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;
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.
Combinatorics and Theoretical Computer Science

Constraint satisfaction problem (CSP): is it possible to assign values to a set of \textit{variables} to satisfy a given set of \textit{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: $x_1, \ldots, x_n \in \{\text{TRUE, FALSE}\} \equiv \{+, -\}$

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:

\[
G(x) = (+x_1 \text{ OR } +x_2 \text{ OR } -x_3) \text{ AND } (+x_3 \text{ OR } +x_4 \text{ OR } -x_5) \\
\text{AND } (-x_1 \text{ OR } -x_4 \text{ OR } +x_5) \text{ AND } (+x_2 \text{ OR } -x_3 \text{ OR } +x_4)
\]
**K-SAT** The random K-SAT problem, a model of a random Boolean formula, is perhaps the canonical random CSP.

**Basic Definition:**

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

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:

\[
G(x) = ( +x_1 \text{ OR } +x_2 \text{ OR } -x_3 ) \text{ AND } ( +x_3 \text{ OR } +x_4 \text{ OR } -x_5 ) \text{ AND } ( -x_1 \text{ OR } -x_4 \text{ OR } +x_5 ) \text{ AND } ( +x_2 \text{ OR } -x_3 \text{ OR } +x_4 )
\]

**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: $G(x) =$

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

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: 

\[ G(\overline{x}) = ( +x_1 \text{ OR } +x_3 \text{ OR } -x_5 \text{ OR } -x_7 ) \text{ AND } (-x_1 \text{ OR } -x_2 \text{ OR } +x_5 \text{ OR } +x_6 ) \]

\[ \text{AND } (-x_3 \text{ OR } +x_4 \text{ OR } -x_6 \text{ OR } +x_7 ) \]

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

(4-SAT: each clause has degree 4)

\[ \text{variables } V \]

\[ \text{clauses } F \]

\[ \text{edges } E \]

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 \(a\)
**Graphical description**: We can encode a K-SAT formula as a bipartite graph:

Take a 4-SAT formula with 3 clauses: \( G(x) = \)

\[ (+x_1 \ OR \ +x_3 \ OR \ -x_5 \ OR \ -x_7) \ AND \ (\ -x_1 \ OR \ -x_2 \ OR \ +x_5 \ OR \ +x_6) \]

\[ \ AND \ (\ -x_3 \ OR \ +x_4 \ OR \ -x_6 \ OR \ +x_7) \]

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

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 $\alpha$ are there satisfying assignments?

Further Question:

Free Energy: How many solutions are there?

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 $\alpha$ are there satisfying assignments?
Initial Question:

- **Satisfiability Threshold**: For which $\alpha$ 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 $\alpha$ 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?
A $k$-NAESAT problem is a $k$-SAT where both $x$ and $\neg x$ are satisfying assignments. Each clause contains one $+$ and one $-$. 

clause of width $k = 4$

\[
\begin{align*}
\text{AND} & \left( +x_1 \text{ OR } +x_3 \text{ OR } -x_5 \text{ OR } -x_7 \right) \\
\text{AND} & \left( -x_1 \text{ OR } -x_2 \text{ OR } +x_5 \text{ OR } +x_6 \right) \\
\text{AND} & \left( -x_3 \text{ OR } +x_4 \text{ OR } -x_6 \text{ OR } +x_7 \right)
\end{align*}
\]
A $k$-NAESAT problem is a $k$-SAT where both $x$ and $\neg x$ are satisfying assignments. Each clause contains one $+$ and one $-$. 

\[
\begin{align*}
\text{clause of width } k = 4 \\
& ( +x_1 \text{ OR } +x_3 \text{ OR } -x_5 \text{ OR } -x_7 ) \\
& \text{AND} ( -x_1 \text{ OR } -x_2 \text{ OR } +x_5 \text{ OR } +x_6 ) \\
& \text{AND} ( -x_3 \text{ OR } +x_4 \text{ OR } -x_6 \text{ OR } +x_7 )
\end{align*}
\]

$d$-Regular NAESAT is an instance where every variable appears in exactly $d$ clauses.
A $k$-NAESAT problem is a $k$-SAT where both $x$ and $-x$ are satisfying assignments. Each clause contains one $+$ and one $-$. A clause of width $k = 4$ can be written as:

\[
\left( +x_1 \ OR \ +x_3 \ OR \ -x_5 \ OR \ -x_7 \right)
\]

AND

\[
\left( -x_1 \ OR \ -x_2 \ OR \ +x_5 \ OR \ +x_6 \right)
\]

AND

\[
\left( -x_3 \ OR \ +x_4 \ OR \ -x_6 \ OR \ +x_7 \right)
\]

$d$-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 -. A clause of width \( k = 4 \):

\[
\begin{align*}
\text{AND} & \quad (+x_1 \text{ OR } +x_3 \text{ OR } -x_5 \text{ OR } -x_7) \\
\text{AND} & \quad (-x_1 \text{ OR } -x_2 \text{ OR } +x_5 \text{ OR } +x_6) \\
\text{AND} & \quad (-x_3 \text{ OR } +x_4 \text{ OR } -x_6 \text{ OR } +x_7)
\end{align*}
\]

\textbf{d-Regular} NAESAT is an instance where every variable appears in exactly \( d \) clauses.

Why regular NAESAT? Same rich set of in phase transitions.
A $k$-NAESAT problem is a $k$-SAT where both $x$ and $\neg x$ are satisfying assignments. Each clause contains one $+$ and one $-$. 

clause of width $k = 4$

\[
\begin{align*}
&\left( +x_1 \text{ OR } +x_3 \text{ OR } -x_5 \text{ OR } -x_7 \right) \\
&\text{AND} \left( -x_1 \text{ OR } -x_2 \text{ OR } +x_5 \text{ OR } +x_6 \right) \\
&\text{AND} \left( -x_3 \text{ OR } +x_4 \text{ OR } -x_6 \text{ OR } +x_7 \right)
\end{align*}
\]

$d$-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 $\text{SOL} \equiv \{\text{solutions of } G\}$ and $Z \equiv |\text{SOL}|$. 
First moment and Second moment

We are interested in \( \text{SOL} \equiv \{\text{solutions of } \mathcal{G}\} \) and \( Z \equiv |\text{SOL}| \).
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 $\text{SOL} \equiv \{\text{solutions of } C\}$ and $Z \equiv |\text{SOL}|$. Keep $k$ fixed and let $\alpha \equiv d/k$.

**First moment method:**

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

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

**Second moment method:**

If $E r Z^2 p E Z q^2$, then $P r Z \approx \varepsilon E Z$ is bounded away from 0. Fails at $\alpha_2 \approx 2^{k-1} \ln 2 - \frac{1}{2} \ln 2$. 

Achlioptas–Moore '06

This $\alpha_2$ can be improved, but not all the way to $\alpha_1$. 

Coja-Oghlan–Zdeborová '12
We are interested in \( \text{SOL} \equiv \{ \text{solutions of } G \} \) and \( Z \equiv |\text{SOL}| \). Keep \( k \) fixed and let \( \alpha \equiv d/k \).

**First moment method:**

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

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

**Second moment method:**

If \( E[Z^2] \preceq (E[Z])^2 \), then \( P[Z > \varepsilon E[Z]] \) 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 \( \text{SOL} \equiv \{\text{solutions of } \mathcal{G}\} \) and \( Z \equiv |\text{SOL}| \).
Keep \( k \) fixed and let \( \alpha \equiv d/k \).

**First moment method:**

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

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

**Second moment method:**

If \( \mathbb{E}[Z^2] \preceq (\mathbb{E}Z)^2 \), then \( \mathbb{P}[Z > \varepsilon \mathbb{E}Z] \) bounded away from 0.
Fails at \( \alpha_2 \approx 2^{k-1} \ln 2 - \frac{1}{2}(\ln 2 + 1) \approx \alpha_1 - \frac{1}{2} \). \hspace{1cm} \text{Achlioptas–Moore '06}

This \( \alpha_2 \) can be improved, but not all the way to \( \alpha_1 \).

\hspace{1cm} \text{Coja-Oghlan–Zdeborová '12}
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 $\alpha \geq \alpha_{\text{cond}}$.  

Coja-Oghlan–Zdeborová '12, Ding–Sly–Sun '13a
Non-concentration of $Z$

First explanation:

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

$\mathbb{E}Z$ is dominated by unusual cluster of solutions of size $\geq 2^{n\epsilon}$.
Non-concentration of $Z$

First explanation:

Typically, any solution $x$ of $G$ has $\geq n\epsilon$ free variables, that can flip without violating any clause. 

— $\mathbb{E}Z$ is dominated by unusual cluster of solutions of size $\geq 2^{n\epsilon}$.
Non-concentration of $Z$

First explanation:

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

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: For \( k \geq k_0 \), 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.

Free energy: \( \Phi(\alpha) \equiv \lim_{n \to \infty} \frac{1}{n} \log Z. \)
Physicist’s Prediction:
Condensation and Replica Symmetry 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 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.

Phase diagram

Two solutions are connected if they differ by one bit.

KMRSZ ’07, MRS ’08
Phase diagram

The solution space \textbf{SOL} starts out as a well-connected cluster.

KMRSZ '07, MRS '08
The solution space **SOL** starts out as a well-connected cluster. After $\alpha_{\text{clust}}$, **SOL** decomposes into exponentially clusters.
The solution space **SOL** starts out as a well-connected cluster.
After $\alpha_{\text{clust}}$, **SOL** decomposes into exponentially clusters
After $\alpha_{\text{cond}}$, **SOL** is dominated by a few large clusters
The solution space SOL starts out as a well-connected cluster.
After $\alpha_{\text{clust}}$, SOL decomposes into exponentially clusters.
After $\alpha_{\text{cond}}$, SOL is dominated by a few large clusters.
After $\alpha_{\text{sat}}$, no solutions w.h.p.
Condensation

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

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

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

$$\mathbb{E}Z = \sum (\text{cluster size}) \times \mathbb{E}[\text{number of clusters of that size}]$$

$\exp\{ns\}$ $\exp\{n\Sigma(s)\}$
Condensation

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\mathbb{E} Z = \sum \frac{\text{(cluster size)}}{\exp\{ns\}} \times \mathbb{E}[\text{number of clusters of that size}] \frac{1}{\exp\{n\Sigma(s)\}}$$

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

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

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

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

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

$$\mathbb{E}Z = \sum \text{(cluster size)} \times \mathbb{E}[\text{number of clusters of that size}]$$

$$\exp\{n\Sigma(s)\}$$

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

In typical picture, mass is dominated by few clusters of this size.
Condensation and non-concentration

The correct prediction:

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

In typical picture, mass is dominated by few clusters of this size.
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.

$\text{CLUSTERS} \equiv$ set of $k$-NAESAT solution clusters

$= \text{set of connected components of } \text{SOL}$

1RSB suggests that there is no extra structure in $\text{CLUSTERS}$.
Counting clusters

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

**First step:** Work with clusters of solutions.

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

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

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

- 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.

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

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

Indeed, counting \( \mathbb{E} |\text{CLUSTERS}| \) has lead to precise result of \( \alpha_{\text{sat}} \).

- 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...
Second Step:

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

Second Step:

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

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

**Second Step:**

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

\[ E|\text{CLUSTERS}| = \sum_s \exp\{n[0 \cdot s + \Sigma(s)]\}, \quad \text{maximized at } \Sigma'(s) = 0. \]
**Second Step:** Weight clusters by \((\text{their size})^\lambda\)

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

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

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

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

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

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

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

In fact, \(\frac{1}{n} \log \mathbb{E}\mathcal{Z}_\lambda\) 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

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

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

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

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

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

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

---

**Main Theorem.** [S.-Sun-Zhang ’16]

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

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

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

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

Some distributional recursion with fixed point

$$
\hat{\mu}_\lambda(B) = 2^{\lambda} \left( \prod_{i=1}^{d-1} y_i + \prod_{i=1}^{d-1} (1 - y_i) \right)^\lambda \prod_{i=1}^{d-1} \left( \frac{1 + \prod_{j=i}^{d-1} y_j}{y_i} \right) \cdot \prod_{i=1}^{d-1} \mu_\lambda(dy_i)
$$

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

$$
w_\lambda \Uparrow \text{Complexity function depending on } (\mu_\lambda, \lambda) \hat{\mu}_\lambda(dy_i)
$$

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

$$
\bar{w}_\lambda(B) = \tilde{Z}_\lambda^{-1} \left( xy + (1 - x)(1 - y) \right)^\lambda \prod_{i=1}^{d-1} \left( \frac{1 + \prod_{j=i}^{d-1} y_j}{y_i} \right) \cdot \prod_{i=1}^{d-1} \mu_\lambda(dx_i) \hat{\mu}_\lambda(dy_i).
$$

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

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

$$
\Phi(\alpha) = \lim_{n \to \infty} \frac{1}{n} \log Z = \frac{1}{\lambda_*} \left[ \log \tilde{Z}_{\lambda_*} + \alpha \log \hat{Z}_{\lambda_*} - d \log \tilde{Z}_{\lambda_*} \right].
$$
Proof Overview
Proof overview: Upper bound (17/23)

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

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

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

Encoding of local neighborhood

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

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

$$\Rightarrow \mathcal{C} \in \text{CLUSTERS} \iff \tau \in \{+, -, \text{f}\}^{V(\mathcal{G})}$$
Encoding of local neighborhood

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

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

$$\Rightarrow \mathcal{C} \in \text{CLUSTERS} \iff \tau \in \{+,-,f\}^{V(G)}$$

This gives a new spin system where

- $f$ 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 $E(G)$.

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

$$\Rightarrow C \in \text{CLUSTERS} \iff \tau \in \{+,-,f\}^V(G)$$

This gives a new spin system where

- $f$ 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.

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

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

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

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

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

Define weight functions $\Psi_v, \Psi_a, \Psi_e$ accordingly s.t. for each $\sigma \in \{f\text{-trees}\}^{E(G)}$

$$w(\sigma) \equiv \prod_v \Psi_v(\sigma_{\delta v}) \prod_a \Psi_a(\sigma_{\delta a}) \prod_{e=(av)} \Psi_e(\sigma_{av})$$

$$= \prod_T (# \text{ of ways of assigning } f \text{'s. in tree } T)$$

$$= (\text{size of cluster})$$
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.

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

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

Define weight functions $\Psi_v, \Psi_a, \Psi_e$ accordingly s.t. for each $\sigma \in \{f\text{-trees}\}^{E(G)}$

$$w(\sigma) \equiv \prod_v \Psi_v(\sigma_{\delta v}) \prod_a \Psi_a(\sigma_{\delta a}) \prod_{e=(av)} \Psi_e(\sigma_{av})$$

$$= \prod_T \left( \# \text{ of ways of assigning } f \text{'s. in tree } T \right)$$

$$= \text{(size of cluster)}$$

Then we can define

$$Z_{\lambda} \equiv \sum_\sigma w^\lambda(\sigma).$$
Optimization

We can write

$$\mathbb{E} Z_\lambda = \frac{\sum_{(G,\sigma)} w^\lambda(\sigma)}{\#G}$$
We can write

$$\mathbb{E} Z_\lambda = \frac{\sum_{(G, \sigma)} \imath(G, \sigma)}{\#G}$$

Then partitioning $\sigma$ according to its empirical distribution $\nu$,

$$\mathbb{E} Z_\lambda[\nu] = \frac{n! (\alpha n) }{\alpha n!} \psi_v^{\lambda n \nu} \psi_a^{\lambda n \hat{\nu}} \psi_e^{\lambda d n \hat{\nu}}$$

$$\equiv \exp \{ n \left[ \sum (\nu) + \lambda s(\nu) \right] + o(n) \}$$

$$\equiv \exp \{ n \Phi_\lambda(\nu) + o(n) \}$$
We can write

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

Then partitioning $\sigma$ according to its empirical distribution $\nu$,

$$\mathbb{E} Z_\lambda[\nu] = \frac{n^{\nu}}{(\alpha n^{\nu})} \frac{\alpha n^{\nu}}{(\alpha n^{\nu})} \psi_v^{\lambda n^{\nu}} \psi_a^{\lambda n^{\nu}} \psi_e^{\lambda n^{\nu}}$$

$$\equiv \exp\{n [\Sigma(\nu) + \lambda s(\nu)] + o(n)\}$$

$$\equiv \exp\{n \Phi_\lambda(\nu) + o(n)\}$$

Can find optimal $\nu$ by finding fixed points of the Belief Propagation equations Dembo–Montanari–Sun ‘13.
Optimization

We can write

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

Then partitioning $\sigma$ according to its empirical distribution $\nu$,

$$\mathbb{E} Z_\lambda[\nu] = \binom{n}{\nu} \binom{\alpha n}{\alpha \nu} \psi_v^{\lambda \nu} \psi_a^{\lambda \alpha n \nu} \psi_e^{\lambda d \nu}$$

$$\equiv \exp\{n [\Sigma(\nu) + \lambda s(\nu)] + o(n)\}$$

$$\equiv \exp\{n \Phi_\lambda(\nu) + o(n)\}$$

Can find optimal $\nu$ 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 \((G, \sigma)\) weighted by \(w^\lambda(\sigma)\) and sample \(\epsilon n\) vertices.
Choose \((G, \sigma)\) weighted by \(w^\lambda(\sigma)\) and sample \(\epsilon n\) vertices.

Sampling

w.h.p. \(\nu_1 = \nu_2 = \nu\)
Choose \((\mathcal{G}, \sigma)\) weighted by \(w^\lambda(\sigma)\) and sample \(\epsilon n\) vertices. Record the boundary,

\[
\begin{align*}
\text{Sampling} & \quad \text{w.h.p.} \quad \nu_1 = \nu_2 = \nu
\end{align*}
\]
Optimization: from graph to trees

Choose \((\mathcal{G}, \sigma)\) weighted by \(w^\lambda(\sigma)\) and sample \(\epsilon n\) vertices. Record the boundary, remove the edges,

\[
\begin{array}{ccccccccccc}
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\end{array}
\]

Sampling \(\text{w.h.p.}\) \(\nu_1 = \nu_2 = \nu\)
Optimization: from graph to trees

Choose \((\mathcal{G}, \sigma)\) weighted by \(w^\lambda(\sigma)\) and sample \(\epsilon n\) vertices. Record the boundary, remove the edges, rematches the graph.

\[
\begin{align*}
- & + + - f + - - - f f - \\
\end{align*}
\]

Sampling w.h.p. \(\nu_1 = \nu_2 = \nu\)
Optimization: from graph to trees

Choose \((\mathcal{G}, \sigma)\) weighted by \(w^\lambda(\sigma)\) and sample \(\epsilon n\) vertices.
Record the boundary, remove the edges, remixes the graph.
Corresponds to sampling i.i.d. trees fixing marginals on leaves.

\[
\begin{align*}
\text{Sampling} & \quad \text{w.h.p.} \quad \nu_1 = \nu_2 = \nu \\
\text{Resampling} & \quad \text{w.h.p.} \quad \nu_1 = \text{BP}(\nu_2)
\end{align*}
\]
Optimization: from graph to trees

Choose \((\mathcal{G}, \sigma)\) weighted by \(w^\lambda(\sigma)\) and sample \(\epsilon n\) 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 \((\mathcal{G}, \sigma)\).

\[
\begin{align*}
\text{Sampling} & \quad \text{w.h.p.} \quad \nu_1 = \nu_2 = \nu \\
\text{Resampling} & \quad \text{w.h.p.} \quad \nu_1 = \text{BP}(\nu_2)
\end{align*}
\]
Optimization: from graph to trees

Choose \((\mathcal{G}, \sigma)\) weighted by \(w^\lambda(\sigma)\) and sample \(\epsilon n\) 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 \((\mathcal{G}, \sigma)\).

\[\text{Sampling} \quad \text{w.h.p.} \quad \nu_1 = \nu_2 = \nu\]
\[\text{Resampling} \quad \text{w.h.p.} \quad \nu_1 = \text{BP}(\nu_2)\]

Hence \(\nu^*_\lambda = \nu_2 = \nu_1\)
Optimization: from graph to trees

Choose \((G, \sigma)\) weighted by \(w^\lambda(\sigma)\) and sample \(\epsilon n\) 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 \((G, \sigma)\).

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

Hence \(\nu^* = \nu_2 = \nu_1 = \text{BP}(\nu_2) = \text{BP}(\nu^*)\).
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 \( \exp\{ns\} \)

\[
\Sigma(s) \quad \text{slope } -\lambda \in (-1, 0)
\]

\[
\Sigma = 0 \quad \text{in typical picture, mass is dominated by few clusters of this size}
\]

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

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