The Complexity of Approximate Counting

Leslie Ann Goldberg, University of Oxford

8th International Conference on Language and Automata Theory and Applications (LATA 2014)
Madrid
10–14 March 2014

Computational Counting

Computational Problems

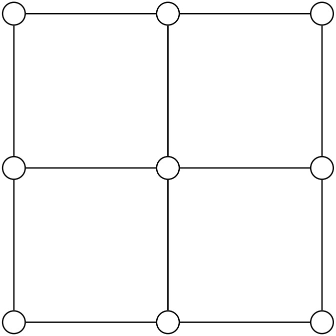
Decision: Is this Boolean formula satisfiable? Does this graph have a Hamiltonian cycle?

Optimisation: What is the maximum flow in this graph? What is the minimum length of a tour of this graph?

Counting: What is the value of this integral? What is the expectation of this random variable? Computing a weighted sum.

Partition Functions

Computational counting is concerned with the evaluation and approximate evaluation of partition functions. A partition function is a sum of products. Example: The Ising model.

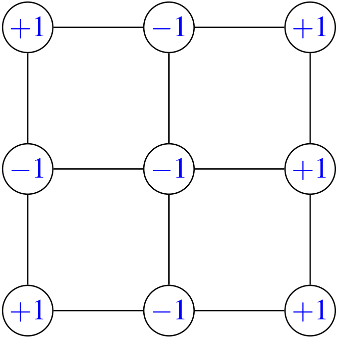


Graph $G = (V, E)$

Edge (i, j): interaction energy $J_{i,j}$

Vertex k:
local external magnetic field μ_k

inverse temperature β



Graph $G = (V, E)$

Edge (i, j): interaction energy $J_{i,j}$

Vertex k:
local external magnetic field μ_k

inverse temperature β

Energy of configuration $x \colon V \to \{-1, +1\}$:

$$H(x) = -\sum_{(i,j)\in E} J_{i,j}\, x_i\, x_j - \sum_{k\in V} \mu_k\, x_k$$

Probability of x in the Boltzmann distribution: $P(x) = e^{-\beta H(x)}/Z$.

The partition function: $Z = \sum_x e^{-\beta H(x)}$.

Monochromatic edge (i,j) contributes a factor of $\exp(\beta J_{i,j})$ to the partition function.

Bichromatic edge (i,j) contributes a factor of $\exp(-\beta J_{i,j})$.

Ferromagnetic Case: $\forall_{i,j}\, J_{i,j} > 0$
(weight of monochromatic edge is > 1)

$+1$ spin at vertex k contributes a factor of $\exp(\beta \mu_k)$

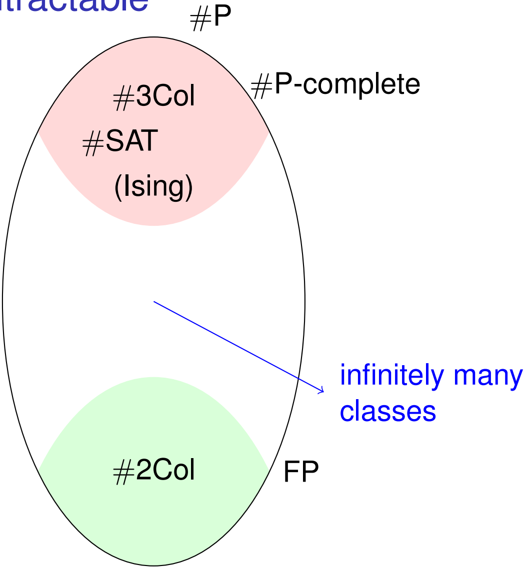
-1 spin at vertex k contributes a factor of $\exp(-\beta \mu_k)$

No fields: $\forall_k \mu_k = 0$. Mixed fields: μ_k values with both signs.

Example: If $V = \{1,2\}$ and $E = \{(1,2)\}$ and $\beta J_{1,2} = \ln 2$ and $\mu_1 = \mu_2 = 0$ then $Z(G) = 2 + 1/2 + 1/2 + 2 = 5$.

The expectation of $f(x)$: $\sum_x f(x) P(x)$.

Early work on counting complexity: Mapping the boundary between tractable and intractable



Valiant 1979

Ladner 1975

A smaller problem domain: CSPs

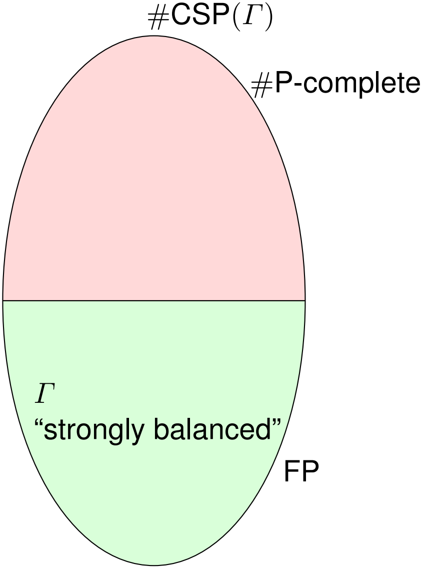
- A finite domain D. Example: $D = \{\text{red}, \text{blue}, \text{green}\}$
- A finite constraint language Γ (a set of relations on D)
  Example: Γ is the set containing the single relation
  $\{(\text{red}, \text{blue}), (\text{red}, \text{green}), (\text{blue}, \text{red}), (\text{blue}, \text{green}), (\text{green}, \text{red}), (\text{green}, \text{blue})\}$

An instance :

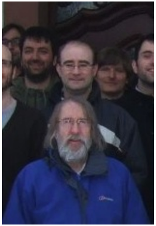
- A set of n variables, taking values in D
  Example: The vertices of a graph
- A set of constraints on the variables. Each constraint is a relation from Γ applied to the scope of the constraint, which is a tuple of variables.
  Example: One constraint per edge

The goal: (for CSP(Γ)) decide whether there is a satisfying assignment, or (for $\#$CSP(Γ)) count the satisfying assignments.

The complexity depends on Γ

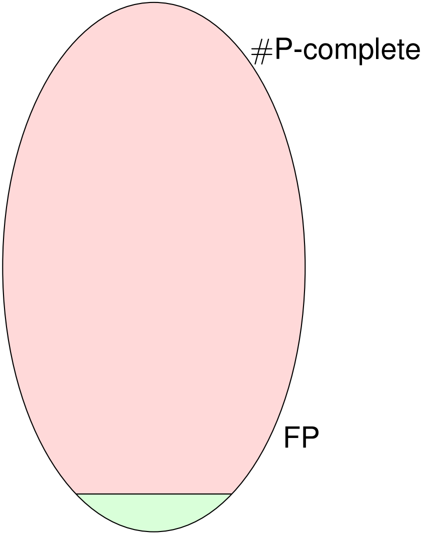


Bulatov 2008

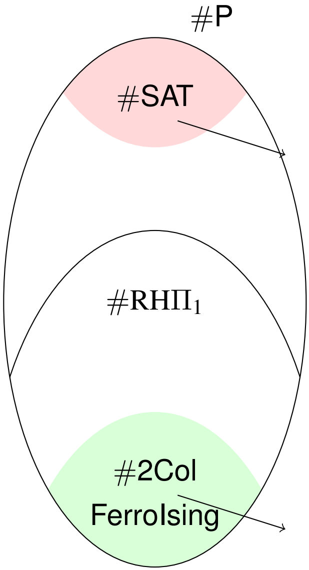


Dyer and Richerby 2010

Many important extensions described by Jin-Yi Cai in LATA 2013


#P-complete
FP

Three approximation complexity classes within $\#$P



Complete for $\#$P wrt AP-Reductions

Counting versions of NP-hard problems.

No FPRAS unless NP $=$ RP.

(Dyer, Goldberg, Greenhill, Jerrum 2003)

More liberal than parsimonious reductions

polynomial interpolation is not preserved by approximation.

FPRAS: Input instance I and ε

get within $1 \pm \varepsilon$ in time $\mathrm{poly}(|I|, \varepsilon^{-1})$

Robust notion: Powerable failure prob.

Typically for partition functions “No FPRAS” means “can’t get within a poly factor”

(FerroIsing: Jerrum Sinclair 1992)

$\#\mathrm{RH}\Pi_1$: Restricted Horn Π_1

logical description of #P (Saluja, Subrahmanyam and Thakur 1995)

Vocabulary: $\{\tilde{R}_0, \ldots, \tilde{R}_{k-1}\}$ relation symbols (specified arities)

Problem in #P: first order sentence φ using $\{\tilde{R}_0, \ldots, \tilde{R}_{k-1}\}$ and also new relation symbols $\tilde{T}_i$ and variables $\tilde{z}_i$.

Input: Structure $\mathbf{A} = (A, R_0, \ldots, R_{k-1})$ where A is finite universe and R_i is a relation with correct arity

Output: # of $\mathbf{T} = (T_1, \ldots, T_r)$ relations and $\mathbf{z} = (z_1, \ldots, z_m)$ (assignments of values in A to the vars) such that $\mathbf{A} \models \varphi(\mathbf{z}, \mathbf{T})$.

Example: #IS: The vocabulary is $\{\sim\}$.

$$\varphi = \forall x, y \quad (x \sim y \implies -I(x) \vee -I(y)) \wedge$$
$$(x \sim y \implies y \sim x) \wedge (x \sim y \implies x \neq y)$$

$\mathbf{T} = (I)$

$\#\mathrm{RH}\Pi_1$: Restricted Horn Π_1

φ is of the form $\forall \mathbf{y}\, \psi(\mathbf{y}, \mathbf{z}, \mathbf{T})$

ψ: unquantified CNF formula. Each clause has at most one unnegated relation symbol from $\mathbf{T}$ and at most one negated relation symbol from $\mathbf{T}$.

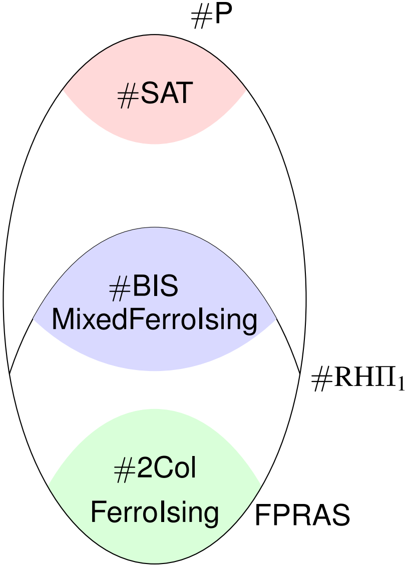
Example: #BIS. The vocabulary is $\{\sim, L\}$.

$$\begin{aligned}\varphi = \forall x, y \quad & (L(x) \wedge x \sim y \wedge X(x) \implies X(y)) \wedge \\ & (x \sim y \implies y \sim x) \wedge (L(x) \wedge x \sim y \implies \neg L(y)) \,.\end{aligned}$$

$X(x)$ is true for left vertices x in the IS and for right vertices which are not in the IS.

Π_1 means only universal quantification. Horn clauses have at most one positive literal. (this is also called restricted Krom SNP — it is related to what you can express in linear Datalog)

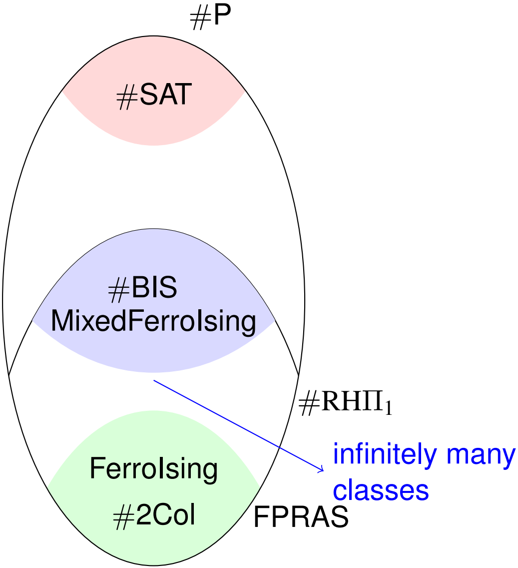
Complete problems for $\#\mathrm{RH}\Pi_1$



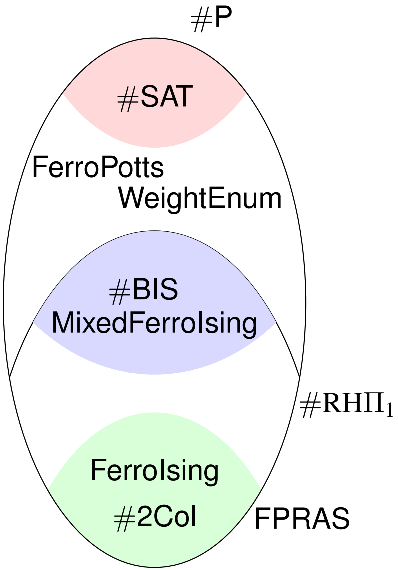
Approximate counting problems which are $=_{\mathrm{AP}}$ #BIS

- Counting downsets in a partial order
- FerroIsing with mixed fields
- FerroIsing in a hypergraph
- Counting stable matchings in some models
- Counting stable roommate assignments in some models
- H-colouring problems
- #CSP problems



#P
#SAT
#BIS
MixedFerroIsing
#RHΠ1
infinitely many classes
FerroIsing
#2Col
FPRAS



- Weights #CSPs
- H-colouring counting problems

Adding weights to $\#$CSPs

- A finite domain D (For the Ising model, $D = \{-1, +1\}$)
- A finite weighted constraint language $\mathcal{F}$: a set of functions which map tuples from D to a codomain R. (In the Ising model, the constraint functions map pairs of spins to interaction energies)

An instance of $\#\text{CSP}(\mathcal{F})$:

- A set of n variables, taking values in D
- A set of constraints on the variables. Each constraint is a function from $\mathcal{F}$ applied to the scope of the constraint, which is a tuple of variables.

Partition Function:

Sum: over assignments of domain elements to variables.

Product: of values of the constraint functions.

Example: ferromagnetic Ising

Assume $J_{i,j} = J > 0$ and $\mu_k = 0$

$D = \{-1, +1\}$

$\mathcal{F} = \{f\}$, where f is the binary function

$$f(x,y) = \begin{cases} \exp(\beta J), & \text{if } x = y; \\ \exp(-\beta J), & \text{otherwise.} \end{cases}$$

An instance encodes a graph $G = (V, E)$ with $V = \{v_1, \ldots, v_n\}$.

- The variables are the vertices in V.
- One f constraint for each edge in E.

$$Z(G) = \sum_{x\colon V \to D} \prod_{(v_i, v_j) \in E} f(x_i, x_j).$$

Example: Counting 3-Colourings

$D = \{\text{red}, \text{blue}, \text{green}\}$. $R = \{0, 1\}$. $\mathcal{F} = \{\text{NEQ}\}$.

$$\text{NEQ}(x, y) = \begin{cases} 1, & \text{if } x \neq y; \\ 0, & \text{otherwise.} \end{cases}$$

An instance encodes a graph $G = (V, E)$ with $V = \{v_1, \ldots, v_n\}$.

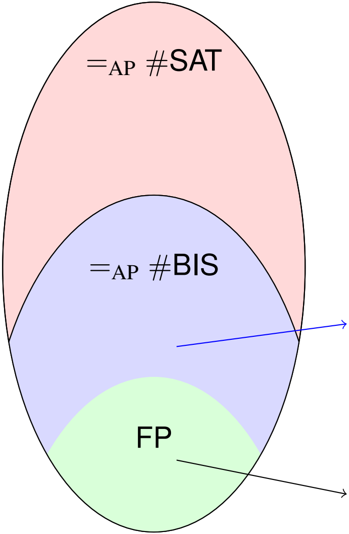
- The variables are the vertices in V.
- One NEQ constraint for each edge in E.

$$Z(G) = \sum_{x\colon V \to D} \prod_{(v_i, v_j) \in E} \text{NEQ}(x_i, x_j).$$

There is an FPRAS for $\#\text{CSP}(\{f\})$ but not for $\#\text{CSP}(\{\text{NEQ}\})$ unless $\text{NP} = \text{RP}$.

A trichotomy for $\#\text{CSP}(\mathcal{F})$ when $D = \{0, 1\}$ and $R = \{0, 1\}$.
(Boolean domain. Functions are relations.)

Dyer, Goldberg, Jerrum, 2010



express each relation in $\mathcal{F}$
as conjunction of
implications and pinnings

express each relation in $\mathcal{F}$
as set of solutions to
system of linear equations over GF(2)

More general weighted constraint languages

- $D = \{0, 1\}$
- $R = \mathbb{R}^p$

  non-negative efficiently-computable real numbers

  (n most significant bits can be computed in poly(n) time)
- $\mathcal{B}^p$: Set of all functions from tuples of Boolean values to $\mathbb{R}^p$.

Given a finite $\mathcal{F} \subset \mathcal{B}^p$:

What is the complexity of approximately solving $\#$CSP$(\mathcal{F})$?

(recent joint work with Bulatov, Chen, Dyer, Jerrum, Lu, McQuillan, Richerby)

A partial classification

Conservative case (all unary functions in $\mathcal{B}^p$ are contained in $\mathcal{F}$)

Theorem.

- *If you can "build" every function in $\mathcal{F}$ using NEQ and unary functions then, for any finite $\mathcal{G} \subset \mathcal{F}$, there is an FPRAS for* $\#\mathrm{CSP}(\mathcal{G})$.
- *Otherwise,*
  - *there is a finite $\mathcal{G} \subset \mathcal{F}$ such that $\#\mathrm{CSP}(\mathcal{G})$ is at least as hard to approximate as* #BIS.
  - *Furthermore, if there is a function $F \in \mathcal{F}$ that is not log-supermodular then there is a finite $\mathcal{G} \subset \mathcal{F}$ such that $\#\mathrm{CSP}(\mathcal{G})$ is* #SAT*-hard to approximate.*

Definition. An n-ary function $F \in \mathcal{B}^p$ is log-supermodular if $F(\boldsymbol{x} \vee \boldsymbol{y})F(\boldsymbol{x} \wedge \boldsymbol{y}) \geq F(\boldsymbol{x})F(\boldsymbol{y})$ for all $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^n$.

Example: The function IMP. For $\boldsymbol{x} = (0,1)$, $\boldsymbol{y} = (1,0)$, $\mathrm{IMP}(1,1)\mathrm{IMP}(0,0) \geq \mathrm{IMP}(0,1)\mathrm{IMP}(1,0)$.

What about larger domains?

Any finite domain D. Codomain $R = \mathbb{Q}_{\geq 0}$.

Conservative case (all unary functions from D to $\mathbb{Q}_{\geq 0}$ are contained in $\mathcal{F}$)

- If $\mathcal{F}$ is weakly log-modular then, for any finite $\mathcal{G} \subset \mathcal{F}$, $\#\text{CSP}(\mathcal{G})$ is exactly solvable in polynomial time.
- Otherwise, there is a finite $\mathcal{G} \subset \mathcal{F}$ such that $\#\text{CSP}(\mathcal{G})$ is at least as hard to approximate as $\#\text{BIS}$. Furthermore,
    - if $\mathcal{F}$ is weakly log-supermodular then, for any finite $\mathcal{G} \subset \mathcal{F}$, there is a finite set $\mathcal{G}'$ of log-supermodular functions on the Boolean domain such that $\#\text{CSP}(\mathcal{G})$ is as easy to approximate as $\#\text{CSP}(\mathcal{G}')$;
    - otherwise, there is a finite $\mathcal{G} \subset \mathcal{F}$ such that $\#\text{CSP}(\mathcal{G})$ is $\#\text{SAT}$-hard to approximate.

Definitions

Definition. A weighted constraint language $\mathcal{F}$ is weakly log-modular if, for all binary functions F that can be built using functions in $\mathcal{F}$ and for all elements $a, b \in D$,

$$\begin{aligned} &F(a,a)F(b,b) = F(a,b)F(b,a), \text{ or} \\ &F(a,a) = F(b,b) = 0, \text{ or} \\ &F(a,b) = F(b,a) = 0. \end{aligned} \tag{1}$$

Definition. $\mathcal{F}$ is weakly log-supermodular if, for all binary functions F that can be built using functions in $\mathcal{F}$ and for all elements $a, b \in D$,

$$F(a,a)F(b,b) \geq F(a,b)F(b,a) \quad \text{or} \quad F(a,a) = F(b,b) = 0. \tag{2}$$

Credits

- The polynomial-time solvability builds on the exact classification of $\#\mathrm{CSP}(\mathcal{F})$ by Cai, Chen, and Lu (2011), and in particular on the key role played by "balance" (introduced by Dyer and Richerby (2010)).
- The LSM-easiness builds on three key studies of the complexity of optimisation CSPs by Takhanov (2010); Cohen, Cooper and Jeavons (2008); and Komogorov and Živný(2012).

A trichotomy for the binary case

$\mathcal{F}$ is conservative, but also all functions in $\mathcal{F}$ have arity 1 or 2

- If $\mathcal{F}$ is weakly log-modular then, for any finite $\mathcal{G} \subset \mathcal{F}$, $\#$CSP$(\mathcal{G})$ is exactly solvable in polyomial time.
- Otherwise, if $\mathcal{F}$ is weakly log-supermodular, then
  - for every finite $\mathcal{G} \subset \mathcal{F}$, $\#$CSP$(\mathcal{G})$ is as easy to approximate as $\#$BIS and
  - there is a finite $\mathcal{G} \subset \mathcal{F}$ such that $\#$CSP$(\mathcal{G})$ is as hard to approximate as $\#$BIS.
- Otherwise, there is a finite $\mathcal{G} \subset \mathcal{F}$ such that $\#$CSP$(\mathcal{G})$ is $\#$SAT-hard to approximate.

(Relies additionally on work of Rudolf and Woeginger (1995) on decomposing Monge matrices.)