z-2) & \end{array} = \mathbf{N}$$

The expressions to the left of each matrix label the rows according to the component order defined above; the columns correspond to $P_n(z,z,z), P_n(z,z,z-1), P_n(z,z-1,z), P_n(z,z-1,z-1)$ in that order. We can also easily compute by hand $\overrightarrow{H_1(b)} = (\frac{k-1}{k}, 0, 0, \frac{b}{k})$.

2.3.1 The two variable generating function

(3) gives us

$$\overrightarrow{G(a,b)} = a\left(\frac{k-1}{k}, 0, 0, \frac{b}{k}\right)(\mathbf{I} - a\mathbf{M} - ab\mathbf{N})^{-1}.$$

Solving this problem with the two variable generating function is computationally intensive, but it's nothing Maple can't handle. We obtain

$$\overrightarrow{G(a,b)}' = \begin{bmatrix} ak^2(k-1)(-k+ab) \\ -a^2b(k-1)^2k \\ -a^2b(k-1)^2k \\ -ab(a^2b^2-abk^2-abk+k^3) \end{bmatrix} \div$$

$$(a^3b^3-a^2b^2(k^2+2k)+a^2b(k^3-3k^2+3k-1)+ab(2k^3+k^2)+a(k^4-3k^3+3k^2-k)-k^4)$$

This potentially gives us the entire distribution of the two variables. The generating function for the expected value of $\overrightarrow{P_n(z)}$, $\mathbf{E}\overrightarrow{P_n(z)} = \sum_z z\overrightarrow{P_n(z)}$, is found by differentiating with respect to b and then evaluating at $b = 1$. We restrict to the $k = 2$ case to make the expression simpler and more readable.

$$\sum_n \mathbf{E}\overrightarrow{P_n(z)}'a^n = \begin{bmatrix} -8a^2(a^3-7a^2+14a-12) \\ 4a^2(a^3-4a^2-a+8) \\ 4a^2(a^3-4a^2-a+8) \\ 8a(3a^3-16a^2+26a-16) \end{bmatrix} (a^3-7a^2+22a-16)^{-2}$$

Using Mathematica's Discrete Math Rsolve package and a little computation by hand, we get

$$\mathbf{E}\overrightarrow{P_n(z)}' = \begin{bmatrix} \frac{32}{121}n - \frac{344}{1331} + 2^{-2n}O(n) \\ \frac{16}{121}n - \frac{40}{1331} + 2^{-2n}O(n) \\ \frac{16}{121}n - \frac{40}{1331} + 2^{-2n}O(n) \\ \frac{24}{121}n + \frac{72}{1331} + 2^{-2n}O(n) \end{bmatrix}$$

where the $O(n)$ terms vary like $n\cos(n\theta)$. Summing these components gives us

$$\mathbf{EL}^B_{n,2,1} = \frac{8}{11}n - \frac{32}{121} + 2^{-2n}O(n).$$

Mathematica can also solve the case for general k, but the expression is difficult to pick apart because it's so long. To get the behavior of $\frac{\mathbf{EL}^B_{n,k,1}}{n}$ divide $\sum_n \mathbf{EL}^B_{n,k,1}a^n$ by a and integrate with respect to a. This generating function has the form

$$\sum_n \frac{\mathbf{EL}^B_{n,k,1}}{n}a^n = \frac{c_1(k)}{1-a} + c_2(k)\ln(a-1) - c_3(k)\ln(O(a^2)) + c_4(k)arctanh(O(a))$$

where $c_i(k)$ are functions only of k; the $O(a^2)$ and $O(a)$ are quadratic and linear polynomials in a with coefficients a function of k. Inferring from the $k = 2$ case, we guess that

$$\mathbf{EL}^B_{n,k,1} = c_1(k)n - c_2(k) + 2^{-2n}O(n)c_5(k).$$

And Maple tells us that

$$c_1(k) = \frac{3k+2}{(k^2+3k+1)},\ c_2(k) = \frac{k(2k^2+3k+2)}{(k^4+6k^3+11k^2+6k+1)}$$

2.3.2 The one variable generating function

$$\det(\mathbf{T}(b)-\lambda\mathbf{I})=g(\lambda,b)=\frac{1}{k^5}(-\lambda k+b)\times$$

$$(b^3-b^2k\lambda(k+2)+b\lambda(k^3+2k^3\lambda-3k^2+k^2\lambda+3k-1)+\lambda^2k(k^3-\lambda k^3-3k^2+3k-1) \quad (8)$$

By (3)

$$\left.\frac{df_1(b)}{db}\right|_{b=1}=-\left.\frac{dg(1,b)}{db}\right|_{b=1}\left.\left(\frac{\lambda-1}{g(\lambda,1)}\right)\right|_{\lambda=1}=$$

$$-\left(-\frac{1}{k^5}(k-1)^3(3k+2)\right)\left(\frac{k^5}{(k^2+3k+1)(k-1)^3}\right)=\frac{3k+2}{(k^2+3k+1)}.$$

Next we compute $\mathbf{e}^*(1)$ (using Maple even though it's not necessary)

$$\mathbf{e}^*(1)=N\begin{bmatrix}k & 1 & 1 & \frac{1+k}{k}\end{bmatrix}$$

Choose N so that $\mathbf{e}^*(1)\mathbf{1}=1\Rightarrow N=\frac{k}{k^2+3k+1}$. From (7) we have

$$\lim_{n\longrightarrow\infty}\frac{\mathbf{E}\overrightarrow{P_n(z)}}{n}=\mathbf{e}^*(1)\left.\frac{df_1(b)}{db}\right|_{b=1}=n\frac{k(3k+2)}{(k^2+3k+1)^2}\begin{bmatrix}k & 1 & 1 & \frac{1+k}{k}\end{bmatrix}$$

Summing all the components gives us

$$\gamma_{k,1}^B=\frac{3k+2}{(k^2+3k+1)}$$

This does not give us as much asymptotic information as the two variable generating function, but it is much less messy and allows us to easily see the limiting behavior of $\mathbf{E}\overrightarrow{P_n(z)}$.

It is interesting to compare this limiting behavior to the conjectured behavior for γ_k^B. It is guessed that $\sqrt{k}\gamma_k^B\longrightarrow 2$ as $k\longrightarrow\infty$, whereas $k\gamma_{k,1}^B\longrightarrow 3$ as $k\longrightarrow\infty$.

2.4 2 and 3 reach

When $r=2$, $\mathbf{M}$ and $\mathbf{N}$ are matrices of size 16×16. For the two variable generating function approach, we will restrict to the case $k=2$. Maple can solve for $\overrightarrow{G(a,b)}$; $\overrightarrow{G(a,b)}\mathbf{1}$ is a polynomial in a and b with leading term $a^{11}b^{11}$ divided by a polynomial with leading term $a^{10}b^{10}$. As with 1-reach, we can find $\int\left(\sum_n \mathbf{EL}_{n,2,2}^B a^{n-1}\right)da$ to obtain the limiting behavior of $\mathbf{EL}_{n,2,2}^B$. The result is an expression about a page long that is very difficult to read. But it appears that most relevant parts of it to the asymptotic behavior are:

$$\frac{a(1-a)}{2(1-a)}+\frac{152}{197(1-a)}+\frac{16872(1-a)}{38809(1-a)}\ln(a-1)$$

From which we conclude

$$\mathbf{EL}^B_{n,2,2} \sim \frac{152}{197} n - \frac{16872}{38809}.$$

This seems to be consistent with the Monte Carlo approximations, as will be seen later.

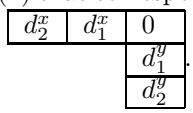
Now for the one variable generating function approach. This can be solved for general k. $g(\lambda, b)$ is too large an expression to be of much worth written down here. The resulting expression for $\frac{df_1(b)}{db}\Big|_{b=1}$ is surprisingly simple however.

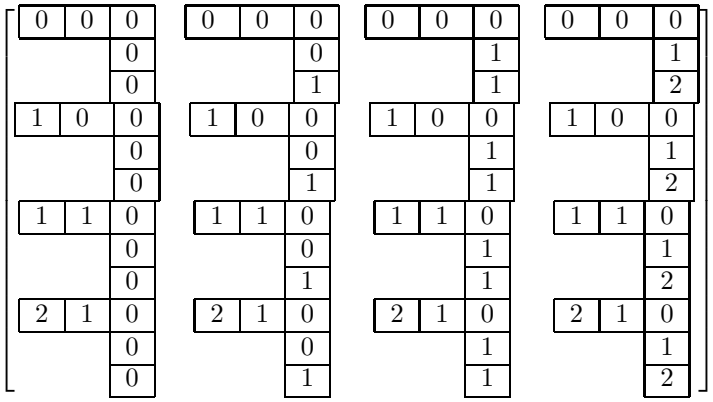
$$\left.\frac{df_1(b)}{db}\right|_{b=1} = -\left.\frac{dg(1,b)}{db}\right|_{b=1} \left.\left(\frac{\lambda - 1}{g(\lambda, 1)}\right)\right|_{\lambda=1} =$$

$$-\left(-\frac{1}{k^{28}}(k+1)(5k^3+20k^2+15k+2)(k^4+k^3+3k^2+k+1)(k-1)^{15}(k^4+3k^3+5k^2+3k+1)\right) \times$$

$$\left(\frac{k^{28}}{(k^4+3k^3+5k^2+3k+1)(k-1)^{15}(k+1)(k^4+k^3+3k^2+k+1)(k^4+10k^3+20k^2+10k+1)}\right) =$$

$$\frac{5k^3+20k^2+15k+2}{k^4+10k^3+20k^2+10k+1} = \gamma^B_{k,2}.$$

when $k = 2$, this gives $\frac{152}{197}$ which confirms part of the guess for $\mathbf{EL}^B_{n,2,2}$ found by the two variable generating function approach. $\mathbf{e}^*(1)$ is illustrated as follows: We reshape the vector into a matrix so that it is easier to read. The component of $\mathbf{e}^*(1)$ that corresponds to $P_n(z, z-d_1^x, z-d_1^y, z-d_2^x, z-d_2^y)$ in $\overrightarrow{P_n(z)}$ is represented by  .



$$\Updownarrow$$

$$\begin{bmatrix} k^2 & 2\,k & k & 1 \\ 2k & k+4 & k+2 & \frac{2\,(k+1)}{k} \\ k & k+2 & k+1 & \frac{2k+1}{k} \\ 1 & \frac{2\,(k+1)}{k} & \frac{2k+1}{k} & \frac{k^2+4\,k+1}{k^2} \end{bmatrix}$$

To normalize $\mathbf{e}^*(1)$, the above matrix must be multiplied by $\frac{k^2}{k^4+10k^3+20k^2+10k+1}$ which finally gives

$$\overrightarrow{\mathbf{E}P_n(z)} \sim n\frac{k^2(5k^3+20k^2+15k+2)}{(k^4+10k^3+20k^2+10k+1)^2}\begin{bmatrix} k^2 & 2\,k & k & 1 \\ 2k & k+4 & k+2 & \frac{2\,(k+1)}{k} \\ k & k+2 & k+1 & \frac{2k+1}{k} \\ 1 & \frac{2\,(k+1)}{k} & \frac{2k+1}{k} & \frac{k^2+4\,k+1}{k^2} \end{bmatrix}$$

The case $r = 3$, $k = 2$ is also computable in a reasonable amount of time (it took Maple about a half an hour on a 1992 Mega Hertz Dell). The result is

$$\gamma_{2,3}^B = \frac{3376}{4279}.$$

3 Applications to the Random String model

The machinery developed for r-reach with the Bernoulli matching model can be applied to 1-reach with the Random String model when $k = 2$. For $r > 1$, it appears this same brute force conditional probability approach is so complicated as to be almost useless. $r = 1$ and $k > 2$ seems significantly more difficult than $r = 1$, $k = 2$, which is rather surprising. We get an interesting reduction for the $k = 2$ case, as will be seen shortly. The reason for pursuing this approach despite its appearance of being difficult to generalize, is that it may lead to a short proof of $\gamma_{2,1}^B > \gamma_{2,1}$, which may be generalizable. It has been conjectured that $\lim_{n\longrightarrow\infty}\gamma_k^B\sqrt{k} = \lim_{n\longrightarrow\infty}\gamma_k\sqrt{k}$ (actually Sankoff and Mainville conjectured that $\lim_{n\longrightarrow\infty}\gamma_k\sqrt{k} = 2$ (see e.g. [3]) and Boutet de Monvel [2] conjectured that $\lim_{n\longrightarrow\infty}\gamma_k^B\sqrt{k} = 2$). If 1-reach is solved for general k, it may provide some insights into this problem.

3.1 Detailed analysis of 1-reach

The reduction for the case $k = 2$ is not difficult, but it requires a fair amount of notation to discuss.

Definition. If ϵ_{ij} is defined for $|i-j| \le r$ and $1 \le i,j \le n$, ϵ_{ij} is a string realizable configuration of weight w if $\epsilon_{ij} = \delta_{u(i),v(j)}$ for w distinct $(u,v)\epsilon\Sigma^n x\Sigma^n$.

It is easy to convince oneself of the following claim by doing a case by case analysis for $n = 3$. Such an analysis extends easily to general n.

Claim 4 *Let $k = 2$ and let ϵ_{ij} be defined for $|i-j| \le 1$ and $1 \le i,j \le n$. ϵ_{ij} is a string realizable configuration of weight 2 if*

$$\forall i \in \{1,...,n\},\ \epsilon_{i-1,i-1} + \epsilon_{i,i-1} + \epsilon_{i-1,i}\ + \epsilon_{i,i} \in \{0,2,4\} \tag{9}$$

and is a string realizable configuration of weight 0 otherwise.

Proof. $k = 2$ means the alphabet, Σ, is $\{0, 1\}$ so that $\begin{bmatrix} \delta_{u(i-1),v(i)} & \delta_{u(i),v(i)} \\ \delta_{u(i-1),v(i-1)} & \delta_{u(i),v(i-1)} \end{bmatrix} \equiv \mathbf{X}(u(i-1,i), v(i-1,i))$ must be in the set

$$\mathbf{Y} \equiv \left\{ \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \right\}. \tag{10}$$

This shows that if the condition in (9) fails, ϵ_{ij} is a string realizable configuration of weight 0.

For the other part of the claim we proceed by induction on n. The case $n = 1$ can be seen by noting that each element of $\mathbf{Y}$ is equal to 2 of the 16 possibilities for $\mathbf{X}(u(i-1,i), v(i-1,i))$. Suppose $n > 1$ and the claim holds for $n-1$. Let u, v, u', v' be the strings of length $n-1$ such that $\forall i, j \in \{1, ..., n-1\}$ and $|i-j| \leq 1$, $\epsilon_{ij} = \delta_{u(i),v(j)} = \delta_{u'(i),v'(j)}$. By hypothesis, $\mathbf{Z}_n \equiv \begin{bmatrix} \epsilon_{n-1,n} & \epsilon_{nn} \\ \epsilon_{n-1,n-1} & \epsilon_{n,n-1} \end{bmatrix}$ is one of the eight matrices belonging to $\mathbf{Y}$. For each matrix in $\mathbf{Y}$, we can choose $u(n)$ and $v(n)$ as shown below so that $\forall i, j \in \{1, ..., n\}$ and $|i-j| \leq 1$, $\epsilon_{ij} = \delta_{u(i),v(j)}$. The same goes for u' and v'.

$\mathbf{Z}_n$	$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$
$u(n)=,$ $v(n)=$	$u(n-1),$ $v(n-1)$	$1-u(n-1),$ $1-v(n-1)$	$1-u(n-1),$ $v(n-1)$	$u(n-1),$ $1-v(n-1)$	$u(n-1),$ $v(n-1)$	$1-u(n-1),$ $1-v(n-1)$	$1-u(n-1),$ $v(n-1)$	$u(n-1),$ $1-v(n-1)$

This shows ϵ_{ij} is a string realizable configuration of weight at least 2. The weight cannot exceed 2 because then ϵ_{ij} restricted to $i, j \in \{1, ..., n-1\}$ would have weight greater than 2. ■

This claim lets us count the probabilities $P_n(z, x_1, y_1)$ much like we did for the Bernoulli Matching model. We define the analogous probability vector but we have to break $P_n(z, x_1, y_1)$ into two pieces: $P_n(z, x_1, y_1) = P_n^{\text{on}}(z, x_1, y_1) + P_n^{\text{off}}(z, x_1, y_1)$.

$P_n^{\text{on}}(z, x_1, y_1) = \Pr(R_n(z, x_1, y_1) \text{ and } \epsilon_{nn} = 1)$, $P_n^{\text{off}}(z, x_1, y_1) = \Pr(R_n(z, x_1, y_1) \text{ and } \epsilon_{nn} = 0)$.

The reason for this split is that we need to know $\epsilon_{n-1,n-1}$ to determine how $\{\mathbf{R}_{n-1,n-1}, \mathbf{R}_{n-2,n-1}, \mathbf{R}_{n-1,n-2}\}$ affects $\{\mathbf{R}_{n,n}, \mathbf{R}_{n-1,n}, \mathbf{R}_{n,n-1}\}$. The computation of $\mathbf{M}$ and $\mathbf{N}$ was done by hand and was a little trickier than for the Bernoulli Matching model.

$$\begin{matrix} P_{n-1}^{\text{off}}(z,z,z) \\ P_{n-1}^{\text{off}}(z,z,z-1) \\ P_{n-1}^{\text{off}}(z,z-1,z) \\ P_{n-1}^{\text{off}}(z,z-1,z-1) \\ P_{n-1}^{\text{on}}(z,z,z) \\ P_{n-1}^{\text{on}}(z,z,z-1) \\ P_{n-1}^{\text{on}}(z,z-1,z) \\ P_{n-1}^{\text{on}}(z,z-1,z-1) \end{matrix} \begin{bmatrix} 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \mathbf{M}$$

$$\begin{array}{l} P^{\text{off}}_{n-1}(z-1,z-1,z-1) \\ P^{\text{off}}_{n-1}(z-1,z-1,z-2) \\ P^{\text{off}}_{n-1}(z-1,z-2,z-1) \\ P^{\text{off}}_{n-1}(z-1,z-2,z-2) \\ P^{\text{on}}_{n-1}(z-1,z-1,z-1) \\ P^{\text{on}}_{n-1}(z-1,z-1,z-2) \\ P^{\text{on}}_{n-1}(z-1,z-2,z-1) \\ P^{\text{on}}_{n-1}(z-1,z-2,z-2) \end{array} \begin{bmatrix} 1/4 & 0 & 0 & 0 & 0 & 1/4 & 1/4 & 0 \\ 0 & 1/4 & 0 & 0 & 0 & 1/4 & 0 & 1/4 \\ 0 & 0 & 1/4 & 0 & 0 & 0 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 1/4 & 1/4 & 0 & 1/4 & 0 & 0 & 1/4 \\ 0 & 1/4 & 0 & 0 & 0 & 1/4 & 0 & 1/4 \\ 0 & 0 & 1/4 & 0 & 0 & 0 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 \end{bmatrix} = \mathbf{N}$$

The two variable generating function approach determines the fine limiting behavior:

$$\sum_n \mathbf{EL}_{n,2,1} a^n = \frac{a(a^2-2a+8)}{2(a^2-4a+8)(a-1)^2}$$

Using Mathematica's Discrete Math Rsolve package and a little computation by hand, we obtain

$$\mathbf{EL}_{n,2,1} = \frac{7}{10}n - \frac{7}{25} + 2^{-3n/2}O(1).$$

where the $O(1)$ term varies like $\cos(n\theta)$. We will compare this result to numerical approximations.

The one variable generating function produces

$$\det(\mathbf{T}(b)-\lambda\mathbf{I})=g(\lambda,b) = -\frac{1}{128}\lambda^3(b-2\lambda)(b-4\lambda)(b^3+b^2(-8\lambda+1)+2b\lambda(-1+10\lambda)+4\lambda^2(-4\lambda+1))$$

and

$$\left.\frac{df_1(b)}{db}\right|_{b=1} = -\left.\frac{dg(1,b)}{db}\right|_{b=1} \left.\left(\frac{\lambda-1}{g(\lambda,1)}\right)\right|_{\lambda=1} = -\left(-\frac{21}{128}\right)\left(\frac{64}{15}\right) = \frac{7}{10}$$

$\mathbf{e}^*(1) = \frac{1}{20}\begin{bmatrix}8 & 1 & 1 & 0 & 0 & 3 & 3 & 4\end{bmatrix}$.

We can also "blow up" the 1-reach Bernoulli Matching model, so that we work with $\overrightarrow{P^{\text{on}}_n(z)}$ and $\overrightarrow{P^{\text{off}}_n(z)}$ even though we don't need to. The resulting matrices are included in the appendix. It is interesting to note that the matrices only differ in the two rows corresponding to $P_{n-1}(z-1,z-1,z-1)$. The result is

$$g^B(\lambda,b) = \frac{1}{32}\lambda^4(b-2\lambda)(b^3-8b^2\lambda+b\lambda(1+20\lambda)+2\lambda^2(1-8\lambda))$$

and this polynomial is the same as one obtained earlier (in (8)) except for the λ^4 term. Also,

$\mathbf{e}^{B*}(1) = \frac{1}{22}\begin{bmatrix}7 & 2 & 2 & 0 & 1 & 2 & 2 & 6\end{bmatrix}$ which is more precise behavior than that determined by the 4×4 matrix method.

It is unclear whether there is a more direct way to see that the difference in the matrices for the Random String model and the Bernoulli Matching model lead to the conclusion $\gamma_{2,1} < \gamma^B_{2,1}$

4 Numerical Work

We ran Monte Carlo simulations for $k = 2$ and $r = 1, 2...10, 15, 20, 25, 35, 40$. 10000 trials were computed up to $n = 1000$ for each r. To obtain behavior varying with n, approximations of $\mathbf{EL}^B_{n,2,r}$ and $\mathbf{EL}_{n,2,r}$ for all n from 1 to 1000 were computed for each trial. Ideally, we should have computed separate trials for each n, but these results appear to lead to good extrapolations to large n. Following the work in [2], we extrapolate to $\gamma^B_{2,r}$ from the small n simulations based on;

$$\mathbf{EL}_{n,2,r} \sim \gamma_{2,r} n - A_r, \ \mathbf{EL}^B_{n,2,r} \sim \gamma^B_{2,r} n - A^B_r. \tag{11}$$

Where A_r (A^B_r) is a constant, and was found by minimizing the variance of $\frac{\mathbf{EL}_{n,2,r}+A_r}{n}$ ($\frac{\mathbf{EL}^B_{n,2,r}+A^B_r}{n}$). Extrapolations for $\gamma^B_{2,r}$, A^B_r, and $\gamma_{2,r}$, A_r based on Monte Carlo simulations are shown below. We did this extrapolation from $n = 50...1000$ to minimize the effect of the $2^{-2n}O(n)$ term (we only saw this for $r = 1$, but there are probably similar terms for larger r).

r	1	2	3	4	5	6	7	8	9
$\gamma^B_{2,r}$	0.72726	0.77166	0.78898	0.79813	0.80396	0.80796	0.81119	0.81284	0.81458
A^B_r	0.2771	0.4626	0.5641	0.6852	0.8033	0.9399	0.9931	1.0814	1.1900
$\gamma_{2,r}$	0.70014	0.73767	0.75610	0.76718	0.77467	0.78004	0.78408	0.78726	0.78976
A_r	0.2652	0.4335	0.5748	0.7048	0.8195	0.9218	1.0163	1.1121	1.2044

r	10	15	20	25	30	35	40
$\gamma^B_{2,r}$	0.81592	0.81994	0.82182	0.82290	0.82355	0.82406	0.82415
A^B_r	1.2653	1.5253	1.6814	1.7536	1.8058	1.8368	1.8395
$\gamma_{2,r}$	0.79180	0.79819	0.80149	0.80340	0.80462	0.80546	0.80603
A_r	1.2877	1.6377	1.8753	2.028	2.1273	2.1939	2.2371

Shown in figure (2) are $\frac{MonteCarlo(\mathbf{EL}^B_{n,2,r})}{n}$ and $\frac{MonteCarlo(\mathbf{EL}^B_{n,2,r})+A_r}{n}$ and the corresponding Random String model data is shown in (3). It appears that the approximation $\mathbf{EL}_{n,2,r} \sim \gamma_{2,r} n - A_r$ gets increasingly worse for larger r and likewise for the Bernoulli Matching model.

For the Bernoulli Matching model we also can compute $\mathbf{EL}^B_{n,2,r}$ exactly for small n by applying (2) directly beginning with $\overrightarrow{P_r(z)}$. This allows us to do two checks on the quality of the Monte Carlo approximations. Firstly, we can observe the difference $\frac{MonteCarlo(\mathbf{EL}^B_{n,2,r})}{n} - \frac{\mathbf{EL}^B_{n,2,r}}{n}$. The statistic

$$S_r \equiv \frac{1}{1000} \sum_{j=1}^{1000} \left(\frac{MonteCarlo(\mathbf{EL}^B_{n,2,r})}{n} - \frac{\mathbf{EL}^B_{n,2,r}}{n} \right)^2$$

gives us an idea of how crude an approximation we get with 10000 trials. Also, we can see how good the approximation $\mathbf{EL}^B_{n,2,r} \sim \gamma^B_{2,r} n - A^B_r$ is by using that

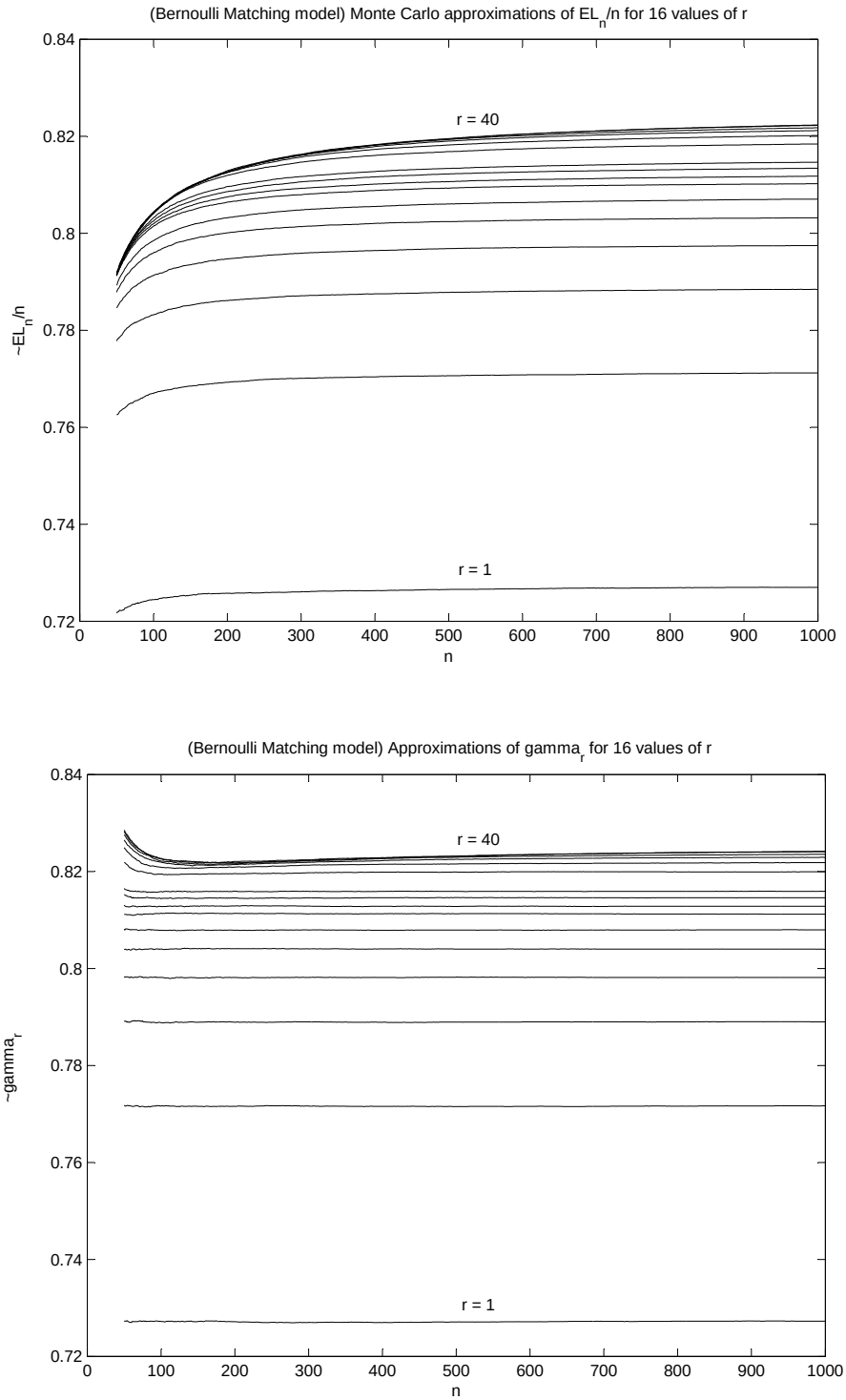
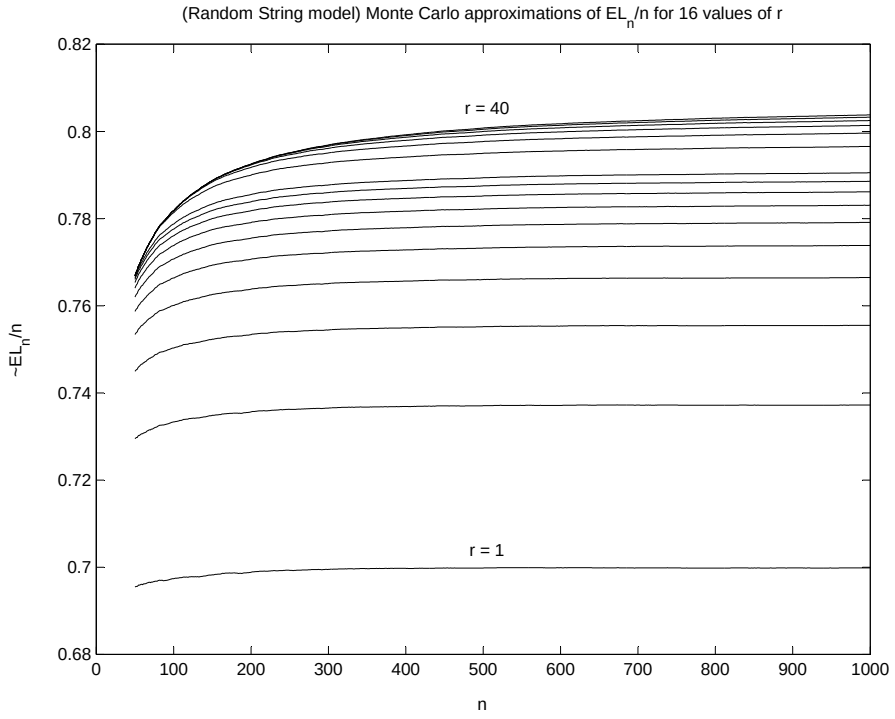


Figure 2: The Monte Carlo approximations of $\frac{\mathbf{EL}^B_{n,2,r}}{n}$ and this same data corrected by (11) to obtain the limiting behavior.



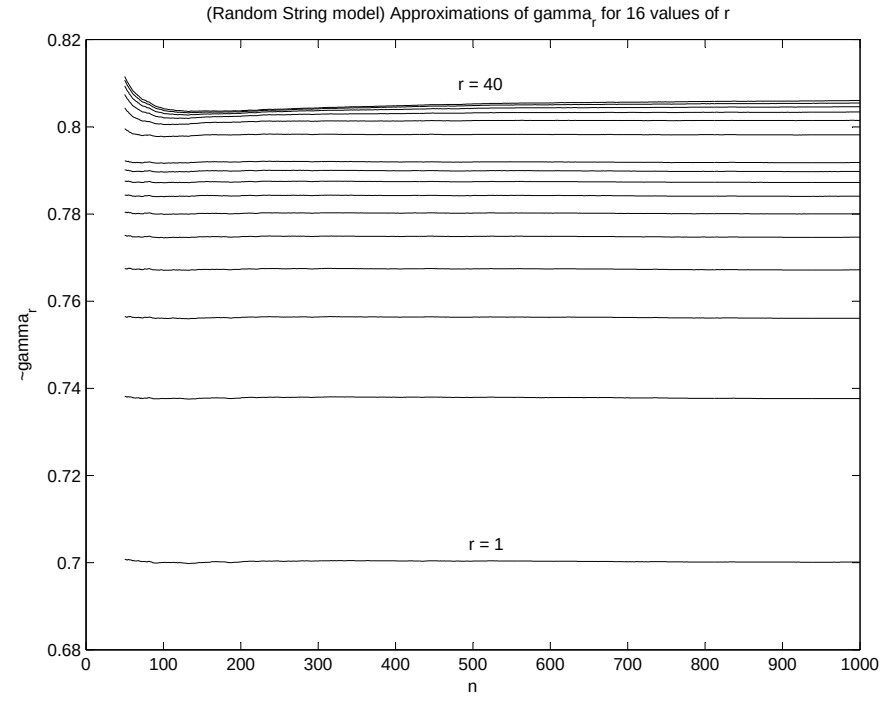


Figure 3: The Monte Carlo approximations of $\frac{\mathbf{EL}_{n,2,r}}{n}$ and this same data corrected by (11) to obtain the limiting behavior.

on the exact values of $\mathbf{EL}^B_{n,2,r}$ to extrapolate $\gamma^B_{2,r}$ (for this extrapolation we use $n = 1...2000$).

r	S_r	Monte Carlo $\gamma^B_{2,r}$	$\gamma^B_{2,r}$ from $\mathbf{EL}^B_{n,2,r}$	$\gamma^B_{2,r}$ from fractions derived previously
1	5.2994×10^{-8}	0.7272634	0.7272727273	0.7272727272
2	5.0758×10^{-8}	0.7716676	0.7715736043	0.7715736040
3	1.0180×10^{-8}	0.7889874	0.7889693851	0.7889693853
4	1.5954×10^{-8}	0.7981354	0.7982222051	–

r	Monte Carlo A^B_r	A^B_r from $\mathbf{EL}^B_{n,2,r}$	A^B_r from fractions derived previously
1	0.2771	0.264463	0.2644628
2	0.4626	0.434745	0.4347445
3	0.5641	0.574312	–
4	0.6852	0.696534	–

We also note that $MonteCarlo(\gamma_{2,1}) = 0.7001417$ compared to $\gamma_{2,1} = .7$ and $MonteCarlo(A_{2,1}) = 0.2652$ compared to $A_{2,1} = .28$

5 Conclusions and future work

It is hoped that the results presented in this paper lead the way to more significant results. In particular, it is hoped that the Random String model analysis may lead to a short proof of $\gamma_{2,1} < \gamma^B_{2,1}$. The limiting behavior of $\gamma_{k,1} - \gamma^B_{k,1}$ would also be of interest. We seek a conjecture for the quantities $\gamma^B_{k,r}$, though it is unclear if trying to determine γ^B_k via $\lim_{r \longrightarrow \infty} \gamma^B_{k,r} = \gamma^B_k$ is a good idea.

The pseudoproof of $\gamma^B_k = \frac{2}{1+\sqrt{k}}$ given by Boutet de Monvel may provide a way to simplify the r-reach computations. The limiting behavior of r-reach may be describable only by differences between adjacent values of $\mathbf{R}$, thereby reducing the "problemsize" from 2^{2r} to $2r$. Preliminary investigations suggest that this reduction may be possible but not as straight forward as the argument in the pseudoproof.

6 Appendix

The expanded version of the Bernoulli Matching model $r = 1, k = 2$ case has matrices as follows. These are given for comparison with the matrices for the Random String model $r = 1, k = 2$ case.

$$
\begin{array}{l}
P^{\text{off}}_{n-1}(z,z,z) \\
P^{\text{off}}_{n-1}(z,z,z-1) \\
P^{\text{off}}_{n-1}(z,z-1,z) \\
P^{\text{off}}_{n-1}(z,z-1,z-1) \\
P^{\text{on}}_{n-1}(z,z,z) \\
P^{\text{on}}_{n-1}(z,z,z-1) \\
P^{\text{on}}_{n-1}(z,z-1,z) \\
P^{\text{on}}_{n-1}(z,z-1,z-1)
\end{array}
\begin{bmatrix}
1/8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} = \mathbf{M}
$$

$$
\begin{array}{l}
P^{\text{off}}_{n-1}(z-1,z-1,z-1) \\
P^{\text{off}}_{n-1}(z-1,z-1,z-2) \\
P^{\text{off}}_{n-1}(z-1,z-2,z-1) \\
P^{\text{off}}_{n-1}(z-1,z-2,z-2) \\
P^{\text{on}}_{n-1}(z-1,z-1,z-1) \\
P^{\text{on}}_{n-1}(z-1,z-1,z-2) \\
P^{\text{on}}_{n-1}(z-1,z-2,z-1) \\
P^{\text{on}}_{n-1}(z-1,z-2,z-2)
\end{array}
\begin{bmatrix}
1/8 & 1/8 & 1/8 & 0 & 1/8 & 1/8 & 1/8 & 1/8 \\
0 & 1/4 & 0 & 0 & 0 & 1/4 & 0 & 1/4 \\
0 & 0 & 1/4 & 0 & 0 & 0 & 1/4 & 1/4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 \\
1/8 & 1/8 & 1/8 & 0 & 1/8 & 1/8 & 1/8 & 1/8 \\
0 & 1/4 & 0 & 0 & 0 & 1/4 & 0 & 1/4 \\
0 & 0 & 1/4 & 0 & 0 & 0 & 1/4 & 1/4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2
\end{bmatrix} = \mathbf{N}
$$

References

[1] R. A. Baeza-Yates, R. Gavaldà, G. Navarro, and R. Scheihing. *Bounding the Expected Length of Longest Common Subsequences and Forests*, Theory Comput. Systems 32 (1999), 435-452.

[2] Boutet de Monvel, J. *Extensive Simulations for Longest Common Subsequences*. Europ. Phys. J. B 7 (1999), 293-308.

[3] V. Dančík. *Expected Length of Longest Common Subsequences*. Ph.D. Thesis, CS Dept., University of Warwick, Warwick, England, 1994.

[4] D. Sankoff and J. B. Kruskal. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, Addison-Wesley, Reading, MA, 1983.

[5] Y. Sinai. *Probability Theory*, Springer-Verlag, Berlin, 1992.