Law of large numbers for matchings, extensions and applications

Marc Lelarge

INRIA & ENS

Joint works with Charles Bordenave (CNRS & Uni. Toulouse), Justin Salez (Paris 7), Mathieu Leconte (Technicolor-INRIA), Laurent Massoulié (MSR-INRIA) and Hang Zhou (ENS).
HASHING

CONTENT PLACEMENT
- \( m \) balls and \( n \) bins
- each ball chooses a bin uniformly at random
- Goal: avoid collisions.

This is known as the Birthday problem. The probability of no collision is given by

\[
p(n, m) = \left( \frac{n-1}{n} \right) \left( \frac{n-2}{n} \right) \ldots \left( \frac{n-m+1}{n} \right)
\]

\[
\approx \exp \left( -\frac{1 + 2 + \cdots + m - 1}{n} \right)
\]

\[
\approx \exp \left( -\frac{m^2}{2n} \right)
\]

To avoid collision we must have

\[
p(n, m) \approx 1 \iff m << \sqrt{n}.
\]

Load factor \( \rho = \frac{m}{n} \to 0 \) as \( n \to \infty \).
CUCKOO HASHING

Introduced by Pagh & Rodler, ESA’01:

- two bins are assigned at random to each ball
- each ball is placed in one of these two bins
- bins have capacity one, i.e. no collision allowed

Q: How many balls $m$ can you put into $n$ bins with these constraints?
Random graph $G(n, m)$. 
Q: How large can $m$ be so that $G(n, m)$ is still orientable?
Recall that the degree is a $\text{Bin} \left( m, \frac{n-1}{\binom{n}{2}} \right)$ random variable with mean $\frac{2m}{n}$ so that if $2m > n$, there is a giant component:
POSITIVE LOAD FACTOR

Recall that the degree is a $\text{Bin} \left( m, \frac{n-1}{2} \right)$ random variable with mean $\frac{2m}{n}$ so that if $2m > n$, there is a giant component:

For cuckoo hashing with two choices, the critical load factor is $\rho = \frac{1}{2}$.
GENERALIZATIONS

Adding capacities to the bins $k \geq 1$:

Q: $k$-orientation of the random graph $G(n, m)$?

Cain, Sanders, Wormald, Fernholz, Ramachandran SODA’07
GENERALIZATIONS

Adding choices for each ball \( h \geq 1 \):

Q: 1-orientation of the random hypergraph \( H(n, m, h) \)?

Dietzfelbinger, Goerdt, Mitzenmacher, Montanari, Fountoulakis, Panagiotou ICALP’10
Frieze, Melsted, Bordenave, Lelarge, Salez
GENERALIZATIONS

Adding balls $h > \ell \geq 1$ proposed by Gao, Wormald STOC’10:

Case $\ell = 1$ solved by Fountoulakis, Kosha, Panagiotou SODA’11

For large $k$, Gao, Wormald STOC’10: “The full definition of [the critical load factor] is rather complicated, involving the solution of a differential equation system given in (3.4-3.14).”
\[ z_{L,h-j}'(x) = \frac{z_{L,h-j}}{z_L} \left( -1 - \frac{(h-j-1)z_{L,h-j}}{z_{B,h-j}} \right) \]
\[ + \frac{z_{L,h-w+1}}{z_L} \left( \frac{(h-w)z_{H,h-w+1}}{z_{B,h-w+1}} \cdot \frac{(k+1)z_A}{z_B-z_L} \cdot \frac{z_{H,h-j}}{z_B-z_L} \right) \]
\[ + \frac{z_{L,h-j+1}(h-j)z_{L,h-j+1}}{z_{B,h-j+1}} \text{, } j = 1, \ldots, w-1, \]  
(3.4)

\[ z_{H,h-j}'(x) = \frac{z_{L,h-j}}{z_L} \left( -1 - \frac{(h-j-1)z_{H,h-j}}{z_{B,h-j}} \right) \]
\[ - \frac{z_{L,h-w+1}}{z_L} \left( \frac{(h-w)z_{H,h-w+1}}{z_{B,h-w+1}} \cdot \frac{(k+1)z_A}{z_B-z_L} \cdot \frac{z_{H,h-j}}{z_B-z_L} \right) \]
\[ + \frac{z_{L,h-j+1}(h-j)z_{H,h-j+1}}{z_{B,h-j+1}} \text{, } j = 1, \ldots, w-1, \]  
(3.5)

\[ z_L'(x) = -1 + \frac{z_{L,h-w+1}}{z_L} \left( - \frac{(h-w)z_{L,h-w+1}}{z_{B,h-w+1}} + (h-w)k \cdot \frac{z_{H,h-w+1}}{z_B-z_L} \cdot \frac{(k+1)z_A}{z_B-z_L} \right) \]  
(3.6)

\[ z_B'(x) = -1 - \frac{(h-w)z_{L,h-w+1}}{z_L} \]  
(3.7)

\[ z_{HV}'(x) = - \frac{z_{L,h-w+1}}{z_L} \frac{(h-w)z_{H,h-w+1}}{z_{B,h-w+1}} \cdot \frac{(k+1)z_A}{z_B-z_L} \]  
(3.8)

\[ \lambda'(x) = \frac{((z_B' - z_L')z_{HV} - (z_B - z_L)z_{HV}')f_{k+1}(\lambda)}{z_{HV}^2(f_k(\lambda) + \lambda e^{-\lambda} \cdot \frac{\lambda^{k-1}}{(k-1)!} - \frac{z_B - z_L}{z_{HV}} \cdot \frac{e^{-\lambda} \cdot \lambda^k}{k!})} \]  
(3.9)

\[ z_{L,h}(x) = z_L(x) - \sum_{i=1}^{w-1} z_{L,h-j}(x), \quad z_{H,h}(x) = z_B(x) - z_L(x) - \sum_{i=1}^{w-1} z_{H,h-j}(x), \]  
(3.10)

\[ z_{B,h-j}(x) = z_{L,h-j}(x) + z_{H,h-j}(x), \text{ for every } 0 \leq j \leq w-1, \]  
(3.11)

\[ z_{A}(x) = \frac{\lambda(x)^{k+1}}{e^{\lambda(x)}(k+1)!f_{k+1}(\lambda(x))}z_{HV}(x), \]  
(3.12)

where \( f_k(\lambda) \) was defined in (3.1). The initial conditions are
\[ z_B(0) = \bar{\mu}, \quad z_{L,h-j}(0) = 0, \quad z_{H,h-j}(0) = 0, \text{ for all } 1 \leq j \leq w-1, \]  
(3.13)

\[ z_L(0) = \bar{\mu}(1 - f_k(\bar{\mu})), \quad z_{HV}(0) = 1 - \exp(-\bar{\mu}) \sum_{i=0}^{k} \bar{\mu}^i/i!, \quad \lambda(0) = \bar{\mu}. \]  
(3.14)
Allocation is possible (in the large $n$ limit w.h.p.) only if $m = cn$ with $c < c_{h,\ell,k}$ and

$$c_{h,\ell,k} = \frac{\xi^*}{h \mathbb{P}(\text{Bin}(h - 1, 1 - Q(\xi^*, k) < \ell))},$$

where $Q(x, y) = e^{-x} \sum_{j \geq y} \frac{x^j}{j!}$ and $\xi^*$ is the unique solution to:

$$hk = \xi^* \frac{\mathbb{E} \left[ (\ell - \text{Bin}(h, 1 - Q(\xi^*, k)))^+ \right]}{Q(\xi^*, k + 1) \mathbb{P}(\text{Bin}(h - 1, 1 - Q(\xi^*, k)) < \ell)}.$$

Lelarge SODA’12
Critical load \( \frac{\ell c_{h,\ell,k}}{k} \) as a function of \( k = 1 \ldots 10 \) capacity of each bin with:

- \( h = 4 \) choices per batch
- \( \ell = 1, 2, 3 \) balls per batch
Critical load $\frac{\ell c_{h,\ell,k}}{k}$ as a function of $k = 1 \ldots 10$ capacity of each bin with:

- $h = 4, 5, 6$ choices per batch
- $\ell = 2$ balls per batch
HASHING

CONTENT PLACEMENT
- $n$ contents
- $m$ servers, each storing $d$ contents sampled independently (but not uniformly).
- the degree of a content is the number of replicas for this content in the system.
OPTIMAL ALLOCATION
SPANNING SUBGRAPHS OF BIPARTITE RANDOM GRAPHS

- Black nodes = \( n \) bins
- Blue nodes = \( m \) batches of \( \ell \) balls
- Edge = possible choice for the balls of the batch. Each blue node has degree \( h > \ell \).
SPANNING SUBGRAPHS OF BIPARTITE RANDOM GRAPHS

- $n$ black nodes

- $m$ blue nodes of degree $h$

- Allocation = for each blue node, select $\ell$ edges such that in the spanning subgraph, all black nodes have degree less than $k$.

Example with $k = \ell = 2$. 
A COMBINATORIAL DETOUR

A simple identity:

\[
\Omega(G, \lambda, x) = \prod_{vew \in E} (1 + \lambda_e x_v x_w) = \sum_{H \subseteq E} \lambda^H x^{\deg(H)},
\]

with \(\lambda^H = \prod_{e \in H} \lambda_e\) and \(x^{\deg(H)} = \prod_{v \in V} x_v^{\deg(v,H)}\).
A simple identity:

\[ \Omega(G, \lambda, x) = \prod_{vew \in E} (1 + \lambda_e x_v x_w) = \sum_{H \subseteq E} \lambda^H x^{\text{deg}(H)}, \]

with \( \lambda^H = \prod_{e \in H} \lambda_e \) and \( x^{\text{deg}(H)} = \prod_{v \in V} x_v^{\text{deg}(v,H)} \).

We are interested in:

\[ Z(G, \lambda, x) = \sum_{H \subseteq E} \lambda^H x^{\text{deg}(H)} \mathbb{I}(H \text{ is a matching}) \]
If $P(z) = \sum_{j=0}^{d} c_j z^j$ is nonvanishing in the open right half-plane and $K(z) = \sum_{j=0}^{d} \binom{d}{j} u_j z^j$ has only real nonpositive zeros, then $Q(z) = \sum_{j=0}^{d} u_j c_j z^j$ is nonvanishing in the open right half-plane.
Consider the case \( u_0 = u_1 = 1 \) and \( u_k = 0 \) for \( k \geq 2 \) and define
\[
K_v(z) = 1 + \deg(v)z.
\]
Let \( F_0(x) = \Omega(G, \lambda, x) \) and define \( F_v(x) \) as the Schur-Szegö composition of \( F_{v-1}(x_v) \) and \( K_v(x_v) \). (Wagner 2009)

\[
F_0(x) = \sum_{H \subseteq E} \lambda^H x^{\deg(H)}
\]

\[
F_1(x) = \sum_{H \subseteq E} \lambda^H \mathbb{I}(\deg(v, H) \leq 1) x^{\deg(H)}
\]

\[\vdots\]

\[
F_n(x) = \sum_{H \subseteq E} \lambda^H \prod_{v=1}^n \mathbb{I}(\deg(v, H) \leq 1) x^{\deg(H)}
\]

\[
= \sum_{H \subseteq E} \lambda^H x^{\deg(H)} \mathbb{I}(H \text{ is a matching}) = Z(G, \lambda, x).
\]
ANALOGY WITH STATISTICAL PHYSICS

\[ Z(G, 1, z^{1/2} \mathbf{1}) = \sum_M z^{|M|} = \sum_k m_k(G) z^k = P_G(z), \] where \( m_k(G) \) is the number of \( k \)-edge matchings of \( G \).

The fact that \( P_G(z) \) has its zeros on the negative real axis allows to define the Gibbs measure

\[ \mu_G^z(M) = \frac{z^{|M|}}{P_G(z)} \]

on infinite graphs (as an 'analytic' limit) = absence of phase transitions.

(Heilmann Lieb 1972)
ANALOGY WITH STATISTICAL PHYSICS

\[ Z(G, 1, z^{1/2} 1) = \sum_M z^{|M|} = \sum_k m_k(G) z^k = P_G(z), \]
where \( m_k(G) \) is the number of \( k \)-edge matchings of \( G \).

The fact that \( P_G(z) \) has its zeros on the negative real axis allows to define the Gibbs measure

\[
\mu_G^z(M) = \frac{z^{|M|}}{P_G(z)}
\]

on infinite graphs (as an 'analytic' limit) = absence of phase transitions.

(Heilmann Lieb 1972)

This technique can be used as a step towards computations BUT it fails for more general spanning subgraphs, i.e. for degree constraints larger than 3.
For simplicity, spanning subgraph $H$ with $\deg(v, H) \leq 2 = w$. 
A SIMPLE GREEDY ALGORITHM ON TREES

For simplicity, spanning subgraph $H$ with $\deg(v, H) \leq 2 = w$. 
A SIMPLE GREEDY ALGORITHM ON TREES

For simplicity, spanning subgraph $H$ with $\text{deg}(v, H) \leq 2 = w$. 
A SIMPLE GREEDY ALGORITHM ON TREES

For simplicity, spanning subgraph $H$ with $\deg(v, H) \leq 2 = w$. 
For simplicity, spanning subgraph $H$ with $\deg(v, H) \leq 2 = w$. 

A SIMPLE GREEDY ALGORITHM ON TREES
A MESSAGE PASSING VERSION OF THE GREEDY ALGORITHM

Black arrow: 'I want to match you'
A MESSAGE PASSING VERSION OF THE GREEDY ALGORITHM

Black arrow: 'I want to match you'

Red arrow: 'Sorry, I am saturated'
A MESSAGE PASSING VERSION OF THE GREEDY ALGORITHM

Black arrow: 'I want to match you'

Red arrow: 'Sorry, I am saturated'
Replace black arrows by 1 messages and red arrows by 0 messages and run simultaneously.

For any directed edge, sum the incoming messages from the other edges. If this sum is larger than $w = 2$ then $\mathcal{P}_G$ returns 0, otherwise returns 1 on this directed edge.
A MESSAGE PASSING VERSION OF THE GREEDY ALGORITHM

\[ I_2 = \mathcal{P}_G(I_1) \]

Iterate...
A MESSAGE PASSING VERSION OF THE GREEDY ALGORITHM

\[ I_3 = \mathcal{P}_G \circ \mathcal{P}_G(I_1) \]

... until you get a fixed point \( I^* \).
A MESSAGE PASSING VERSION OF THE GREEDY ALGORITHM

On finite trees, the algorithm converges and $I^*$ allows to get the size of a maximum spanning subgraph.

$$\sum_{v \in V} \left( w \mathbb{1} \left( \sum_{\partial v} I^*_\partial \geq w + 1 \right) + \frac{1}{2} \mathbb{1} \left( \sum_{\partial v} I^*_\partial \leq w \right) \sum_{\partial v} I^*_\partial \right)$$
RUNNING THE ALGORITHM ON AN INFINITE TREE

Let simplify further $\ell = k = 1$ and Poisson Galton-Watson tree with mean offspring $\lambda$.

- Let $p$ be the probability of sending a 1 message

$$p = \mathbb{P}(I^*_{\ell} = 1)$$

- Thanks to the branching property:

$$p = \mathbb{P} \text{ (no children send a 1 message)} = e^{-\lambda p}$$

and so $p = \frac{W(\lambda)}{\lambda}$. 
The function $\frac{W(\lambda)}{\lambda}$ as a function of $\lambda$. 
The true value of $p$ as a function of $\lambda$. 

TRUTH
Let $p_k$ be the probability of the root sending message 1 for the tree truncated at depth $k$.

- $p_0 = 1$
- $p_1 = e^{-\lambda}$
- then for $k \geq 0$

$$p_{k+1} = e^{-\lambda p_k}$$

We computed the fixed point of the map $p \mapsto e^{-\lambda p}$ but the truth is given by iterating it...
 ITERATING

\[ \lambda = 2.5 \]
\[ \lambda = 2.5 \]
\[ \lambda = 2.9 \]
$\lambda = 2.9$
Influence of the boundary conditions remains positive.
- Introduce the Gibbs measure on allocations:

\[ \mu^z_G(B) = \frac{z^{\sum_e B_e}}{P_G(z)} \]

so that the size of a maximum allocation of the graph \( G = (V, E) \) is given by

\[ \frac{1}{2} \lim_{z \to \infty} \sum_{v \in V} \sum_{e \in \partial v} \mu^z_G(B_e = 1). \]
- Introduce the Gibbs measure on allocations:

$$\mu^z_G(\mathbf{B}) = \frac{z \sum_e B_e}{P_G(z)}$$

so that the size of a maximum allocation of the graph $G = (V, E)$ is given by

$$\frac{1}{2} \lim_{z \to \infty} \sum_{v \in V} \sum_{e \in \partial v} \mu^z_G(B_e = 1).$$

- Show that on trees, the marginal $\mu^z_G(B_e = 1)$ can be computed by a message passing algorithm with a unique fixed point.
MESSAGE PASSING ALGORITHM

Define $Y_e(z) \in \mathbb{R}$ by $\mu_{G,e}^z(B_e = 1) = \frac{Y_e(z)}{1 + Y_e(z)}$. Then the recursion is

$$Y^{t+1}_e(z) = z R_G(Y^t(z))$$

with

$$R_e(Y) = \frac{\sum_{S \prec e, |S| \leq w-1} \prod_{f \in S} Y_f}{\sum_{S \prec e, |S| \leq w} \prod_{f \in S} Y_f}.$$
Define $Y_e(z) \in \mathbb{R}$ by $\mu_{G,e}^z(B_e = 1) = \frac{Y_e(z)}{1+Y_e(z)}$. Then the recursion is

$$Y^{t+1}(z) = zR_G(Y^t(z))$$

with

$$R_e(Y) = \frac{\sum_{S \prec e, |S| \leq w-1} \prod_{f \in S} Y_f}{\sum_{S \prec e, |S| \leq w} \prod_{f \in S} Y_f}.$$ 

In the case of matchings, $w = 1$ so that

$$R_e(Y) = \frac{1}{1 + \sum_{f \prec e} Y_f}.$$
- Introduce the Gibbs measure on allocations:

\[ \mu^z_G(B) = \frac{z \sum_e B_e}{P_G(z)} \]

so that the size of a maximum allocation of the graph \( G = (V, E) \) is given by

\[ \frac{1}{2} \lim_{z \to \infty} \sum_{v \in V} \sum_{e \in \partial v} \mu^z_G(B_e = 1). \]

- Show that on trees, the marginal \( \mu^z_G(B_e = 1) \) can be computed by a message passing algorithm with a unique fixed point.

- Show that on trees, when \( z \to \infty \), this message passing algorithm reduces to the previously described 0 – 1 valued message passing algorithm and that the limit of \( \mu^z_G(B_e = 1) \) can be computed from the minimal fixed point solution.
BYPASSING CORRELATION DECAY

- Introduce the Gibbs measure on allocations:

\[ \mu^z_G(B) = \frac{z \sum_e B_e}{P_G(z)} \]

so that the size of a maximum allocation of the graph \( G = (V, E) \) is given by

\[ \frac{1}{2} \lim_{z \to \infty} \sum_{v \in V} \sum_{e \in \partial v} \mu^z_G(B_e = 1). \]

- Show that on trees, the marginal \( \mu^z_G(B_e = 1) \) can be computed by a message passing algorithm with a unique fixed point.

- Show that on trees, when \( z \to \infty \), this message passing algorithm reduces to the previously described 0 – 1 valued message passing algorithm and that the limit of \( \mu^z_G(B_e = 1) \) can be computed from the minimal fixed point solution.

- Using a convexity argument, invert the limits in \( n \) and \( z \).
RESULT ON INFINITE UNIMODULAR TREES

Assumption: $G_n$ has random weak limit $\rho ([G, \circ])$, a unimodular probability measure concentrated on trees.

For any $I \in \{0, 1\}^\mathbb{E}$,

$$F_\circ(I) = w_\circ \mathbb{I}(\sum_{x \in \partial_\circ} P_{x \to \circ}(I) \geq w_\circ + 1) + w_\circ \wedge \sum_{x \in \partial_\circ} I_{x \to \circ}.$$  

Then

$$\lim_{n \to \infty} \frac{1}{n} M(G_n) = \frac{1}{2} \inf \left\{ \int F_\circ(I) d\rho([G, \circ]) \right\},$$

where the infimum is over all spatially invariant solutions of $I = \mathcal{P}_G \circ \mathcal{P}_G(I)$. 


ON GALTON-WATSON TREES

For matchings, the Recursive Distributional Equation (RDE) becomes:

\[
Y(z) \overset{d}{=} \frac{z}{1 + \sum_{i=1}^{N} Y_i(z)}
\]

where \( N \sim \) the standard size biased degree distribution of the random graph.

By iterating once

\[
\frac{Y(z)}{z} \overset{d}{=} \frac{1}{1 + \sum_{i=1}^{N} \frac{1}{z + \sum_{j=1}^{N \cap i,j} Y_{ij}(z)}}
\]

so that we obtain for \( X = \lim_{z \to \infty} \frac{Y(z)}{z} \in [0, 1] \) the simple RDE:

\[
X \overset{d}{=} \frac{1}{1 + \sum_{i=1}^{N} \frac{1}{\sum_{j=1}^{N \cap i,j} X_{ij}}}
\]
SOLVING THE RDE AT $z = \infty$

If $\varphi$ is the generating function of the asymptotic degree distribution, let

$$G(x) = \varphi'(1)x\overline{x} + \varphi(1 - x) + \varphi(1 - \overline{x}) - 1,$$

where $\overline{x} = \varphi'(1 - x)/\varphi'(1)$.

$G$ admits an historical record at $x$ if $x = \overline{x}$ and $G(x) > G(y)$ for any $0 \leq y < x$.

**Theorem 1.** If $p_1 < \ldots < p_r$ are the locations of the historical records of $G$, then the RDE admits exactly $r$ solutions, say $0 \leq X_1 < \ldots < X_r \leq 1$, and for any $i \in \{1, \ldots, r\}$, $\mathbb{E}[X_i] = G(p_i)$ and $\mathbb{P}(X_i > 0) = p_i$.

From the values $p_1 < \ldots < p_r$, we can compute the limit of the matching number (rescaled by $n$) when $n \to \infty$. 
CONCLUSION

- General method to compute law of large numbers for combinatorial structures on sparse (random) graphs.

  (a) to bypass the correlation decay, add a (small) noise parameter.

  (b) crucially use monotonicity of the recursions

- Our method works for matchings, spanning subgraphs with degree constraints and $b$-matchings.

- The absence of phase transition has also algorithmic implications: sublinear algorithms to approximate the number of matchings.

- Open problem: Counting of other large subgraphs: long cycles (Marinari & Semerjian 2006).
THANK YOU!