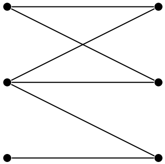
Theorem (Hell & Nešetřil 1990)

Let H be a fixed graph. The H-Coloring *problem in in P, if H either has a loop (self-loop) or is bipartite.*
Otherwise, H-Coloring *is NP-complete.*

Easy cases:



The problem of counting graph homomorphisms

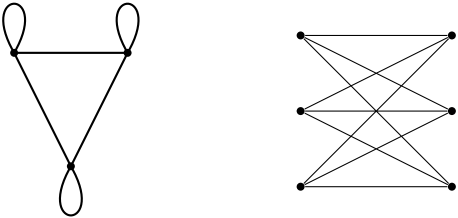
Theorem (Dyer & Greenhill 2000)

Let H be a fixed graph. The #H-Colorings *problem is in FP, if every connected component of H is*

1. *either a complete graph with all loops present*
2. *or a complete bipartite graph with no loops present.*

Otherwise, #H-Colorings *is #P-complete.*

Easy cases:



Suppose that H is a complete graph with all loops present, or a complete bipartite graph with no loops. Then, #H-COLORINGS can be solved in polynomial time.

Proof.

1. If H is an isolated vertex without a loop, then $Z_H(G) = 0$, unless G is a collection of isolated vertices, in which case $Z_H(G) = 1$.

Suppose that H is a complete graph with all loops present, or a complete bipartite graph with no loops. Then, $\#\text{H-Colorings}$ can be solved in polynomial time.

Proof.

1. If H is an isolated vertex without a loop, then $Z_H(G) = 0$, unless G is a collection of isolated vertices, in which case $Z_H(G) = 1$.
2. If H is the complete graph on k vertices with all loops present, then if G has n vertices, it holds that $Z_H(G) = k^n$.

Suppose that H is a complete graph with all loops present, or a complete bipartite graph with no loops. Then, #H-COLORINGS can be solved in polynomial time.

Proof.

1. If H is an isolated vertex without a loop, then $Z_H(G) = 0$, unless G is a collection of isolated vertices, in which case $Z_H(G) = 1$.
2. If H is the complete graph on k vertices with all loops present, then if G has n vertices, it holds that $Z_H(G) = k^n$.
3. Suppose H is the complete bipartite graph with vertex bipartition $C_1 \cup C_2$, $|C_i| = k_i$, $i = 1, 2$, and with no loops.
   - If G is not bipartite, then $Z_H(G) = 0$.
   - If G is bipartite with vertex bipartition $V_1 \cup V_2$, $|V_i| = n_i$, $i = 1, 2$, then

$$Z_H(G) = k_1^{n_1} \cdot k_2^{n_2} + k_1^{n_2} \cdot k_2^{n_1}.$$

□

Generalization to non-negative weights

Theorem (Bulatov & Grohe 2005)

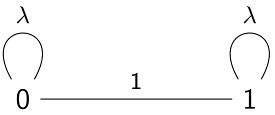
Let H be a fixed weighted graph and A the corresponding adjacency matrix with non-negative real entries. On input G, computing $Z_A(G)$ is in FP if every connected component of H is

1. *either not bipartite and the rank of A is at most 1*
2. *or bipartite and A is in block form $\begin{bmatrix} 0 & B \\ B^\top & 0 \end{bmatrix}$, where B is of rank at most 1.*

Otherwise, computing $Z_A(G)$ is #P-hard.

Examples of #P-hard counting problems

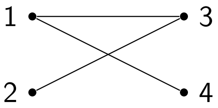
$\mathbf{H_{Ising}}$



Adjacency matrix

$$A = \begin{bmatrix} \lambda & 1 \\ 1 & \lambda \end{bmatrix}$$

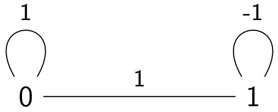
$\mathbf{H_{BIS}}$



Adjacency matrix

$$A = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

What about real weights?

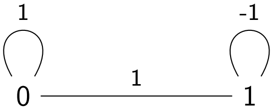


Adjacency matrix

$$A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- Every assignment σ from $V(G)$ to $V(H)$ corresponds to the subgraph induced by the nodes of G assigned to node 1 of H.
- Let an assignment σ. Then σ contributes either a $+1$ to the sum $Z_A(G)$ if an even number of edges have both their endpoints assigned to node 1 of H, or a -1, otherwise.
- $X = \frac{2^n + Z_A(G)}{2}$ is the number of induced subgraphs of G with even number of edges.

What about real weights?



Adjacency matrix

$$A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- A has rank 2, but the problem is in FP.
- Consider the quadratic form over $\mathbb{F}_2$ defined by

$$Q(X) = \sum_{\{u,v\} \in E(G)} x_u x_v.$$

- Then $Q(X) = 0$ if an even number of edges have both their endpoints assigned to 1, and $Q(X) = 1$, otherwise.
- Computing the number of solutions to $Q(X) = 0$ can be done in polynomial time.

Dichotomy theorems for #CSP($\mathcal{F}$): the Boolean case

- Creignou and Hermann (1996) proved a dichotomy theorem for the case of $\mathcal{F}$ being a finite set of $\{0, 1\}$-valued functions.

Dichotomy theorems for $\#\mathrm{CSP}(\mathcal{F})$: the Boolean case

- Creignou and Hermann (1996) proved a dichotomy theorem for the case of $\mathcal{F}$ being a finite set of $\{0, 1\}$-valued functions.
- Dyer, Goldberg and Jerrum (2009) a similar dichotomy when $\mathcal{F}$ is a finite set of non-negative real-valued functions.

Dichotomy theorems for $\#\mathrm{CSP}(\mathcal{F})$: the Boolean case

- Creignou and Hermann (1996) proved a dichotomy theorem for the case of $\mathcal{F}$ being a finite set of $\{0,1\}$-valued functions.
- Dyer, Goldberg and Jerrum (2009) a similar dichotomy when $\mathcal{F}$ is a finite set of non-negative real-valued functions.
- Bulatov, Dyer, Goldberg, Jalsenius and Richerby (2009) extended the Boolean $\#\mathrm{CSP}$ dichotomy to arbitrary real weights.

Dichotomy theorems for $\#CSP(\mathcal{F})$: the Boolean case

- Creignou and Hermann (1996) proved a dichotomy theorem for the case of $\mathcal{F}$ being a finite set of $\{0,1\}$-valued functions.
- Dyer, Goldberg and Jerrum (2009) a similar dichotomy when $\mathcal{F}$ is a finite set of non-negative real-valued functions.
- Bulatov, Dyer, Goldberg, Jalsenius and Richerby (2009) extended the Boolean #CSP dichotomy to arbitrary real weights.
- Cai, Lu and Xia (2014) proved the dichotomy theorem for complex-valued functions.

The dichotomy theorem for Boolean CSPs

Theorem (Schaefer 1978)

Let Γ be a finite set of relations on the domain $\{0, 1\}$. The problem CSP(Γ) is in P if every relation in Γ satisfies one of the conditions below.

The dichotomy theorem for Boolean CSPs

Theorem (Schaefer 1978)

Let Γ *be a finite set of relations on the domain* $\{0,1\}$*. The problem* $CSP(\Gamma)$ *is in P if every relation in* Γ *satisfies one of the conditions below.*

1. *It is 0-valid (it is satisfied when all variables are 0).*

The dichotomy theorem for Boolean CSPs

Theorem (Schaefer 1978)

Let Γ *be a finite set of relations on the domain* $\{0,1\}$*. The problem CSP*(Γ) *is in P if every relation in* Γ *satisfies one of the conditions below.*

1. *It is 0-valid (it is satisfied when all variables are 0).*
2. *It is 1-valid (it is satisfied when all variables are 1).*

The dichotomy theorem for Boolean CSPs

Theorem (Schaefer 1978)

Let Γ be a finite set of relations on the domain $\{0, 1\}$. The problem CSP(Γ) is in P if every relation in Γ satisfies one of the conditions below.

1. *It is 0-valid (it is satisfied when all variables are 0).*
2. *It is 1-valid (it is satisfied when all variables are 1).*
3. *It is Horn (it is equivalent to a conjunction of clauses that contain at most one positive literal).*

The dichotomy theorem for Boolean CSPs

Theorem (Schaefer 1978)

Let Γ be a finite set of relations on the domain $\{0, 1\}$. The problem CSP(Γ) is in P if every relation in Γ satisfies one of the conditions below.

1. *It is 0-valid (it is satisfied when all variables are 0).*
2. *It is 1-valid (it is satisfied when all variables are 1).*
3. *It is Horn (it is equivalent to a conjunction of clauses that contain at most one positive literal).*
4. *It is dual Horn.*

The dichotomy theorem for Boolean CSPs

Theorem (Schaefer 1978)

Let Γ be a finite set of relations on the domain $\{0, 1\}$. The problem CSP(Γ) is in P if every relation in Γ satisfies one of the conditions below.

1. *It is 0-valid (it is satisfied when all variables are 0).*
2. *It is 1-valid (it is satisfied when all variables are 1).*
3. *It is Horn (it is equivalent to a conjunction of clauses that contain at most one positive literal).*
4. *It is dual Horn.*
5. *It is bijunctive (it is equivalent to a conjunction of clauses with at most two literals).*

The dichotomy theorem for Boolean CSPs

Theorem (Schaefer 1978)

Let Γ be a finite set of relations on the domain $\{0, 1\}$. The problem CSP(Γ) is in P if every relation in Γ satisfies one of the conditions below.

1. *It is 0-valid (it is satisfied when all variables are 0).*
2. *It is 1-valid (it is satisfied when all variables are 1).*
3. *It is Horn (it is equivalent to a conjunction of clauses that contain at most one positive literal).*
4. *It is dual Horn.*
5. *It is bijunctive (it is equivalent to a conjunction of clauses with at most two literals).*
6. *It is affine (it is the solution set to a system of linear equations over $\mathbb{F}_2$, i.e. the two-element field).*

The dichotomy theorem for Boolean CSPs

Theorem (Schaefer 1978)

Let Γ be a finite set of relations on the domain $\{0, 1\}$. The problem CSP(Γ) is in P if every relation in Γ satisfies one of the conditions below.

1. *It is 0-valid (it is satisfied when all variables are 0).*
2. *It is 1-valid (it is satisfied when all variables are 1).*
3. *It is Horn (it is equivalent to a conjunction of clauses that contain at most one positive literal).*
4. *It is dual Horn.*
5. *It is bijunctive (it is equivalent to a conjunction of clauses with at most two literals).*
6. *It is affine (it is the solution set to a system of linear equations over $\mathbb{F}_2$, i.e. the two-element field).*

Otherwise, CSP(Γ) is NP-complete.

Easy cases

- MONSAT: Every instance is satisfied when all variables are 1.

Easy cases

- MonSat: Every instance is satisfied when all variables are 1.
- 2Sat: An instance can be transformed into a directed graph. The satisfiability of the input formula is equivalent to an easy-to-check condition in this graph.

Easy cases

- MonSat: Every instance is satisfied when all variables are 1.
- 2Sat: An instance can be transformed into a directed graph. The satisfiability of the input formula is equivalent to an easy-to-check condition in this graph.
- HornSat: There is a polynomial-time algorithm that starts with all variables set to 0 and builds the minimum model of the input Horn formula step by step.

Easy cases

- MONSAT: Every instance is satisfied when all variables are 1.
- 2SAT: An instance can be transformed into a directed graph. The satisfiability of the input formula is equivalent to an easy-to-check condition in this graph.
- HORNSAT: There is a polynomial-time algorithm that starts with all variables set to 0 and builds the minimum model of the input Horn formula step by step.
- XORSAT: The satisfiability of an input formula can be formulated as a system of linear equations over $\mathbb{Z}_2$. For example, $\phi = (x_1 \oplus x_3) \wedge (\neg x_1 \oplus x_2) \wedge (x_1 \oplus x_2 \oplus \neg x_3)$ corresponds to

$$\begin{aligned} x_1 \quad & \quad + x_3 &= 1 \\ 1 + x_1 \quad & + x_2 &= 1 \\ x_1 \quad & + x_2 + 1 + x_3 &= 1 \end{aligned}$$

Easy cases

- MonSat: Every instance is satisfied when all variables are 1.
- 2Sat: An instance can be transformed into a directed graph. The satisfiability of the input formula is equivalent to an easy-to-check condition in the graph.
- HornSat: There is a polynomial-time algorithm that starts with all variables set to 0 and builds the minimum model of the input Horn formula step by step.
- XORSat: The satisfiability of an input formula can be formulated as a system of linear equations over $\mathbb{Z}_2$. For example, $\phi = (x_1 \oplus x_3) \wedge (\neg x_1 \oplus x_2) \wedge (x_1 \oplus x_2 \oplus \neg x_3)$ corresponds to

$$\begin{aligned} x_1 \quad & \quad + x_3 &= 1 \\ x_1 \quad & + x_2 &= 0 \\ x_1 \quad & + x_2 + x_3 &= 0 \end{aligned}$$

Hard cases

- 3-Sat: Every constraint corresponds to a clause with exactly three variables, where at least one literal is true. For example, constraint $C = \{0,1\}^3 \setminus (1,0,0)$ corresponds to $\neg x_1 \vee x_2 \vee x_3$.
- 3-NAE-Sat: Every constraint corresponds to a clause with exactly three variables, where at least one literal is true and not all literals are the same. For example, constraint $C = \{0,1\}^3 \setminus \{(1,0,0),(0,1,1)\}$ corresponds to $\neg x_1 \vee x_2 \vee x_3$.
- 3-Exactly-One: Every constraint corresponds to a clause with exactly three variables, where exactly one literal is true. For example, constraint $C = \{(0,0,0),(1,1,0),(1,0,1)\}$ corresponds to $\neg x_1 \vee x_2 \vee x_3$.

Counting CSPs: $\{0,1\}$-valued functions over the Boolean domain

Theorem (Creignou & Hermann 1996)

Let Γ be a finite set of relations on the domain $\{0,1\}$. The problem $\#CSP(\Gamma)$ is in FP if Γ is affine. Otherwise, $\#CSP(\Gamma)$ is $\#P$-complete.

- Equivalently, this is a dichotomy result for $\#CSP(\mathcal{F})$, where $\mathcal{F}$ consists of $\{0,1\}$-valued functions on Boolean variables.
- Instead of a relation, we consider its characteristic function.
- A function f is affine if its support is an affine relation, where

$$\text{supp}(f) = \{(x_1, ..., x_k) \mid f(x_1, ..., x_k) \neq 0\}.$$

Easy cases

- Every relation in Γ must be affine.
- A relation R is affine if the set of tuples $x \in R$ is the set of solutions to a system of linear equations over $\mathbb{F}_2$.
- These equations are of the form $x_1 \oplus \ldots \oplus x_n = 0$ and $x_1 \oplus \ldots \oplus x_n = 1$.

Easy cases

- Every relation in Γ must be affine.
- A relation R is affine if the set of tuples $x \in R$ is the set of solutions to a system of linear equations over $\mathbb{F}_2$.
- These equations are of the form $x_1 \oplus \ldots \oplus x_n = 0$ and $x_1 \oplus \ldots \oplus x_n = 1$.
- A relation is affine iff $a, b, c \in R$ implies $d = a \oplus b \oplus c \in R$.
- There is an algorithm for determining whether a finite set of relations is affine, so for determining whether $\#\mathrm{CSP}(\Gamma)$ is in FP or #P-complete.

An example of a #P-complete problem

- The relation $IMP = \{(0,0),(0,1),(1,1)\}$ is bijunctive, 0-valid, 1-valid, Horn, dual Horn, but not affine. So, $\#$CSP($\{IMP\}$) is $\#$P-complete.

An example of a $\#$P-complete problem

- The relation $IMP = \{(0,0),(0,1),(1,1)\}$ is bijunctive, 0-valid, 1-valid, Horn, dual Horn, but not affine. So, $\#$CSP($\{IMP\}$) is $\#$P-complete.
- $\#$CSP($\{IMP\}$) is equivalent to $\#2\textsc{HornSat}$. For example an instance of this problem is $(\neg x_1 \vee x_2) \wedge (\neg x_3 \vee x_1) \wedge (\neg x_2 \vee x_4)$.

An example of a #P-complete problem

- The relation $IMP = \{(0,0),(0,1),(1,1)\}$ is bijunctive, 0-valid, 1-valid, Horn, dual Horn, but not affine. So, #CSP($\{IMP\}$) is #P-complete.
- #CSP($\{IMP\}$) is equivalent to $\#2\text{HornSat}$. For example an instance of this problem is $(\neg x_1 \vee x_2) \wedge (\neg x_3 \vee x_1) \wedge (\neg x_2 \vee x_4)$.
- It can also be expressed as a counting homomorphism problem with

H 

and adjacency matrix $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$.

An example of a #P-complete problem

- The relation $IMP = \{(0,0),(0,1),(1,1)\}$ is bijunctive, 0-valid, 1-valid, Horn, dual Horn, but not affine. So, #CSP($\{IMP\}$) is #P-complete.

- #CSP($\{IMP\}$) is equivalent to #2HornSat. For example an instance of this problem is $(\neg x_1 \vee x_2) \wedge (\neg x_3 \vee x_1) \wedge (\neg x_2 \vee x_4)$.

- It can also be expressed as a counting homomorphism problem with

  H 

and adjacency matrix $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$.

- Given an acyclic directed graph G, which defines a partial order, Hom(G, H) is equal to the number of downsets of the partial order (i.e. downward closed sets w.r.t. the partial order) (Exercise).

Generalization to non-negative real weights

Definition 1

A function has product type if it can be expressed as a product of unary functions, binary *Equality* functions ($=_2$), and binary *Disequality* functions ($\neq_2$), on not necessarily disjoint subsets of variables.
We denote by $\mathcal{P}$ the set of all functions of product type.

Definition 2

A function is pure affine if its support is an affine relation and its range is a subset of $\{0, b\}$ for some $b \in \mathbb{R}_{\geq 0}$.
We denote by $\mathcal{A}$ the set of all pure affine functions.

Counting CSPs: non-negative real-valued functions over the Boolean domain

Theorem (Dyer, Goldberg & Jerrum 2009)

For every finite $\mathcal{F}$ consisting of non-negative-valued functions, $\#\text{CSP}(\mathcal{F})$ is in FP if one of the following conditions holds.

1. *Every function in $\mathcal{F}$ is of product type.*
2. *Every function in $\mathcal{F}$ is a pure affine function.*

Otherwise, $\#\text{CSP}(\mathcal{F})$ is $\#$P-hard.

Dichotomy theorems for $\#\text{CSP}(\mathcal{F})$: the case of arbitrary finite domain

- Bulatov (2008) gave a dichotomy for exact counting in the whole of unweighted $\#$CSP.
  (general domain, $\{0, 1\}$-valued)

Dichotomy theorems for $\#\mathrm{CSP}(\mathcal{F})$: the case of arbitrary finite domain

- Bulatov (2008) gave a dichotomy for exact counting in the whole of unweighted #CSP.
  (general domain, $\{0, 1\}$-valued)
- Dyer and Richerby (2011) have given an easier proof of this theorem. They established a new criterion for the #CSP dichotomy and proved that this property is decidable.
  (general domain, $\{0, 1\}$-valued and decidable)

Dichotomy theorems for $\#\mathrm{CSP}(\mathcal{F})$: the case of arbitrary finite domain

- Bulatov (2008) gave a dichotomy for exact counting in the whole of unweighted #CSP.
  (general domain, $\{0, 1\}$-valued)
- Dyer and Richerby (2011) have given an easier proof of this theorem. They established a new criterion for the #CSP dichotomy and proved that this property is decidable.
  (general domain, $\{0, 1\}$-valued and decidable)
- Cai, Chen and Lu (2011) gave an effective complexity dichotomy for non-negative real-valued functions.
  (general domain, non-negative-valued)

Dichotomy theorems for #CSP($\mathcal{F}$): the case of arbitrary finite domain

- Bulatov (2008) gave a dichotomy for exact counting in the whole of unweighted #CSP.
  (general domain, $\{0,1\}$-valued)
- Dyer and Richerby (2011) have given an easier proof of this theorem. They established a new criterion for the #CSP dichotomy and proved that this property is decidable.
  (general domain, $\{0,1\}$-valued and decidable)
- Cai, Chen and Lu (2011) gave an effective complexity dichotomy for non-negative real-valued functions.
  (general domain, non-negative-valued)
- Cai and Chen (2011) gave a dichotomy result for complex-valued functions. Decidability is an open question.
  (general domain, complex-valued, not known to be decidable)

Gödel prize 2021

- **Bulatov (2008) gave a dichotomy for exact counting in the whole of unweighted #CSP.**
- **Dyer and Richerby (2011) have given an easier proof of this theorem. They established a new criterion for the #CSP dichotomy and proved that this property is decidable.**
- Cai, Chen and Lu (2011) gave a dichotomy result for non-negative real-valued functions.
- **Cai and Chen (2011) gave a dichotomy result for complex-valued functions. Decidability is an open question.**

Dichotomy theorems for Holant Problems

- Cai, Lu, Xia (2009): Symmetric complex-valued Holant* on Boolean domain. ($\text{Holant}^*(\mathcal{F}) = \text{Holant}(\mathcal{F} \cup \mathcal{U})$)
- Huang and Lu (2012): Symmetric real-valued Holant on Boolean domain.
- Cai, Guo and Williams (2013): Symmetric complex-valued Holant on Boolean domain.
- Backens (2017): Complex-valued Holant^c on Boolean domain. ($\text{Holant}^c(\mathcal{F}) = \text{Holant}(\mathcal{F} \cup \{\Delta_0, \Delta_1\})$)
- Lin and Wang (2017): Non-negative-valued Holant on Boolean domain.

Dichotomy theorems for Holant Problems

- Cai, Lu, Xia (2009): Symmetric complex-valued Holant* on Boolean domain. ($\text{Holant}^*(\mathcal{F}) = \text{Holant}(\mathcal{F} \cup \mathcal{U})$)
- Huang and Lu (2012): Symmetric real-valued Holant on Boolean domain.
- Cai, Guo and Williams (2013): Symmetric complex-valued Holant on Boolean domain.
- Backens (2017): Complex-valued Holant^c on Boolean domain. ($\text{Holant}^c(\mathcal{F}) = \text{Holant}(\mathcal{F} \cup \{\Delta_0, \Delta_1\})$)
- Lin and Wang (2017): Non-negative-valued Holant on Boolean domain.

- Cai, Luo, Xia (2013): Holant* with domain of size 3 and a single complex-valued ternary symmetric constraint.
- Cai, Guo, Williams (2014): Edge k-Colorings is #P-hard over planar r-regular graphs for $k \geq r \geq 3$ ($r \leq 5$). It is polynomial-time computable for all other values of k and r.

Overview

Overview

Exact counting

Exact counting is rare:

- #2-Colorings.
- #Perfect Matchings in planar graphs.
- #Spanning trees in general graphs.

Approximate counting

Definition

A fully polynomial randomized approximation scheme (fpras) for a counting problem $f : \Sigma^* \to \mathbb{N}$ is a randomized algorithm that takes as input an instance $x \in \Sigma^*$, an error tolerance $0 < \varepsilon < 1$, and $0 < \delta < 1$, and outputs a number $\widehat{f(x)} \in \mathbb{N}$ such that

$$\Pr[(1-\varepsilon)f(x) \leq \widehat{f(x)} \leq (1+\varepsilon)f(x)] \geq 1-\delta.$$

The algorithm must run in time polynomial in $|x|$, $1/\varepsilon$ and $\log(1/\delta)$.

Alternatively we could define the fpras so that on input (x, ε) outputs $\widehat{f(x)}$ satisfying

$$\Pr[(1-\varepsilon)f(x) \le \widehat{f(x)} \le (1+\varepsilon)f(x)] \ge \frac{3}{4} \quad \text{or}$$

$$\Pr[e^{-\varepsilon}f(x) \le \widehat{f(x)} \le e^{\varepsilon}f(x)] \ge \frac{3}{4}$$

The probability $\frac{3}{4}$ can be boosted to $1-\delta$ for any desired $\delta > 0$ using $\mathcal{O}(\log \delta^{-1})$ repeated trials.

Boost the success probability to $1 - \delta$

Chernoff bound

Let $X_1, ..., X_m$ be independent, identically distibuted $\{0, 1\}$ random variables, where $p = \mathbb{E}[X_i]$. For all $\varepsilon \leq 3/2$,

$$\Pr[|\sum_{i=1}^{n} X_i - pn| > \varepsilon pn] \leq 2\exp(-\varepsilon^2 pn/3).$$

- Let an fpras for $f(x)$ such that

$$\Pr[(1-\varepsilon)f(x) \leq \widehat{f(x)} \leq (1+\varepsilon)f(x)] = \frac{3}{4}.$$

Boost the success probability to $1 - \delta$

Chernoff bound

Let $X_1, \ldots, X_m$ be independent, identically distibuted $\{0, 1\}$ random variables, where $p = \mathbb{E}[X_i]$. For all $\varepsilon \leq 3/2$,

$$\Pr[|\sum_{i=1}^{n} X_i - pn| > \varepsilon pn] \leq 2\exp(-\varepsilon^2 pn/3).$$

- Let an fpras for $f(x)$ such that

$$\Pr[(1-\varepsilon)f(x) \leq \widehat{f(x)} \leq (1+\varepsilon)f(x)] = \frac{3}{4}.$$

- Then run this algorithm for $k = 36\log(2/\delta)$ times, obtaining outputs $y_1, \ldots, y_k$. Output the median of these outputs, let's say y_m.

Boost the success probability to $1 - \delta$

Chernoff bound

Let $X_1, ..., X_m$ be independent, identically distibuted $\{0, 1\}$ random variables, where $p = \mathbb{E}[X_i]$. For all $\varepsilon \leq 3/2$,

$$\Pr[|\sum_{i=1}^{n} X_i - pn| > \varepsilon pn] \leq 2\exp(-\varepsilon^2 pn/3).$$

- Let an fpras for $f(x)$ such that

$$\Pr[(1-\varepsilon)f(x) \leq \widehat{f(x)} \leq (1+\varepsilon)f(x)] = \frac{3}{4}.$$

- Then run this algorithm for $k = 36\log(2/\delta)$ times, obtaining outputs $y_1, ..., y_k$. Output the median of these outputs, let's say y_m.
- Let $X_i = \begin{cases} 1, & \text{if } y_i \in (1 \pm \varepsilon)f(x) \\ 0, & \text{otherwise} \end{cases}.$

Boost the success probability to $1 - \delta$

Chernoff bound

Let $X_1, \dots, X_m$ be independent, identically distibuted $\{0, 1\}$ random variables, where $p = \mathbb{E}[X_i]$. For all $\varepsilon \leq 3/2$,

$$\Pr[|\sum_{i=1}^{n} X_i - pn| > \varepsilon pn] \leq 2\exp(-\varepsilon^2 pn/3).$$

- Let an fpras for $f(x)$ such that

$$\Pr[(1-\varepsilon)f(x) \leq \widehat{f(x)} \leq (1+\varepsilon)f(x)] = \frac{3}{4}.$$

- Then run this algorithm for $k = 36\log(2/\delta)$ times, obtaining outputs $y_1, \dots, y_k$. Output the median of these outputs, let's say y_m.
- Let $X_i = \begin{cases} 1, & \text{if } y_i \in (1 \pm \varepsilon)f(x) \\ 0, & \text{otherwise} \end{cases}$.
- $\mathbb{E}[X_i] = \frac{3}{4}$ and $\mathbb{E}[\sum X_i] = \frac{3}{4}k$.

Boost the success probability to $1 - \delta$

Chernoff

Let $X_1, \ldots, X_m$ be independent, identically distibuted $\{0, 1\}$ random variables, where $p = \mathbb{E}[X_i]$. For all $\varepsilon \leq 3/2$,

$$\Pr[|\sum_{i=1}^{n} X_i - pn| > \varepsilon pn] \leq 2\exp(-\varepsilon^2 pn/3).$$

$$\begin{aligned}
\Pr[y_m \notin (1 \pm \varepsilon) f(x)] &\leq \Pr[\sum_{i=1}^{k} X_i < \frac{k}{2}] \\
&\leq \Pr[|\sum X_i - \mathbb{E}[\sum X_i]| > \frac{k}{4}] \\
&\leq \Pr[|\sum X_i - \frac{3}{4}k| > \frac{1}{3} \cdot \frac{3}{4} \cdot k] \\
&\leq 2\exp\left(-\frac{\left(\frac{1}{3}\right)^2 \frac{3}{4}k}{3}\right) = 2\exp(-k/36) = \delta.
\end{aligned}$$

Uniform sampling

- A sampling problem is specified by a relation $R \subseteq \Sigma^* \times \Sigma^*$ between problem instances and solutions, i.e. $(x, w) \in R$ iff w is a solution for the problem instance x.
- We denote the solution set $\{w \mid (x, w) \in R\}$ by $R(x)$.
- A uniform sampler for a solution set $R \in \Sigma^* \times \Sigma^*$ is a randomized algorithm that takes as input an instance $x \in \Sigma^*$ and outputs a solution $W \in R(x)$ uniformly at random.

Total variation distance

To define *approximate sampling* we first need to define the following notion of distance between two probability distributions.

Definition

For two probability distributions μ and ν on a countable set Ω, define the total variation distance between π and π' to be

$$||\mu - \nu||_{TV} = \frac{1}{2} \sum_{\omega \in \Omega} |\mu(\omega) - \nu(\omega)|.$$

Claim

For two probability distributions μ and ν,

$$||\mu - \nu||_{TV} = \max_{A \subseteq \Omega} |\mu(A) - \nu(A)|.$$

For two probability distributions μ and ν,

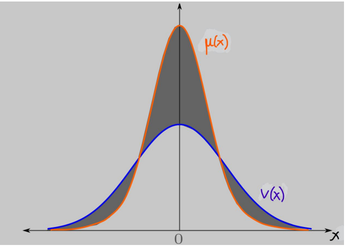
$$||\mu - \nu||_{TV} = \max_{A \subseteq \Omega} |\mu(A) - \nu(A)|.$$

Proof.

Let $A = \{x \mid \mu(x) \geq \nu(x)\}$.
Since μ, ν are probability distributions

$$\sum_{x \in A} \mu(x) - \nu(x) = \sum_{x \notin A} \nu(x) - \mu(x) =$$

$$\frac{1}{2}||\mu - \nu||_1 = ||\mu - \nu||_{TV}.$$



For any set $B \neq A$,

$$\sum_{x \in B} \mu(x) - \nu(x) \leq \sum_{x \in A} \mu(x) - \nu(x).$$

□