A Appendix

Proofs from section 3 Here were present proofs ommitted in the main paper.

Proof. [Proposition 3.1]

If $m = \mathbf{x}^T\mathbf{1}$, the joint density is:

$$f(\mathbf{p},\mathbf{x}|\rho,\boldsymbol{\alpha}) = p_{\text{Yule}}(m|\rho).f_{\text{Dir}}(\mathbf{p}|\boldsymbol{\alpha}).p_{\text{Mult}}(\mathbf{x}|m,\mathbf{p})$$
$$= \rho B(m, 1+\rho)\frac{\Gamma(\boldsymbol{\alpha}^T\mathbf{1})\Gamma(m+1)}{\prod_{i=1}^d \Gamma(\alpha_i)\Gamma(x_i+1)}\prod_{i=1}^d p_i^{x_i+\alpha_i-1}$$

To complete the proof, integrate out $\mathbf{p}$ and use the multivariate Beta function $B(\mathbf{x}) = \frac{\prod_{i=1}^d \Gamma(x_i)}{\Gamma(\mathbf{x}^T\mathbf{1})}$. ■

We now sketch the proof of Theorem 3.1.

Proof. [Theorem 3.1] The first part follows from the fact: if $(X_1, ..., X_d) \sim \text{Dir}(\alpha_1, ..., \alpha_d)$, we have that $(X_1 + X_2, X_3..., X_d) \sim \text{Dir}(\alpha_1 + \alpha_2, \alpha_3, ..., \alpha_d)$. The approximation to log-logistic distribution is proved in the following lemma. ■

LEMMA A.1. *The tail of the (discrete) Yule distribution is asymptotically log-logistic.*

Proof. Abusing notation, let p_i represent the probability of making an observation i from Yule($\frac{1}{1-s}$), corresponding to preferential attachment with probability parameter s, and let $\rho = 1/(1-s)$.

Stirling's approximation tells us that

$$\lim_{n\to\infty}\frac{n!}{\sqrt{2\pi n}(n/e)^n} = 1.$$

Applying this on the Beta function, for constant y,

$$\lim_{x\to\infty}\frac{B(x,y)}{\Gamma(y)e^y x^{-y}} = 1.$$

Thus, as $i \to \infty$, $p_i \to \rho \exp(1+\rho)\Gamma(1+\rho)\cdot i^{-(1+\rho)}$.

If F is the cumulative distribution function of the Yule distribution, we have, $F(i) = \sum_{j=1}^i p_j = 1 - \sum_{j=i+1}^\infty p_j$. Consider $S(i) = \sum_{j=i+1}^\infty i^{-(1+\rho)}$. It can be seen that

$$\int_{i+1}^\infty x^{-(1+\rho)}dx \le S(i) \le \int_i^\infty x^{-(1+\rho)}dx$$

or

$$(i+1)^{-\rho} \le \rho\cdot S(i) \le i^{-\rho}$$

The logarithm of odds ratio, $\log OR(i) = \log F(i)/(1-F(i))$ will now be bounded as:

$$\lim_{i\to\infty}\frac{\log OR(i)}{\log\left((i+1)^\rho/(\exp(1+\rho)\Gamma(1+\rho)) - 1\right)} \le 1$$

and

$$\lim_{i\to\infty}\frac{\log OR(i)}{\log\left(i^\rho/(\exp(1+\rho)\Gamma(1+\rho)) - 1\right)} \ge 1.$$

Notice that in the limit $i \to \infty$ both denominators are the same as $\log\left(i^\rho/(\exp(1+\rho)\Gamma(1+\rho)\right)$, to get

$$\log OR(i) \to \rho\log i - c$$

where $c = \log\Gamma(1+\rho) + 1 + \rho$ is a constant.

A characteristic property of the log-logistic distribution with parameters α, β, is that the log odds are linear in $\log x$ with slope β and intercept $-\beta\log\alpha$. This completes the proof that the tail of the Yule distribution is asymptotically log-logistic. ■

Before proving the result on confidence intervals, define $\mathbb{I}(z)$ be the indicator that z is true. Under this notation, $n_i = \sum_{j=1}^n \mathbb{I}(\mathbf{x}^j = \mathbf{x}^i)$.

Proof. [Theorem 3.2] Consider $n(\mathbf{x}) = \sum_{j=1}^n \mathbb{I}(\mathbf{x}^j = \mathbf{x})$. Its expectation is $\mathbb{E}(n(\mathbf{x})) = \sum_{j=1}^n \mathbb{P}(\mathbf{x}^j = \mathbf{x}) = np(\mathbf{x})$. Its variance is $\mathbb{V}(n(\mathbf{x})) = \sum_{j=1}^n p(\mathbf{x}^j = \mathbf{x})(1 - p(\mathbf{x}^j = \mathbf{x})) + \tilde{C}(\mathbf{x})$ where $\tilde{C}(\mathbf{x}) := 2\sum_{j=1}^n\sum_{k=1}^{j-1}\text{Cov}(\mathbb{I}(\mathbf{x}^j = \mathbf{x}), \mathbb{I}(\mathbf{x}^k = \mathbf{x}))$. Note that $1 - p(\mathbf{x}) \approx 1$ for almost all unique $\mathbf{x}^i$, since the distribution is skewed. Further, we neglect covariances to get that $\mathbb{V}(n(\mathbf{x})) \approx np(\mathbf{x})$. In other words, an approximate distribution for $n(\mathbf{x})$ is poisson with mean $np(\mathbf{x})$. The confidence interval in Eq. 3.5 follows immediately from [8]. ■

One may use any other confidence intervals for the Poisson distribution.

Estimation of $\boldsymbol{\alpha}$: Here, we describe the math that goes into deriving the fitting algorithm for $\boldsymbol{\alpha}$. The basic ideas are the same as the maximum likelihood estimation of the Dirichlet-Multinomial distribution, as described in [15]. They are included here for completeness.

The likelihood function for $\boldsymbol{\alpha}$, its first and second derivatives are:

(A.1)
$$l(\boldsymbol{\alpha}) = \sum_{i=1}^d[\sum_{j=1}^n \log(\frac{\Gamma(\alpha_i + x_i^{(j)})}{\Gamma(\alpha_i)})] - \sum_{j=1}^n \log\frac{\Gamma(m^{(j)} + \boldsymbol{\alpha}^T\mathbf{1})}{\Gamma(\boldsymbol{\alpha}^T\mathbf{1})}$$

(A.2)
$$(\nabla l(\boldsymbol{\alpha}))_k = n\psi(\boldsymbol{\alpha}^T\mathbf{1}) + \sum_{j=1}^n \psi(\alpha_k + x_k^{(j)}) - n\psi(\alpha_k) - \sum_{j=1}^n \psi(m^{(j)} + \boldsymbol{\alpha}^T\mathbf{1})$$

```
Input: Data, X = {x^(1), x^(2), ..., x^(n)} and α(0), the starting guess
Output: α
for t ← 1 to T do
    α ← α(t − 1)
    z ← nψ'(α^T 1) − Σ_{j=1}^n ψ'(m^(j) + α^T 1)
    //g ∈ R^d and D ∈ R^{d×d} is a diagonal matrix
    for k ← 1 to d do
        g_k ← nψ(α^T 1) + Σ_{j=1}^n ψ(α_k + x_k^(j)) − nψ(α_k) − Σ_{j=1}^n ψ(m^(j) + α^T 1)
        d_kk ← Σ_{j=1}^n ψ'(α_k + x_k^(j)) − nψ'(α_k)
    b ← (Σ_{j=1}^d g_j / d_jj) / (z^{-1} + Σ_{j=1}^d d_jj^{-1})
    s ← D^{-1}(g − b · 1)
    α(t) ← α − s
return α(T)
```

Algorithm 1: MLE-Alpha: Algorithm to find MLE of $\boldsymbol{\alpha}$ of FusionRP. Note that ψ and ψ' are the digamma and trigamma functions respectively

$$\nabla^2 l(\boldsymbol{\alpha}) = \mathbf{D} + z\mathbf{1}\mathbf{1}^T \tag{A.3}$$

where ψ, ψ' are the digamma and trigamma functions respectively, $z = n\psi'(\boldsymbol{\alpha}^T\mathbf{1}) - \sum_{j=1}^n \psi'(m^{(j)} + \boldsymbol{\alpha}^T\mathbf{1})$ and $\mathbf{D}$ is a diagonal matrix with $[\mathbf{D}]_{kk} = \sum_{j=1}^n \psi'(\alpha_k + x_k^{(j)}) - n\psi'(\alpha_k)$. As in the case of the Dirchlet-Multinomial distribution [15], the Hessian $\nabla^2 l(\boldsymbol{\alpha})$ can be inverted efficiently using the Sherman-Morisson identity as:

$$(\nabla^2 l(\boldsymbol{\alpha}))^{-1} = \mathbf{D}^{-1} - \frac{\mathbf{D}^{-1}\mathbf{1}\mathbf{1}^T\mathbf{D}^{-1}}{z^{-1} + \mathbf{1}^T\mathbf{D}^{-1}\mathbf{1}} \tag{A.4}$$

The Newton step will now be: $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} - (\nabla^2 l(\boldsymbol{\alpha}))^{-1}\nabla l(\boldsymbol{\alpha})$. This can be simplified to give algorithm 1. Proof of Proposition 3.2 is now direct.

Proof. [Proposition 3.2] Estimation of s just requires two counts. Because of the special structure in the Hessian (Eq. A.4), Netwon's method will be efficient. Let n_0 be the number of *unique* observations (and not n, the total observations). Group the observations into (unique observation, count) pairs, and each iteration requires us to over all such pairs once. Also, the gradient and the matrix $\mathbf{D}$ are d dimensional. And, in practice, Newton's method requires 5-10 iterations to converge. ∎

It can be seen that $l(\boldsymbol{\alpha})$ is not concave in $\boldsymbol{\alpha}$ and hence we can only efficiently find a local maximizer. Firstly, if we find a good enough starting point $\boldsymbol{\alpha}_0$, we may find a good maximum. The estimate of $\boldsymbol{\alpha}_0$ from $\mathbf{x}^1/m^1, \mathbf{x}^2/m^2, ..., \mathbf{x}^n/m^n \sim \mathrm{Dir}(\boldsymbol{\alpha}_0)$ works as a good initialization, in practice. We use the moment matching estimate of [15].

The fitting algorithm is described in Algorithm 1