Counting Solutions to Random Constraint Satisfaction Problems

Allan Sly, UC Berkeley

Joint work with Nike Sun and Yumeng Zhang

Simons Institute May 2

Introduction:
random constraint satisfaction problems;

Combinatorics and Theoretical Computer Science

Constraint satisfaction problem (CSP): is it possible to assign values to a set of *variables* to satisfy a given set of *constraints*?

- System of linear equations.
- Colouring a graph or finding a large independent set.
- Satisfying a Boolean formula.

Combinatorics and Theoretical Computer Science

Constraint satisfaction problem (CSP): is it possible to assign values to a set of *variables* to satisfy a given set of *constraints*?

- System of linear equations.
- Colouring a graph or finding a large independent set.
- Satisfying a Boolean formula.

A subclass of CSPs is *NP-complete*; these have a central role in the theory of computational complexity

Combinatorics and Theoretical Computer Science

Constraint satisfaction problem (CSP): is it possible to assign values to a set of *variables* to satisfy a given set of *constraints*?

- System of linear equations.
- Colouring a graph or finding a large independent set.
- Satisfying a Boolean formula.

A subclass of CSPs is *NP-complete*; these have a central role in the theory of computational complexity

Theoretical Physics

Disordered systems such as *spin glasses* are models of interacting particles/variables with frustrated interactions.
Many random constraint satisfaction problems can be recast as dilute mean-field spin glasses.

K-SAT The random K-SAT problem, a model of a random Boolean formula, is perhaps the canonical random CSP.

K-SAT The random K-SAT problem, a model of a random Boolean formula, is perhaps the canonical random CSP.

Basic Definition:

Variables: $x_1, \ldots, x_n \in \{\text{TRUE}, \text{FALSE}\} \equiv \{+, -\}$

Constraints: m clauses taking the OR of k variables uniformly chosen from $\{+x_1, -x_1, \ldots, +x_n, -x_n\}$.

K-SAT The random K-SAT problem, a model of a random Boolean formula, is perhaps the canonical random CSP.

Basic Definition:

Variables: $x_1, \ldots, x_n \in \{\text{TRUE}, \text{FALSE}\} \equiv \{+, -\}$

Constraints: m clauses taking the OR of k variables uniformly chosen from $\{+x_1, -x_1, \ldots, +x_n, -x_n\}$.

Example: A 3-SAT formula with 4 clauses:

$$\mathcal{G}(\underline{x}) = (+x_1 \text{ OR } +x_2 \text{ OR } -x_3) \text{ AND } \overbrace{(+x_3 \text{ OR } +x_4 \text{ OR } -x_5)}^{\text{clause}}$$
$$\text{AND } (-x_1 \text{ OR } -x_4 \text{ OR } +x_5) \text{ AND } (+x_2 \text{ OR } -x_3 \text{ OR } +x_4)$$

K-SAT The random K-SAT problem, a model of a random Boolean formula, is perhaps the canonical random CSP.

Basic Definition:

Variables: $x_1, \ldots, x_n \in \{\text{TRUE}, \text{FALSE}\} \equiv \{+, -\}$

Constraints: m clauses taking the OR of k variables uniformly chosen from $\{+x_1, -x_1, \ldots, +x_n, -x_n\}$.

Example: A 3-SAT formula with 4 clauses:

$$\mathcal{G}(\underline{x}) = (+x_1 \text{ OR } +x_2 \text{ OR } -x_3) \text{ AND } \overbrace{(+x_3 \text{ OR } +x_4 \text{ OR } -x_5)}^{\text{clause}}$$
$$\text{AND } (-x_1 \text{ OR } -x_4 \text{ OR } +x_5) \text{ AND } (+x_2 \text{ OR } -x_3 \text{ OR } +x_4)$$

Clause density: The K-SAT model is parameterized the problem by the density of clauses $\alpha = m/n$.

Graphical description: We can encode a K-SAT formula as a bipartite graph:

Graphical description: We can encode a K-SAT formula as a bipartite graph:

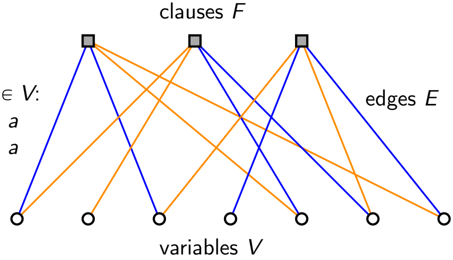
Take a 4-SAT formula with 3 clauses: $\mathcal{G}(\underline{x}) =$

$$(+x_1 \text{ OR } +x_3 \text{ OR } -x_5 \text{ OR } -x_7) \text{ AND } (-x_1 \text{ OR } -x_2 \text{ OR } +x_5 \text{ OR } +x_6)$$
$$\text{AND } (-x_3 \text{ OR } +x_4 \text{ OR } -x_6 \text{ OR } +x_7)$$

Graphical description: We can encode a K-SAT formula as a bipartite graph:
Take a 4-SAT formula with 3 clauses: $\mathcal{G}(\underline{x}) =$
$(+x_1 \text{ OR } +x_3 \text{ OR } -x_5 \text{ OR } -x_7) \text{ AND } (-x_1 \text{ OR } -x_2 \text{ OR } +x_5 \text{ OR } +x_6)$
$\text{AND } (-x_3 \text{ OR } +x_4 \text{ OR } -x_6 \text{ OR } +x_7)$

We can encode the formula as a bipartite graph $\mathcal{G} \equiv (V, F, E)$:

clause $a \in F$, variable $v \in V$:
blue edge *(av)* if $+x_v$ in clause a
yellow edge *(av)* if $-x_v$ in clause a



(4-SAT: each clause has degree 4)

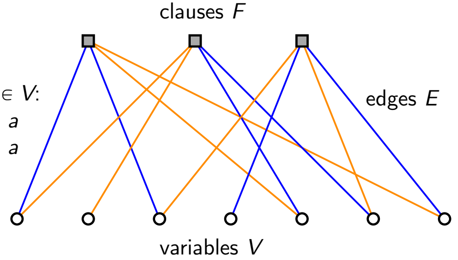
Graphical description: We can encode a K-SAT formula as a bipartite graph:

Take a 4-SAT formula with 3 clauses: $\mathcal{G}(\underline{x}) =$

$$(+x_1 \text{ OR } +x_3 \text{ OR } -x_5 \text{ OR } -x_7) \text{ AND } (-x_1 \text{ OR } -x_2 \text{ OR } +x_5 \text{ OR } +x_6)$$
$$\text{AND } (-x_3 \text{ OR } +x_4 \text{ OR } -x_6 \text{ OR } +x_7)$$

We can encode the formula as a bipartite graph $\mathcal{G} \equiv (V, F, E)$:

clause $a \in F$, variable $v \in V$:
blue edge *(av)* if $+x_v$ in clause a
yellow edge *(av)* if $-x_v$ in clause a



(4-SAT: each clause has degree 4)

The resulting random graph is locally tree-like, almost no short cycles and it's local distribution can be described completely.

Initial Question:

- *Satisfiability Threshold*: For which α are there satisfying assignments?

Initial Question:

- *Satisfiability Threshold*: For which α are there satisfying assignments?

Further Question:

- *Free Energy*: How many solutions are there?

$$\Phi(\alpha) \equiv \lim_{n\to\infty} \frac{1}{n} \log Z.$$

- *Local Statistics*: Properties of solutions such as how many clauses are satisfied only once?
- *Algorithmic*: Can solutions be found efficiently?

Initial Question:

- *Satisfiability Threshold*: For which α are there satisfying assignments?

Further Question:

- **Free Energy: How many solutions are there?**

$$\Phi(\alpha) \equiv \lim_{n\to\infty} \frac{1}{n} \log Z.$$

- *Local Statistics*: Properties of solutions such as how many clauses are satisfied only once?
- *Algorithmic*: Can solutions be found efficiently?

Not-All-Equal SAT Model

A k-NAESAT problem is a k-SAT where both x and $-x$ are satisfying assignments. Each clause contains one + and one –.

clause of width $k = 4$

$$
\begin{aligned}
& (+x_1 \text{ OR } +x_3 \text{ OR } -x_5 \text{ OR } -x_7) \\
\text{AND } & (-x_1 \text{ OR } -x_2 \text{ OR } +x_5 \text{ OR } +x_6) \\
\text{AND } & (-x_3 \text{ OR } +x_4 \text{ OR } -x_6 \text{ OR } +x_7)
\end{aligned}
$$

Not-All-Equal SAT Model

A k-NAESAT problem is a k-SAT where both x and $-x$ are satisfying assignments. Each clause contains one + and one −.

$$\overbrace{\left(+x_1 \text{ OR } +x_3 \text{ OR } -x_5 \text{ OR } -x_7 \right)}^{\text{clause of width } k=4}$$
$$\text{AND } \left(-x_1 \text{ OR } -x_2 \text{ OR } +x_5 \text{ OR } +x_6 \right)$$
$$\text{AND } \left(-x_3 \text{ OR } +x_4 \text{ OR } -x_6 \text{ OR } +x_7 \right)$$

d-**Regular** NAESAT is an instance where every variable appears in exactly d clauses.

Not-All-Equal SAT Model

A k-NAESAT problem is a k-SAT where both x and $-x$ are satisfying assignments. Each clause contains one + and one −.

clause of width $k = 4$

$$\begin{aligned} &(+x_1 \text{ OR } +x_3 \text{ OR } -x_5 \text{ OR } -x_7) \\ \text{AND } &(-x_1 \text{ OR } -x_2 \text{ OR } +x_5 \text{ OR } +x_6) \\ \text{AND } &(-x_3 \text{ OR } +x_4 \text{ OR } -x_6 \text{ OR } +x_7) \end{aligned}$$

d-**Regular** NAESAT is an instance where every variable appears in exactly d clauses.
Why regular NAESAT?

Not-All-Equal SAT Model

A k-NAESAT problem is a k-SAT where both x and $-x$ are satisfying assignments. Each clause contains one + and one −.

clause of width $k = 4$

$$
\begin{aligned}
& (+x_1 \text{ OR } +x_3 \text{ OR } -x_5 \text{ OR } -x_7) \\
\text{AND } & (-x_1 \text{ OR } -x_2 \text{ OR } +x_5 \text{ OR } +x_6) \\
\text{AND } & (-x_3 \text{ OR } +x_4 \text{ OR } -x_6 \text{ OR } +x_7)
\end{aligned}
$$

d-**Regular** NAESAT is an instance where every variable appears in exactly d clauses.

Why regular NAESAT? Same rich set of in phase transitions.

Not-All-Equal SAT Model

A k-NAESAT problem is a k-SAT where both x and $-x$ are satisfying assignments. Each clause contains one + and one –.

$$\overbrace{\left(+x_1 \text{ OR } +x_3 \text{ OR } -x_5 \text{ OR } -x_7 \right)}^{\text{clause of width } k=4}$$
$$\text{AND } \left(-x_1 \text{ OR } -x_2 \text{ OR } +x_5 \text{ OR } +x_6 \right)$$
$$\text{AND } \left(-x_3 \text{ OR } +x_4 \text{ OR } -x_6 \text{ OR } +x_7 \right)$$

d-**Regular** NAESAT is an instance where every variable appears in exactly d clauses.

Why regular NAESAT? Same rich set of in phase transitions.

Binary, symmetric, locally homogeneous.

First moment and Second moment

We are interested in $\textbf{SOL} \equiv \{\text{solutions of } \mathcal{G}\}$ and $Z \equiv |\textbf{SOL}|$.

First moment and Second moment

We are interested in **SOL** $\equiv$ {solutions of $\mathcal{G}$} and $Z \equiv |\textbf{SOL}|$.

Keep k fixed and let $\alpha \equiv d/k$.

First moment method:

$$\mathbb{E}Z = 2^n(1 - 2/2^k)^m$$

First moment and Second moment

We are interested in **SOL** $\equiv$ {solutions of $\mathcal{G}$} and $Z \equiv |\textbf{SOL}|$.

Keep k fixed and let $\alpha \equiv d/k$.

First moment method:

$$\mathbb{E}Z = 2^n(1-2/2^k)^m = \exp\{n\underbrace{[\ln 2 + \alpha \log(1-2/2^k)]}\}$$

exponent decreases in α, crosses zero at $\alpha_1 \approx (2^{k-1} - \frac{1}{2})\ln 2$.

First moment and Second moment

We are interested in **SOL** $\equiv \{\text{solutions of } \mathcal{G}\}$ and $Z \equiv |\textbf{SOL}|$.

Keep k fixed and let $\alpha \equiv d/k$.

First moment method:

$$\mathbb{E}Z = 2^n(1-2/2^k)^m = \exp\{n \underbrace{[\ln 2 + \alpha \log(1-2/2^k)]}\}$$

exponent decreases in α, crosses zero at $\alpha_1 \approx (2^{k-1} - \frac{1}{2}) \ln 2$.

Second moment method:

If $\mathbb{E}[Z^2] \asymp (\mathbb{E}Z)^2$, then $\mathbb{P}[Z > \varepsilon \mathbb{E}Z]$ bounded away from 0.

Fails at $\alpha_2 \approx 2^{k-1} \ln 2 - \frac{1}{2}(\ln 2 + 1) \approx \alpha_1 - \frac{1}{2}$. Achlioptas–Moore '06

First moment and Second moment

We are interested in **SOL** $\equiv$ {solutions of $\mathcal{G}$} and $Z \equiv |\textbf{SOL}|$.

Keep k fixed and let $\alpha \equiv d/k$.

First moment method:

$$\mathbb{E}Z = 2^n(1-2/2^k)^m = \exp\{n \underbrace{[\ln 2 + \alpha \log(1-2/2^k)]}\}$$

exponent decreases in α, crosses zero at $\alpha_1 \approx (2^{k-1} - \frac{1}{2}) \ln 2$.

Second moment method:

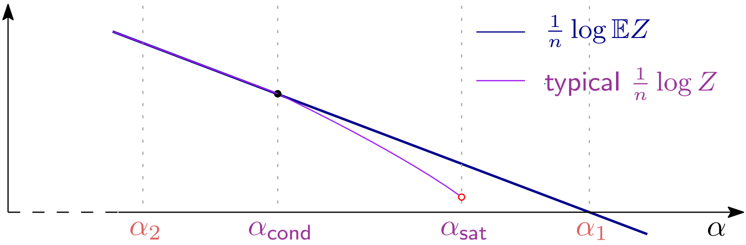
If $\mathbb{E}[Z^2] \asymp (\mathbb{E}Z)^2$, then $\mathbb{P}[Z > \varepsilon \mathbb{E}Z]$ bounded away from 0.

Fails at $\alpha_2 \approx 2^{k-1} \ln 2 - \frac{1}{2}(\ln 2 + 1) \approx \alpha_1 - \frac{1}{2}$. Achlioptas–Moore '06

This α_2 can be improved, but not all the way to α_1.

Coja-Oghlan–Zdeborová '12

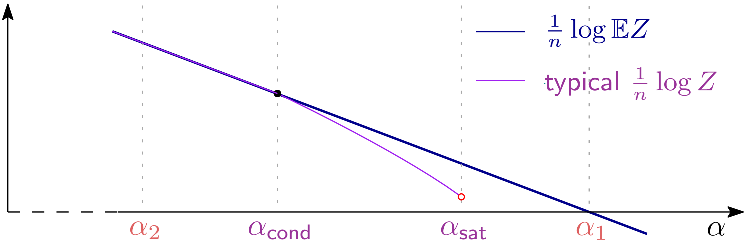
Non-concentration of Z



In fact there exist $\alpha_2 < \alpha_{\text{cond}} < \alpha_{\text{sat}} < \alpha_1$ such that:

$$\begin{cases} \log Z = \log \mathbb{E}Z + O_p(1) & \alpha < \alpha_{\text{cond}} \\ \log Z < \log \mathbb{E}Z - \Omega(n) & \alpha_{\text{cond}} < \alpha < \alpha_{\text{sat}} \\ \mathbb{P}(Z = 0) \to 1 & \alpha > \alpha_{\text{sat}} \end{cases}$$

Non-concentration of Z



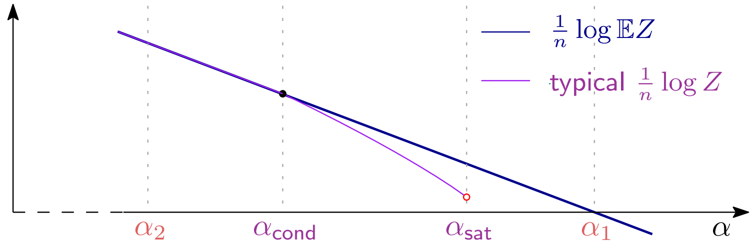
In fact there exist $\alpha_2 < \alpha_{\text{cond}} < \alpha_{\text{sat}} < \alpha_1$ such that:

$$\begin{cases} \log Z = \log \mathbb{E}Z + O_p(1) & \alpha < \alpha_{\text{cond}} \\ \log Z < \log \mathbb{E}Z - \Omega(n) & \alpha_{\text{cond}} < \alpha < \alpha_{\text{sat}} \\ \mathbb{P}(Z = 0) \to 1 & \alpha > \alpha_{\text{sat}} \end{cases}$$

— $\mathbb{E}Z$ fails to describe Z for $\alpha \geqslant \alpha_{\text{cond}}$.

Coja-Oghlan–Zdeborová '12, Ding–Sly–Sun '13a

Non-concentration of Z

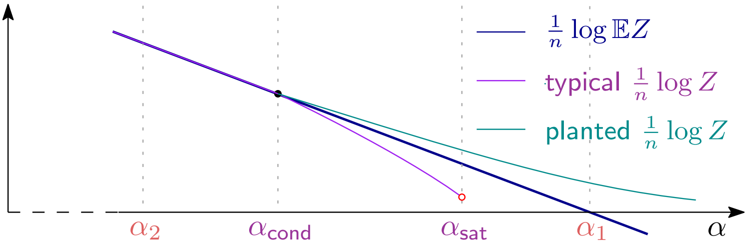


First explanation:

Typically, any solution $\underline{x}$ of $\mathcal{G}$ has $\geqslant n\epsilon$ *free variables*, that can flip without violating any clause.

— $\mathbb{E}Z$ is dominated by unusual **cluster of solutions** of size $\geqslant 2^{n\epsilon}$.

Non-concentration of Z

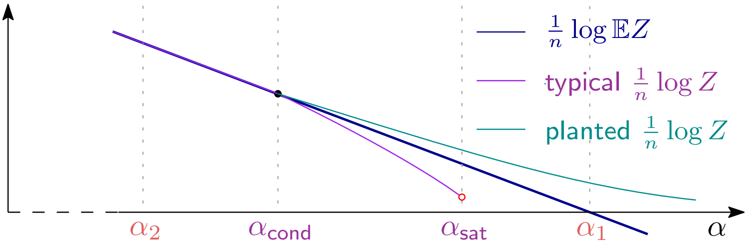


First explanation:

Typically, any solution $\underline{x}$ of $\mathcal{G}$ has $\geqslant n\epsilon$ *free variables*, that can flip without violating any clause.

— $\mathbb{E}Z$ is dominated by unusual **cluster of solutions** of size $\geqslant 2^{n\epsilon}$.

Non-concentration of Z

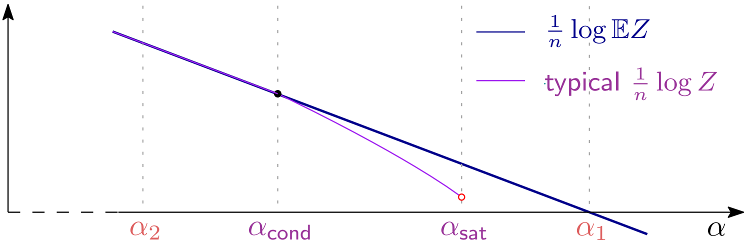


First explanation:

Typically, any solution $\underline{x}$ of $\mathcal{G}$ has $\geqslant n\epsilon$ *free variables*, that can flip without violating any clause.
— $\mathbb{E}Z$ is dominated by unusual **cluster of solutions** of size $\geqslant 2^{n\epsilon}$.

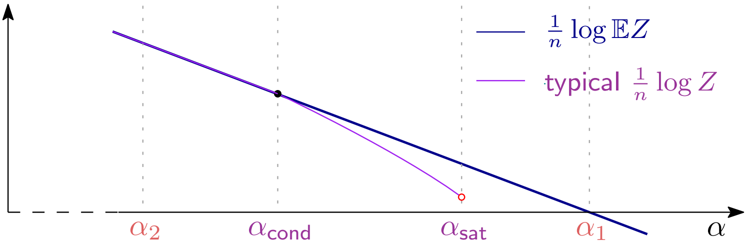
Deeper reason: 1RSB Theory from statistical physics.

Main result



Free energy: $\Phi(\alpha) \equiv \lim_{n\to\infty} \frac{1}{n} \log Z$.

Main result



Free energy: $\Phi(\alpha) \equiv \lim_{n\to\infty} \frac{1}{n} \log Z.$

Main result: For $k \geqslant k_0$, the limit $\Phi(\alpha)$ does exist for $\alpha_{\text{cond}} < \alpha < \alpha_{\text{sat}}$, and we give an explicit formula matching the 1RSB prediction from statistical physics.

Physicist's Prediction: Condensation and Replica Symetry Breaking

Statistical physics for random CSPs

Statistical physicists made major advances in this field by showing how to adapt heuristics from the study of spin glasses (disordered magnets) to explain the CSP solution space.

Mézard–Parisi '85, '86, '87; Fu–Anderson '86

Statistical physics for random CSPs

Statistical physicists made major advances in this field by showing how to adapt heuristics from the study of spin glasses (disordered magnets) to explain the CSP solution space.

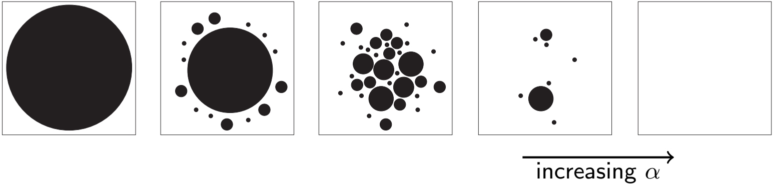
Mézard–Parisi '85, '86, '87; Fu–Anderson '86

In particular, physicists proposed a class of sparse random CSPs — the *one-step replica symmetry breaking* (1RSB) models, which exhibit the similar phase diagram at predicted locations.

Krząkała–Montanari–Ricci-Tersenghi–Semerjian–Zdeborová '07, Zdeborová–Krząkała '07, Montanari–Ricci-Tersenghi–Semerjian '08.

Phase diagram

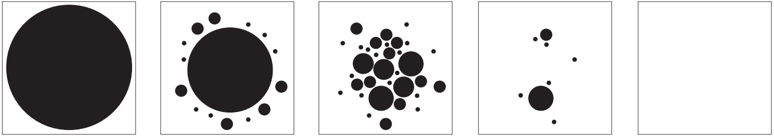
Two solutions are connected if they differ by one bit.



KMRSZ '07, MRS '08

Phase diagram

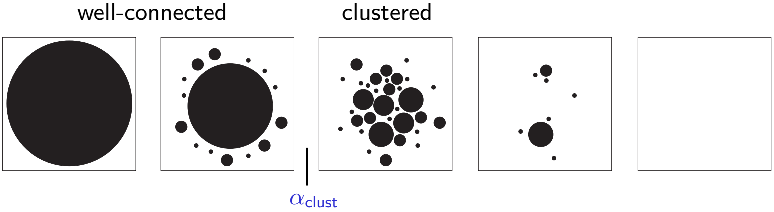
well-connected



KMRSZ '07, MRS '08

The solution space **SOL** starts out as a well-connected cluster.

Phase diagram

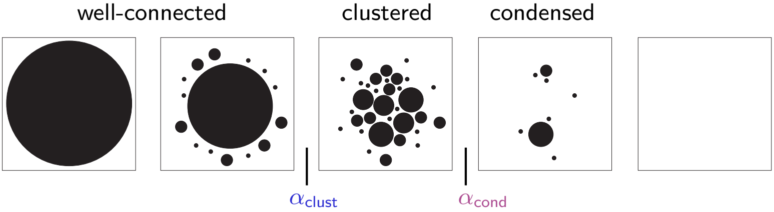


KMRSZ '07, MRS '08

The solution space **SOL** starts out as a well-connected cluster.

After α_{clust}, **SOL** decomposes into exponentially clusters

Phase diagram



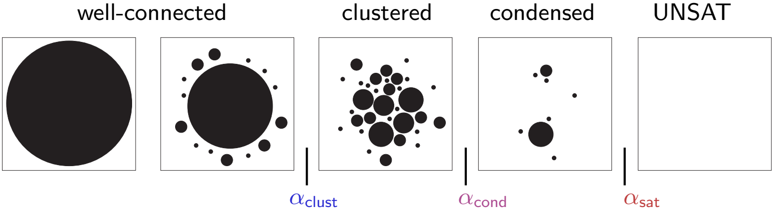
KMRSZ '07, MRS '08

The solution space **SOL** starts out as a well-connected cluster.

After α_{clust}, **SOL** decomposes into exponentially clusters

After α_{cond}, **SOL** is dominated by a few large clusters

Phase diagram



KMRSZ '07, MRS '08

The solution space **SOL** starts out as a well-connected cluster.

After α_{clust}, **SOL** decomposes into exponentially clusters

After α_{cond}, **SOL** is dominated by a few large clusters

After α_{sat}, no solutions w.h.p.

Condensation

Complexity function $\Sigma \equiv \Sigma_\alpha(s)$ such that:

$$\mathbb{E}Z = \sum \underbrace{(\text{cluster size})}_{\exp\{ns\}} \times \underbrace{\mathbb{E}[\text{number of clusters of that size}]}_{\exp\{n\Sigma(s)\}}$$

Condensation

Complexity function $\Sigma \equiv \Sigma_\alpha(s)$ such that:

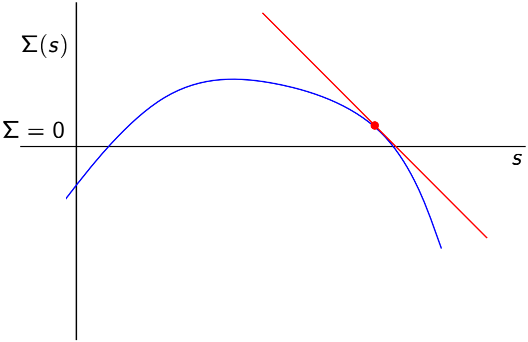
$$\mathbb{E}Z = \sum \underbrace{(\text{cluster size})}_{\exp\{ns\}} \times \underbrace{\mathbb{E}[\text{number of clusters of that size}]}_{\exp\{n\Sigma(s)\}}$$

Condensation

Complexity function $\Sigma \equiv \Sigma_\alpha(s)$ such that:

$$\mathbb{E}Z = \sum \underbrace{(\text{cluster size})}_{\exp\{ns\}} \times \underbrace{\mathbb{E}[\text{number of clusters of that size}]}_{\exp\{n\Sigma(s)\}}$$

$\mathbb{E}Z$ is dominated by s where $\Sigma'(s) \equiv -1$ (depending on α).

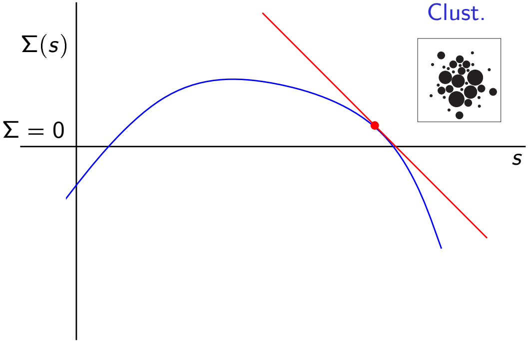


Condensation

Complexity function $\Sigma \equiv \Sigma_\alpha(s)$ such that:

$$\mathbb{E}Z = \sum \underbrace{(\text{cluster size})}_{\exp\{ns\}} \times \underbrace{\mathbb{E}[\text{number of clusters of that size}]}_{\exp\{n\Sigma(s)\}}$$

$\mathbb{E}Z$ is dominated by s where $\Sigma'(s) \equiv -1$ (depending on α).

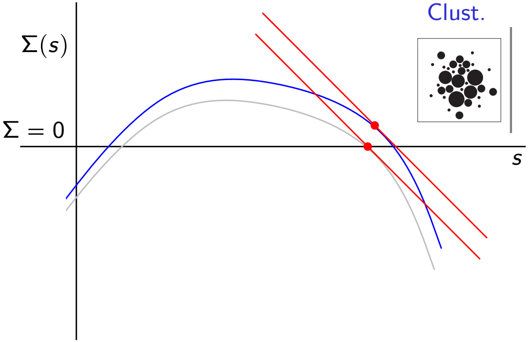


Condensation

Complexity function $\Sigma \equiv \Sigma_\alpha(s)$ such that:

$$\mathbb{E}Z = \sum \underbrace{(\text{cluster size})}_{\exp\{ns\}} \times \underbrace{\mathbb{E}[\text{number of clusters of that size}]}_{\exp\{n\Sigma(s)\}}$$

$\mathbb{E}Z$ is dominated by s where $\Sigma'(s) \equiv -1$ (depending on α).

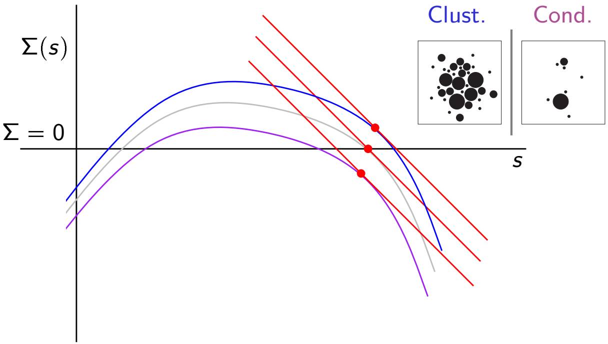


Condensation

Complexity function $\Sigma \equiv \Sigma_\alpha(s)$ such that:

$$\mathbb{E}Z = \sum \underbrace{(\text{cluster size})}_{\exp\{ns\}} \times \underbrace{\mathbb{E}[\text{number of clusters of that size}]}_{\exp\{n\Sigma(s)\}}$$

$\mathbb{E}Z$ is dominated by s where $\Sigma'(s) \equiv -1$ (depending on α).

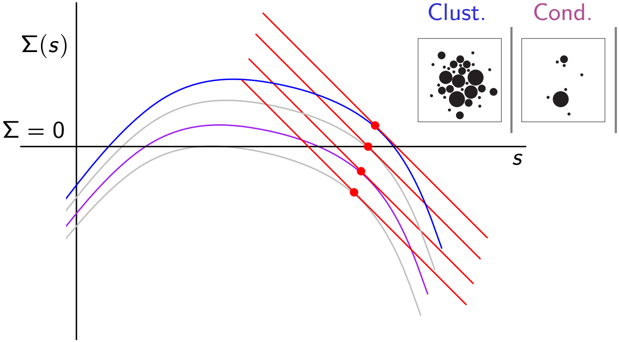


Condensation

Complexity function $\Sigma \equiv \Sigma_\alpha(s)$ such that:

$$\mathbb{E}Z = \sum \underbrace{(\text{cluster size})}_{\exp\{ns\}} \times \underbrace{\mathbb{E}[\text{number of clusters of that size}]}_{\exp\{n\Sigma(s)\}}$$

$\mathbb{E}Z$ is dominated by s where $\Sigma'(s) \equiv -1$ (depending on α).

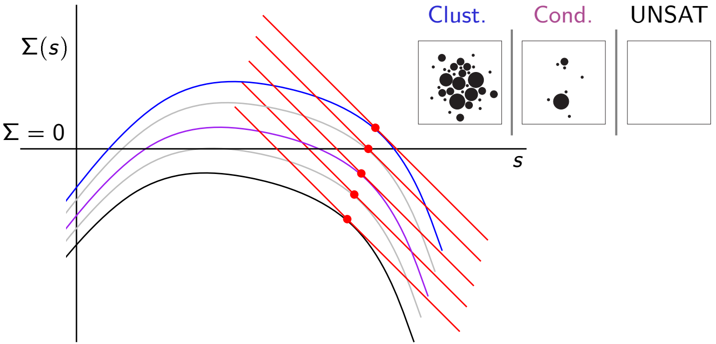


Condensation

Complexity function $\Sigma \equiv \Sigma_\alpha(s)$ such that:

$$\mathbb{E}Z = \sum \underbrace{(\text{cluster size})}_{\exp\{ns\}} \times \underbrace{\mathbb{E}[\text{number of clusters of that size}]}_{\exp\{n\Sigma(s)\}}$$

$\mathbb{E}Z$ is dominated by s where $\Sigma'(s) \equiv -1$ (depending on α).

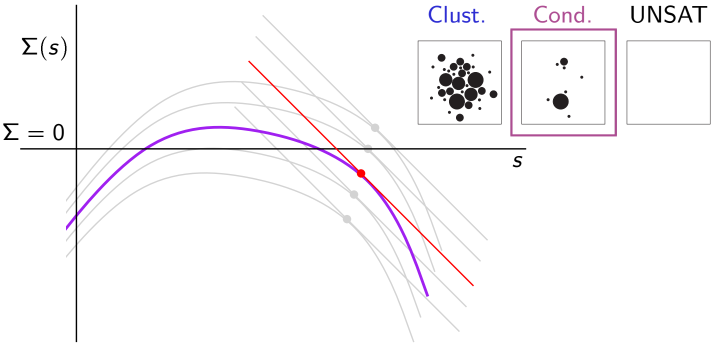


Condensation

Complexity function $\Sigma \equiv \Sigma_\alpha(s)$ such that:

$$\mathbb{E}Z = \sum \underbrace{(\text{cluster size})}_{\exp\{ns\}} \times \underbrace{\mathbb{E}[\text{number of clusters of that size}]}_{\exp\{n\Sigma(s)\}}$$

$\mathbb{E}Z$ is dominated by s where $\Sigma'(s) \equiv -1$ (depending on α).

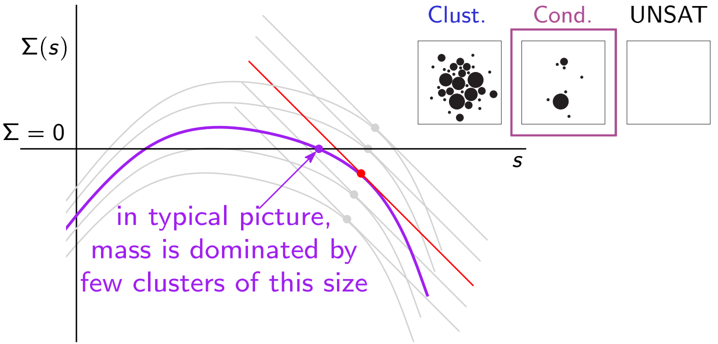


Condensation

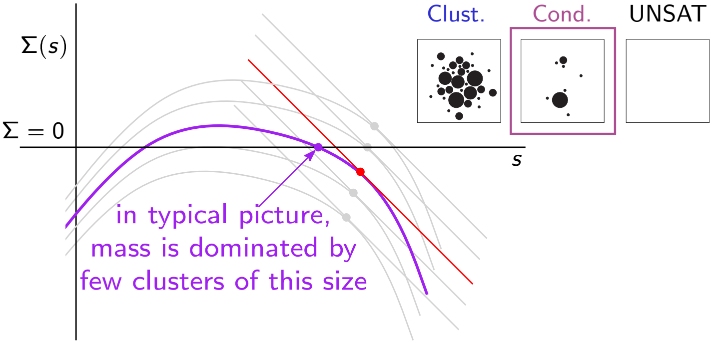
Complexity function $\Sigma \equiv \Sigma_\alpha(s)$ such that:

$$\mathbb{E}Z = \sum \underbrace{(\text{cluster size})}_{\exp\{ns\}} \times \underbrace{\mathbb{E}[\text{number of clusters of that size}]}_{\exp\{n\Sigma(s)\}}$$

$\mathbb{E}Z$ is dominated by s where $\Sigma'(s) \equiv -1$ (depending on α).



Condensation and non-concentration



The correct prediction:

$$\lim_{n\to\infty} \frac{1}{n} \log Z = \sup\{s + \Sigma(s) : \Sigma(s) > 0\} = \sup\{s : \Sigma(s) > 0\}$$

Physicist's Calculation: One-step Replica Symmetry Breaking

Counting clusters

Question: How do we find $\sup\{s : \Sigma(s) > 0\}$?

Counting clusters

Question: How do we find $\sup\{s : \Sigma(s) > 0\}$?

First step: Work with clusters of solutions.

$$\begin{aligned}\textbf{CLUSTERS} &\equiv \text{set of } k\text{-NAESAT solution clusters} \\ &= \text{set of connected components of } \textbf{SOL}\end{aligned}$$

1RSB suggests that there is no extra structure in **CLUSTERS**.

Counting clusters

Question: How do we find $\sup\{s : \Sigma(s) > 0\}$?

First step: Work with clusters of solutions.

$$\textbf{CLUSTERS} \equiv \text{set of } k\text{-NAESAT solution clusters}$$
$$= \text{set of connected components of } \textbf{SOL}$$

1RSB suggests that there is no extra structure in **CLUSTERS**.

Indeed, counting $\mathbb{E}|\textbf{CLUSTERS}|$ has lead to precise result of α_{sat}.

k-NAESAT: Coja-Oghlan–Panagiotou '12, Ding–Sly–Sun '13a

Independent set: Ding–Sly–Sun '13b

k-SAT: Coja-Oghlan–Panagiotou '13 '14, Ding–Sly–Sun '14

Counting clusters

Question: How do we find $\sup\{s : \Sigma(s) > 0\}$?

First step: Work with clusters of solutions.

$$\textbf{CLUSTERS} \equiv \text{set of } k\text{-NAESAT solution clusters}$$
$$= \text{set of connected components of } \textbf{SOL}$$

1RSB suggests that there is no extra structure in **CLUSTERS**.

Indeed, counting $\mathbb{E}|\textbf{CLUSTERS}|$ has lead to precise result of α_{sat}.

k-NAESAT: Coja-Oghlan–Panagiotou '12, Ding–Sly–Sun '13a

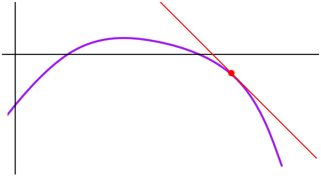
Independent set: Ding–Sly–Sun '13b

k-SAT: Coja-Oghlan–Panagiotou '13 '14, Ding–Sly–Sun '14

Not enough for our purpose...

Counting clusters weighted

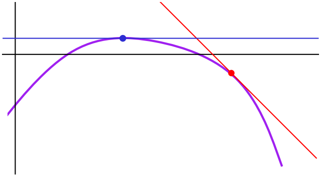
Second Step:



$$\mathbb{E}|\mathbf{SOL}| = \sum_s \exp\{n[1 \cdot s + \Sigma(s)]\}, \quad \text{maximized at } \Sigma'(s) = -1.$$

Counting clusters weighted

Second Step:

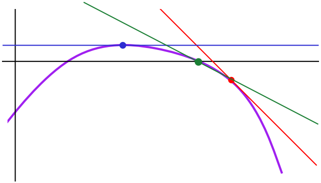


$$\mathbb{E}|\textbf{SOL}| = \sum_s \exp\{n[1 \cdot s + \Sigma(s)]\}, \quad \text{maximized at } \Sigma'(s) = -1.$$

$$\mathbb{E}|\textbf{CLUSTERS}| = \sum_s \exp\{n[0 \cdot s + \Sigma(s)]\}, \quad \text{maximized at } \Sigma'(s) = 0.$$

Counting clusters weighted

Second Step:

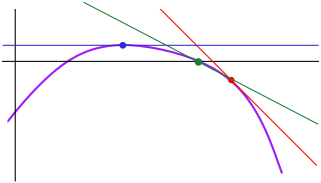


$$\mathbb{E}|\textbf{SOL}| = \sum_s \exp\{n[1 \cdot s + \Sigma(s)]\}, \quad \text{maximized at } \Sigma'(s) = -1.$$

$$\mathbb{E}|\textbf{CLUSTERS}| = \sum_s \exp\{n[0 \cdot s + \Sigma(s)]\}, \quad \text{maximized at } \Sigma'(s) = 0.$$

Counting clusters weighted

Second Step: Weight clusters by (their size)$^\lambda$



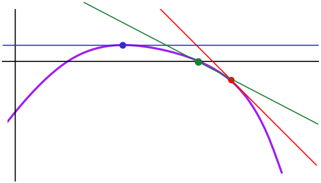
$$\mathbb{E}|\textbf{SOL}| = \sum_s \exp\{n[1 \cdot s + \Sigma(s)]\}, \quad \text{maximized at } \Sigma'(s) = -1.$$

$$\mathbb{E}|\textbf{CLUSTERS}| = \sum_s \exp\{n[0 \cdot s + \Sigma(s)]\}, \quad \text{maximized at } \Sigma'(s) = 0.$$

$$\mathbb{E}Z_\lambda \equiv \sum_s \exp\{n[\lambda \cdot s + \Sigma(s)]\}, \quad \text{maximized at } \Sigma'(s) = -\lambda$$

Counting clusters weighted

Second Step: Weight clusters by (their size)$^\lambda$



$$\mathbb{E}|\textbf{SOL}| = \sum\nolimits_s \exp\{n[1 \cdot s + \Sigma(s)]\}, \quad \text{maximized at } \Sigma'(s) = -1.$$

$$\mathbb{E}|\textbf{CLUSTERS}| = \sum\nolimits_s \exp\{n[0 \cdot s + \Sigma(s)]\}, \quad \text{maximized at } \Sigma'(s) = 0.$$

$$\mathbb{E}Z_\lambda \equiv \sum\nolimits_s \exp\{n[\lambda \cdot s + \Sigma(s)]\}, \quad \text{maximized at } \Sigma'(s) = -\lambda$$

In fact, $\frac{1}{n} \log \mathbb{E}Z_\lambda$ is the Legendre transformation of $\Sigma(s)$.

Explicit formula

For each $\lambda \in [0,1]$, there exist prob. measure $\mu_\lambda, \hat{\mu}_\lambda$ on $[0,1]$ such that

$$\mu_\lambda(B) = \mathscr{Z}_\lambda^{-1} \int \left(2 - \prod_{i=1}^{k-1} x_i - \prod_{i=1}^{k-1}(1-x_i)\right)^\lambda 1\left\{\frac{1 - \prod_{i=1}^{k-1} x_i}{2 - \prod_{i=1}^{k-1} x_i - \prod_{i=1}^{k-1}(1-x_i)} \in B\right\} \cdot \prod_{i=1}^{k-1} \hat{\mu}_\lambda(dx_i)$$

$$\hat{\mu}_\lambda(B) = \widehat{\mathscr{Z}}_\lambda^{-1} \int \left(\prod_{i=1}^{d-1} y_i + \prod_{i=1}^{d-1}(1-y_i)\right)^\lambda 1\left\{\frac{\prod_{i=1}^{d-1} y_i}{\prod_{i=1}^{d-1} y_i + \prod_{i=1}^{d-1}(1-y_i)} \in B\right\} \cdot \prod_{i=1}^{d-1} \mu_\lambda(dy_i)$$

Define $\Sigma(\lambda) \equiv \mathrm{Ent}(w_\lambda) + \alpha\,\mathrm{Ent}(\hat{w}_\lambda) - d\,\mathrm{Ent}(\bar{w}_\lambda)$, where

$$w_\lambda(B) = \mathcal{Z}_\lambda^{-1} \int \left(\prod_{i=1}^{d} y_i + \prod_{i=1}^{d}(1-y_i)\right)^\lambda 1\left\{\prod_{i=1}^{d} y_i + \prod_{i=1}^{d}(1-y_i) \in B\right\} \prod_{i=1}^{d} \hat{\mu}_\lambda(dy_i)$$

$$\hat{w}_\lambda(B) = \hat{\mathcal{Z}}_\lambda^{-1} \int \left(1 - \prod_{i=1}^{k} x_i - \prod_{i=1}^{k}(1-x_i)\right)^\lambda 1\left\{1 - \prod_{i=1}^{k} x_i - \prod_{i=1}^{k}(1-x_i) \in B\right\} \cdot \prod_{i=1}^{k} \mu_\lambda(dx_i)$$

$$\bar{w}_\lambda(B) = \bar{\mathcal{Z}}_\lambda^{-1} \iint \left(xy + (1-x)(1-y)\right)^\lambda 1\left\{xy + (1-x)(1-y) \in B\right\} \mu_\lambda(dx)\hat{\mu}_\lambda(dy).$$

Main Theorem.[S.-Sun-Zhang '16]

For $k \geqslant k_0$, $\alpha_{\text{cond}} \leqslant \alpha < \alpha_{\text{sat}}$. Let $\lambda_\star \equiv \sup\{\lambda : \Sigma(\lambda) > 0\}$.

$$\Phi(\alpha) = \lim_{n\to\infty} \frac{1}{n} \log Z = \frac{1}{\lambda_\star}\left[\log \mathcal{Z}_{\lambda_\star} + \alpha \log \hat{\mathcal{Z}}_{\lambda_\star} - d \log \bar{\mathcal{Z}}_{\lambda_\star}\right].$$

Explicit formula

For each $\lambda \in [0,1]$, there exist prob. measure $\mu_\lambda, \hat{\mu}_\lambda$ on $[0,1]$ such that

$$\mu_\lambda(B) = \mathscr{Z}_\lambda^{-1} \int \Big(2 - \prod^{k-1} x_i - \prod^{k-1} (1 - x_i)\Big)^\lambda 1\Big\{ \frac{1 - \prod_{i=1}^{k-1} x_i}{\cdots} \in B \Big\} \cdot \prod_{i=1}^{k-1} \hat{\mu}_\lambda(dx_i)$$

Some distributional recursion with fixed point

$$\hat{\mu}_\lambda(B) = \mathscr{Z}_\lambda \int \Big(\prod_{i=1} y_i + \prod_{i=1} (1 - y_i) \Big) 1\Big\{ \frac{\cdots}{\prod_{i=1}^{d-1} y_i + \prod_{i=1}^{d-1} (1 - y_i)} \in B \Big\} \cdot \prod_{i=1}^{d-1} \mu_\lambda(dy_i)$$

Define $\Sigma(\lambda) \equiv \mathrm{Ent}(w_\lambda) + \alpha\, \mathrm{Ent}(\hat{w}_\lambda) - d\, \mathrm{Ent}(\bar{w}_\lambda)$, where

w_λ $\Uparrow$ Complexity function depending on (μ_λ, λ) $\hat{\mu}_\lambda(dy_i)$

Some functional of (μ_λ, λ) $\equiv \frac{1}{\lambda}\big[\lambda s(\mu_\lambda, \lambda) + \Sigma(\mu_\lambda, \lambda)\big]$

$$\bar{w}_\lambda(B) = \bar{\mathcal{Z}}_\lambda^{-1} \iint \Big(xy + (1-x)(1-y)\Big)^\lambda 1\Big\{ xy + (1-x)(1-y) \in B \Big\} \mu_\lambda(dx) \hat{\mu}_\lambda(dy).$$

Main Theorem.[S.-Sun-Zhang '16]

For $k \geqslant k_0$, $\alpha_{\mathrm{cond}} \leqslant \alpha < \alpha_{\mathrm{sat}}$. Let $\lambda_\star \equiv \sup\{\lambda : \Sigma(\lambda) > 0\}$.

$$\Phi(\alpha) = \lim_{n\to\infty} \frac{1}{n} \log Z = \frac{1}{\lambda_\star}\Big[\log \mathcal{Z}_{\lambda_\star} + \alpha \log \hat{\mathcal{Z}}_{\lambda_\star} - d \log \bar{\mathcal{Z}}_{\lambda_\star}\Big].$$

Proof Overview

Upper bound

For upperbound, we prove a regular version of the interpolation bound of Franz–Leone '03, Panchenko–Talagrand '04. The proof resembles the proof of Bayati–Gamarnik–Tetali '13.

In particular, it implies that

$$\frac{1}{n}\log Z \leqslant s(\nu_\lambda^\star) + \lambda^{-1}\Sigma(\nu_\lambda^\star),$$

matching the lowerbound $s(\nu_\lambda^\star)$ as $\Sigma(\nu_\lambda^\star) \to 0$.

Encoding of local neighborhood

We represent elements of **CLUSTERS** as a spin system on $E(\mathcal{G})$.

- Start from $\underline{x} \in \{+, -\}^{V(\mathcal{G})}$ and explore the cluster $\mathcal{C}$.

Encoding of local neighborhood

We represent elements of **CLUSTERS** as a spin system on $E(\mathcal{G})$.

- Start from $\underline{x} \in \{+, -\}^{V(\mathcal{G})}$ and explore the cluster $\mathcal{C}$.
- Map each variable to a value from $\{+, -, \mathtt{f}\}$,
  s.t. a variable is marked $\mathtt{f}$ if it can take multiple values.

$$\Rightarrow \mathcal{C} \in \textbf{CLUSTERS} \leftrightarrow \underline{\tau} \in \{+, -, \mathtt{f}\}^{V(\mathcal{G})}$$

Encoding of local neighborhood

We represent elements of **CLUSTERS** as a spin system on $E(\mathcal{G})$.

- Start from $\underline{x} \in \{+, -\}^{V(\mathcal{G})}$ and explore the cluster $\mathcal{C}$.
- Map each variable to a value from $\{+, -, \mathtt{f}\}$,
  s.t. a variable is marked $\mathtt{f}$ if it can take multiple values.

$$\Rightarrow \mathcal{C} \in \textbf{CLUSTERS} \leftrightarrow \underline{\tau} \in \{+, -, \mathtt{f}\}^{V(\mathcal{G})}$$

This gives a new spin system where

- $\mathtt{f}$ are not forced by any clause.
- $+$ and $-$ variables must be forced by at least one clause.

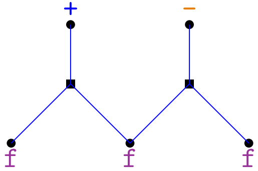
Encoding of local neighborhood

We represent elements of **CLUSTERS** as a spin system on $E(\mathcal{G})$.

- Start from $\underline{x} \in \{+,-\}^{V(\mathcal{G})}$ and explore the cluster $\mathcal{C}$.
- Map each variable to a value from $\{+,-,\mathtt{f}\}$, s.t. a variable is marked $\mathtt{f}$ if it can take multiple values.

$$\Rightarrow \mathcal{C} \in \textbf{CLUSTERS} \leftrightarrow \underline{\tau} \in \{+,-,\mathtt{f}\}^{V(\mathcal{G})}$$

  This gives a new spin system where
  - $\mathtt{f}$ are not forced by any clause.
  - $+$ and $-$ variables must be forced by at least one clause.
- Dependencies in free variable must be taken into account when counting solutions in clusters



Counting solutions within a cluster

We divide the subgraph of `f`'s into a forest of O(1)-size trees such that assigning values to one tree does not affect the others.

Counting solutions within a cluster

We divide the subgraph of f's into a forest of O(1)-size trees such that assigning values to one tree does not affect the others.
Every edge encodes the 'f-tree' it resides in.

$$\Rightarrow \underline{\tau} \in \{+, -, \mathtt{f}\}^{V(\mathcal{G})} \leftrightarrow \underline{\sigma} \in \{\mathtt{f}\text{-trees}\}^{E(\mathcal{G})}$$

Counting solutions within a cluster

We divide the subgraph of f's into a forest of O(1)-size trees such that assigning values to one tree does not affect the others.
Every edge encodes the 'f-tree' it resides in.

$$\Rightarrow \underline{\tau} \in \{+, -, \mathtt{f}\}^{V(\mathcal{G})} \leftrightarrow \underline{\sigma} \in \{\mathtt{f}\text{-trees}\}^{E(\mathcal{G})}$$

Recall that we can use BP algorithm to count solutions on trees.

Counting solutions within a cluster

We divide the subgraph of $\texttt{f}$'s into a forest of O(1)-size trees such that assigning values to one tree does not affect the others.
Every edge encodes the '$\texttt{f}$-tree' it resides in.

$$\Rightarrow \underline{\tau} \in \{+, -, \texttt{f}\}^{V(\mathcal{G})} \leftrightarrow \underline{\sigma} \in \{\texttt{f}\text{-trees}\}^{E(\mathcal{G})}$$

Recall that we can use BP algorithm to count solutions on trees.

Define weight functions Ψ_v, Ψ_a, Ψ_e accordingly s.t. for each $\underline{\sigma} \in \{\texttt{f}\text{-trees}\}^{E(\mathcal{G})}$

$$\begin{aligned} w(\underline{\sigma}) &\equiv \prod_v \Psi_v(\underline{\sigma}_{\delta v}) \prod_a \Psi_a(\underline{\sigma}_{\delta a}) \prod_{e=(av)} \Psi_e(\underline{\sigma}_{(av)}) \\ &= \prod_T (\#\text{ of ways of assigning } \texttt{f}\text{'s. in tree } T) \\ &= (\text{size of cluster}) \end{aligned}$$

Counting solutions within a cluster

We divide the subgraph of f's into a forest of O(1)-size trees such that assigning values to one tree does not affect the others.
Every edge encodes the 'f-tree' it resides in.

$$\Rightarrow \underline{\tau} \in \{+, -, \mathtt{f}\}^{V(\mathcal{G})} \leftrightarrow \underline{\sigma} \in \{\mathtt{f}\text{-trees}\}^{E(\mathcal{G})}$$

Recall that we can use BP algorithm to count solutions on trees.

Define weight functions Ψ_v, Ψ_a, Ψ_e accordingly s.t. for each $\underline{\sigma} \in \{\mathtt{f}\text{-trees}\}^{E(\mathcal{G})}$

$$\begin{aligned} w(\underline{\sigma}) &\equiv \prod\nolimits_v \Psi_v(\underline{\sigma}_{\delta v}) \prod\nolimits_a \Psi_a(\underline{\sigma}_{\delta a}) \prod\nolimits_{e=(av)} \Psi_e(\underline{\sigma}_{(av)}) \\ &= \prod\nolimits_T (\#\text{ of ways of assigning } \mathtt{f}\text{'s. in tree } T) \\ &= (\text{size of cluster}) \end{aligned}$$

Then we can define

$$Z_\lambda \equiv \sum\nolimits_{\underline{\sigma}} w^\lambda(\underline{\sigma}).$$

Optimization

We can write

$$\mathbb{E}Z_\lambda = \frac{\sum_{(\mathcal{G},\underline{\sigma})} w^\lambda(\underline{\sigma})}{\#\mathcal{G}}$$

Optimization

We can write

$$\mathbb{E}Z_\lambda = \frac{\sum_{(\mathcal{G},\underline{\sigma})} w^\lambda(\underline{\sigma})}{\#\mathcal{G}}$$

Then partitioning $\underline{\sigma}$ according to its empirical distribution ν,

$$\begin{aligned}\mathbb{E}Z_\lambda[\nu] &= \frac{\binom{n}{n\dot{\nu}}\binom{\alpha n}{\alpha n\hat{\nu}}}{\binom{dn}{dn\bar{\nu}}} \Psi_v{}^{\lambda n\dot{\nu}} \Psi_a{}^{\lambda\alpha n\hat{\nu}} \Psi_e{}^{\lambda dn\bar{\nu}} \\ &\equiv \exp\{n\,[\Sigma(\nu) + \lambda s(\nu)] + o(n)\} \\ &\equiv \exp\{n\,\Phi_\lambda(\nu) + o(n)\}\end{aligned}$$

Optimization

We can write

$$\mathbb{E}Z_\lambda = \frac{\sum_{(\mathcal{G},\underline{\sigma})} w^\lambda(\underline{\sigma})}{\#\mathcal{G}}$$

Then partitioning $\underline{\sigma}$ according to its empirical distribution ν,

$$\begin{aligned}\mathbb{E}Z_\lambda[\nu] &= \frac{\binom{n}{n\dot{\nu}}\binom{\alpha n}{\alpha n\hat{\nu}}}{\binom{dn}{dn\bar{\nu}}}{\Psi_v}^{\lambda n\dot{\nu}}{\Psi_a}^{\lambda\alpha n\hat{\nu}}{\Psi_e}^{\lambda dn\bar{\nu}} \\ &\doteq \exp\{n\,[\Sigma(\nu)+\lambda s(\nu)]+o(n)\} \\ &\doteq \exp\{n\,\Phi_\lambda(\nu)+o(n)\}\end{aligned}$$

Can find optimal ν by finding fixed points of the Belief Propagation equations Dembo–Montanari–Sun '13.

Optimization

We can write

$$\mathbb{E}Z_\lambda = \frac{\sum_{(\mathcal{G},\underline{\sigma})} w^\lambda(\underline{\sigma})}{\#\mathcal{G}}$$

Then partitioning $\underline{\sigma}$ according to its empirical distribution ν,

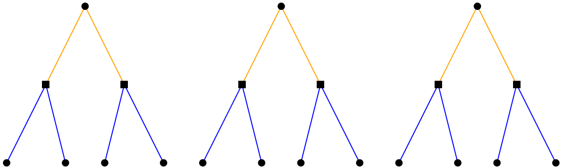
$$\begin{aligned}\mathbb{E}Z_\lambda[\nu] &= \frac{\binom{n}{n\dot{\nu}}\binom{\alpha n}{\alpha n\hat{\nu}}}{\binom{dn}{dn\bar{\nu}}} {\Psi_v}^{\lambda n\dot{\nu}} {\Psi_a}^{\lambda\alpha n\hat{\nu}} {\Psi_e}^{\lambda dn\bar{\nu}} \\ &\equiv \exp\{n\,[\Sigma(\nu) + \lambda s(\nu)] + o(n)\} \\ &\equiv \exp\{n\,\Phi_\lambda(\nu) + o(n)\}\end{aligned}$$

Can find optimal ν by finding fixed points of the Belief Propagation equations Dembo–Montanari–Sun '13.

Fixed points are distributions over bi-directional pairs of messages.

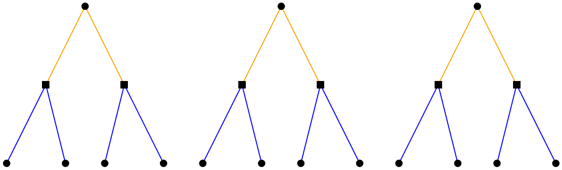
Optimization: from graph to trees

Choose $(\mathcal{G}, \underline{\sigma})$ weighted by $w^\lambda(\underline{\sigma})$ and sample ϵn vertices.



Optimization: from graph to trees

Choose $(\mathcal{G}, \underline{\sigma})$ weighted by $w^\lambda(\underline{\sigma})$ and sample ϵn vertices.

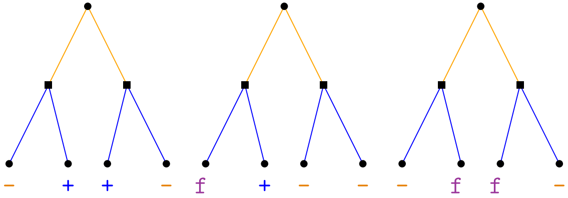


Sampling w.h.p. $\nu_1 = \nu_2 = \nu$

Optimization: from graph to trees

Choose $(\mathcal{G}, \underline{\sigma})$ weighted by $w^{\lambda}(\underline{\sigma})$ and sample ϵn vertices.

Record the boundary,

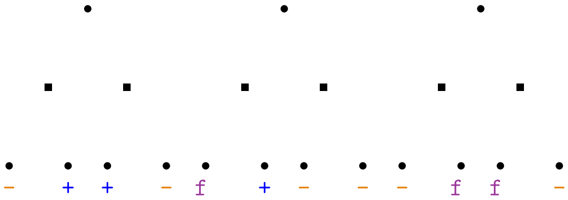


Sampling w.h.p. $\nu_1 = \nu_2 = \nu$

Optimization: from graph to trees

Choose $(\mathcal{G}, \underline{\sigma})$ weighted by $w^{\lambda}(\underline{\sigma})$ and sample ϵn vertices.

Record the boundary, remove the edges,

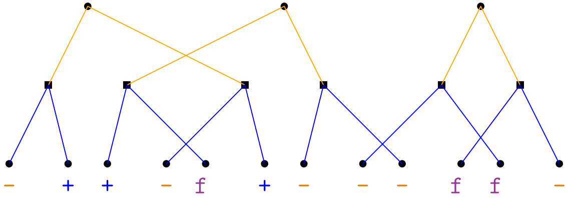


Sampling w.h.p. $\nu_1 = \nu_2 = \nu$

Optimization: from graph to trees

Choose $(\mathcal{G}, \underline{\sigma})$ weighted by $w^\lambda(\underline{\sigma})$ and sample ϵn vertices.

Record the boundary, remove the edges, rematches the graph.



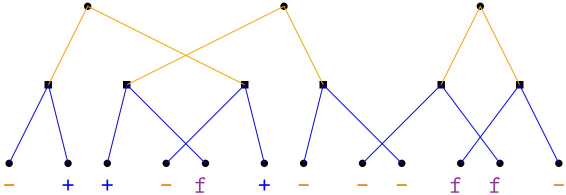
Sampling w.h.p. $\nu_1 = \nu_2 = \nu$

Optimization: from graph to trees

Choose $(\mathcal{G}, \underline{\sigma})$ weighted by $w^{\lambda}(\underline{\sigma})$ and sample ϵn vertices.

Record the boundary, remove the edges, rematches the graph.

Corresponds to sampling i.i.d. trees fixing marginals on leaves.



Sampling w.h.p. $\nu_1 = \nu_2 = \nu$

Resampling w.h.p. $\nu_1 = \mathrm{BP}(\nu_2)$

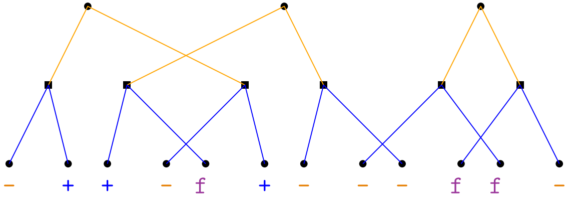
Optimization: from graph to trees

Choose $(\mathcal{G}, \underline{\sigma})$ weighted by $w^{\lambda}(\underline{\sigma})$ and sample ϵn vertices.

Record the boundary, remove the edges, rematches the graph.

Corresponds to sampling i.i.d. trees fixing marginals on leaves.

Preserves the distribution over $(\mathcal{G}, \underline{\sigma})$.



Sampling w.h.p. $\nu_1 = \nu_2 = \nu$

Resampling w.h.p. $\nu_1 = \mathrm{BP}(\nu_2)$

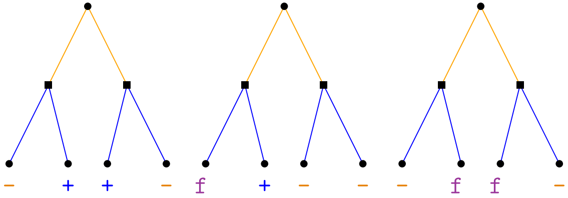
Optimization: from graph to trees

Choose $(\mathcal{G}, \underline{\sigma})$ weighted by $w^{\lambda}(\underline{\sigma})$ and sample ϵn vertices.

Record the boundary, remove the edges, rematches the graph.

Corresponds to sampling i.i.d. trees fixing marginals on leaves.

Preserves the distribution over $(\mathcal{G}, \underline{\sigma})$.



Sampling	w.h.p. $\nu_1 = \nu_2 = \nu$
Resampling	w.h.p. $\nu_1 = \mathrm{BP}(\nu_2)$

Hence $\nu^{\star}_{\lambda} = \nu_2 = \nu_1$

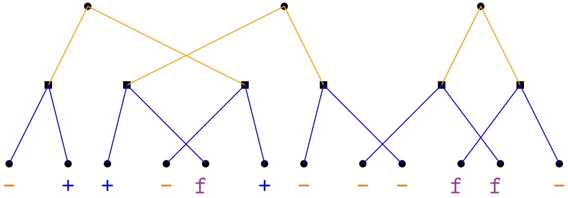
Optimization: from graph to trees

Choose $(\mathcal{G}, \underline{\sigma})$ weighted by $w^{\lambda}(\underline{\sigma})$ and sample ϵn vertices.

Record the boundary, remove the edges, rematches the graph.

Corresponds to sampling i.i.d. trees fixing marginals on leaves.

Preserves the distribution over $(\mathcal{G}, \underline{\sigma})$.



Sampling	w.h.p. $\nu_1 = \nu_2 = \nu$
Resampling	w.h.p. $\nu_1 = \mathrm{BP}(\nu_2)$

Hence $\nu^{\star}_{\lambda} = \nu_2 = \nu_1 = \mathrm{BP}(\nu_2) = \mathrm{BP}(\nu^{\star}_{\lambda})$.

Fixed point of a much simpler uni-directional BP equation.

Further directions

Extend to other models: Hardcore model, k-SAT, graph coloring...

Extend to other type of graphs: Erdos-Renyi graph.

Another source of non-concentration: atypical neighborhood.

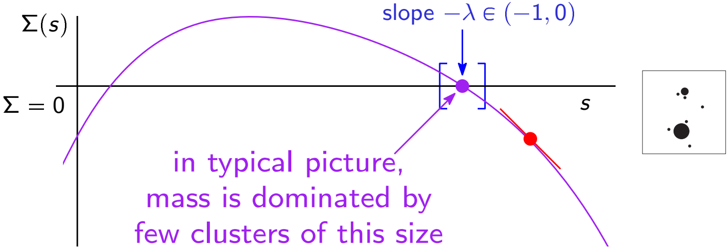
Show that the proportion of clusters are given by Poisson-Dirichlet process.

Applications to the stochastic block model.

Thank you.

Further directions: Poisson weighted clusters

Physics: $\exp\{n\Sigma(s)\}$ is the expected #clusters of size $\exp\{ns\}$



Expected #clusters of size $\exp\{ns_\star + x + dx\}$ is $\exp\{-\lambda x\}dx$;
so expected #clusters of size $\exp\{ns_\star\}(u + du)$ is $u^{-\lambda-1}du$

Therefore, cluster weights are given (up to normalization) by Poisson point process with intensity $u^{-\lambda-1}du$