Parameterized Analogues of Probabilistic Computation

Ankit Chauhan Raghavendra Rao B.V.

Department of Computer Science and Engineering
IIT Madras

February 8, 2015

Outline

Outline

Why Parameterized Complexity?

- Classical complexity theory analyzes and classifies problems by time or space of input.

Why Parameterized Complexity?

- Classical complexity theory analyzes and classifies problems by time or space of input.
- Sometimes considering parameters based on structural properties makes problems easier.

Why Parameterized Complexity?

- Classical complexity theory analyzes and classifies problems by time or space of input.
- Sometimes considering parameters based on structural properties makes problems easier.
- Parameterized Complexity analyse problem with an additional parameter which depend on input instance in some way.

Why Parameterized Complexity?

- Classical complexity theory analyzes and classifies problems by time or space of input.
- Sometimes considering parameters based on structural properties makes problems easier.
- Parameterized Complexity analyse problem with an additional parameter which depend on input instance in some way.
- Provide more fine-tuned analysis of harder problems.

Motivating Examples

- Evaluating a database query. Database is normally very big and query size is small. Query size is parameter.

Motivating Examples

- Evaluating a database query. Database is normally very big and query size is small. Query size is parameter.
- Solvable in $O(n^k)$, where n is database size and k is the parameter.

Motivating Examples

- Evaluating a database query. Database is normally very big and query size is small. Query size is parameter.
- Solvable in $O(n^k)$, where n is database size and k is the parameter.
- Generally k is very small as compared to database size.

Parameterized Complexity

Definition (Parameterized Problem)

parameterized problem is a set $P \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet. If $(x, k) \in \Sigma^* \times \mathbb{N}$ is an instance of a parameterized problem, we refer to x as the input and to k as the parameter.

Definition (Fixed-Parameter Tractable or *fpt* Problem)

A parameterized problem $P \subseteq \Sigma^* \times N$ is fixed-parameter tractable if there is an algorithm that solves P in $f(k).p(n)$ time where $f(k)$ is an arbitrary computable function.

Parameterized Complexity

Definition (Parameterized Problem)

parameterized problem is a set $P \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet. If $(x, k) \in \Sigma^* \times \mathbb{N}$ is an instance of a parameterized problem, we refer to x as the input and to k as the parameter.

Definition (Fixed-Parameter Tractable or *fpt* Problem)

A parameterized problem$P \subseteq \Sigma^* \times N$ is fixed-parameter tractable if there is an algorithm that solves P in $f(k).p(n)$ time where $f(k)$ is an arbitrary computable function.

- *p-VC* is in FPT. [Downey-Fellow '99]

W-Hierarchy

Problem ($p - WSat(C)$)

Instance: *A circuit C and $k \in \mathbb{N}$.*
Parameter: *k*
Problem: *Decide whether C is k-satisfiable.*

W-Hierarchy

Problem ($p - WSat(C)$)

Instance: A circuit C and $k \in \mathbb{N}$.
Parameter: k
Problem: Decide whether C is k-satisfiable.

Definition (W[P])

$\mathrm{W[P]} = [p\text{-}WSat(C)]^{FPT}$.

W-Hierarchy

Definition (W-hierarchy)

$W[t] = [p - WSat(C_{(t)}, t \geq 1]^{FPT}$, where t is the maxinmum number of unbounded fan-in gate in a path from leave to root.

W-Hierarchy

Definition (W-hierarchy)

$W[t] = [p - WSat(C_{(t)}, t \geq 1]^{FPT}$, where t is the maxinmum number of unbounded fan-in gate in a path from leave to root.

- $p - Clique$, $p - Independent\ Set$ are $W[1]$ complete.

W-Hierarchy

Definition (W-hierarchy)

$W[t] = [p - WSat(C_{(t)}, t \geq 1]^{FPT}$, where t is the maxinmum number of unbounded fan-in gate in a path from leave to root.

- $p - Clique$, $p - Independent\ Set$ are $W[1]$ complete.
- $p - dominating\ Set$, $p - Hitting\ Set$ are $W[2]$ Complete.

Outline

Parameterized Counting Complexity

Definition (Parameterized Counting Problem)

A *parameterized Counting problem* is a set $F : \Sigma^* \times \mathbb{N} \to \mathbb{N}$, where Σ is alphabet set.

Parameterized Counting Complexity

Definition (Parameterized Counting Problem)

A *parameterized Counting problem* is a set $F : \Sigma^* \times \mathbb{N} \to \mathbb{N}$, where Σ is alphabet set.

Definition (Fixed-Parameter Tractable Counting Problem)

A counting parameterized problem is fixed-parameter tractable if there an algorithm computing $F(x, k)$ in at most $f(k)|x|^c$ steps, where f is an arbitrary computable function.

Parameterized Counting Complexity

Definition (Parameterized Counting Problem)

A *parameterized Counting problem* is a set $F : \Sigma^* \times \mathbb{N} \to \mathbb{N}$, where Σ is alphabet set.

Definition (Fixed-Parameter Tractable Counting Problem)

A counting parameterized problem is fixed-parameter tractable if there an algorithm computing $F(x, k)$ in at most $f(k)|x|^c$ steps, where f is an arbitrary computable function.

- $p\text{-}\#VC$ are Fixed-Parameter Tractable Counting Problem i.e, in FPT

#W-Hierarchy

Problem ($p - \#WSat(C)$)

Instance: *A circuit C and $k \in \mathbb{N}$.*
Parameter: *k*
Problem: *To count the k-satisfiable assignment in C.*

Definition (#W[P])

$\#\mathrm{W[P]} = [p - \#WSat(C)]^{FPT}$.

Machine Characterization of W[P]

Definition (k-restricted T.M.)

A k-restricted Turing machine is a non-deterministic $g(k)\text{poly}(n)$ time bounded Turing machine that makes at most $f(k)\log n$ non-deterministic moves, where f and g are arbitrary computable functions.

Machine Characterization of W[P]

Definition (k-restricted T.M.)

A k-restricted Turing machine is a non-deterministic $g(k)\text{poly}(n)$ time bounded Turing machine that makes at most $f(k)\log n$ non-deterministic moves, where f and g are arbitrary computable functions.

- W[P] is the class of all parameterized problems (Q, k) that can be decided by a k-restricted non-deterministic Turing machine.[Chen,Flum,Grohe '03]

#W[P]

Definition (#W[P])

A parameterized counting function (f, k) over the alphabet Σ is in #W[P] if there is a k-restricted non-deterministic Turing machine M such that $f(x, k) = \#\mathsf{acc}_M(x, k)$.

Probabilistic Computation

Definition (W[P]-PFPT)

Let L be a parametrized language. L is said to be in the class W[P]-PFPT if and only if if there is a k-restricted probabilistic Turing machine M such that for any $(x, k) \in \Sigma^* \times \mathbb{N}$ we have,

$$(x, k) \in L \Rightarrow \Pr[M \text{ accepts } (x, k)] > \frac{1}{2}; \text{ and}$$

$$(x, k) \notin L \Rightarrow \Pr[M \text{ accepts } (x, k)] \leq \frac{1}{2}$$

where the probabilities are over the random choices made by M.

Probabilistic Computation

Definition (Diff-FPT, Gap-FPT)

A parametrized function $f : \Sigma^* \times k \to \mathbb{Z}$ is said to be in Diff-FPT if there are two functions $g, h \in \#\mathrm{W[P]}$ such that $f(x,k) = g(x,k) - h(x,k)$.

f is said to be in Gap-FPT if there is a k-restricted TM M such that $f(x,k) = \#\mathrm{acc}_M(x,k) - \#\mathrm{rej}_M(x,k)$, $\forall (x,k) \in \Sigma^* \times k$.

Probabilistic Computation

Definition (Diff-FPT, Gap-FPT)

A parametrized function $f : \Sigma^* \times k \to \mathbb{Z}$ is said to be in Diff-FPT if there are two functions $g, h \in \#\mathrm{W[P]}$ such that $f(x,k) = g(x,k) - h(x,k)$.
f is said to be in Gap-FPT if there is a k-restricted TM M such that $f(x,k) = \#\mathrm{acc}_M(x,k) - \#\mathrm{rej}_M(x,k)$, $\forall (x,k) \in \Sigma^* \times k$.

Lemma

1. Gap-FPT = Diff-FPT
2. Gap-FPT *is closed under taking p-bounded summations and products, i.e., if* $g_1, \ldots, g_{t(k)} \in$ Gap-FPT, *then so are* $g_1 + g_2 \cdots + g_{t(k)}$ *and* $g_1 \times g_2 \times \cdots \times g_{t(k)}$

Probabilistic Computation

Theorem (Structural Properties)

Let L be a parameterized language. The following are equivalent:

Probabilistic Computation

Theorem (Structural Properties)

Let L be a parameterized language. The following are equivalent:

1. $L \in$ W[P]-PFPT.

Probabilistic Computation

Theorem (Structural Properties)

Let L be a parameterized language. The following are equivalent:

1. $L \in$ W[P]-PFPT.
2. *There is a $k-$restricted Turing machine M such that $(x,k) \in L \iff \#\text{accept}_M(x,k) - \#\text{reject}_M(x,k) > 0$.*

Probabilistic Computation

Theorem (Structural Properties)

Let L be a parameterized language. The following are equivalent:

1. $L \in$ W[P]-PFPT.
2. *There is a $k-$restricted Turing machine M such that* $(x,k) \in L \iff \#\text{accept}_M(x,k) - \#\text{reject}_M(x,k) > 0$.
3. *There is a function $f \in$ Gap-FPT such that* $(x,k) \in L \iff f(x,k) > 0$

Probabilistic Computation

Theorem (Structural Properties)

Let L be a parameterized language. The following are equivalent:

1. $L \in$ W[P]-PFPT.
2. *There is a $k-$restricted Turing machine M such that* $(x,k) \in L \iff \#\text{accept}_M(x,k) - \#\text{reject}_M(x,k) > 0$.
3. *There is a function $f \in$ Gap-FPT such that* $(x,k) \in L \iff f(x,k) > 0$
4. *There is a $B \in$ FPT, and $P(n,k) = f(k)\log n$ such that* $(x,k) \in L \iff |\{y \in \{0,1\}^{P(n,k)} \mid (x,y,k) \in B\}| \geq 2^{P(n,k)-1} + 1$.

Probabilistic Computation

Theorem (Closure properties)

1. W[P]-PFPT *is closed under complementation.*
2. W[P]-PFPT *is closed under symmetric difference.*

Theorem

$\mathsf{FPT}^{\#\mathsf{W[P]}} = \mathsf{FPT}^{\mathsf{W[P]\text{-}PFPT}}$

Outline

Definitions

Definition (Arithmetic Circuit)

An *arithmetic circuit* C is a directed acyclic graph where nodes of in-degree zero are called *input* gates and are labelled by $\{-1, 0, 1\} \cup \{x_1, \ldots, x_n\}$ where $x_1, \ldots, x_n$ are the input variables. The remaining gates are labelled $\times$ or $+$.

Definitions

Definition (Arithmetic Circuit)

An *arithmetic circuit* C is a directed acyclic graph where nodes of in-degree zero are called *input* gates and are labelled by $\{-1, 0, 1\} \cup \{x_1, \ldots, x_n\}$ where $x_1, \ldots, x_n$ are the input variables. The remaining gates are labelled $\times$ or $+$.

Definition (Syntactic Degree)

Every gate of an arithmetic circuit C. For a leaf node v, $\text{syntdeg}(v) = 1$. If $v = v_1 + v_2$ then $\text{syntdeg}(v) = \max\{\text{syntdeg}(v_1), \text{syntdeg}(v_2)\}$ and if $v = v_1 \times v_2$ then $\deg(v) = \text{syntdeg}(v_1) + \text{syntdeg}(v_2)$

Parameterized Random Classes

Definition (RFPT)

Class of languages for which there exist f(k)poly(n) time bounded randomized algorithm accepting languages with bounded one-sided error probability.

Definition (W[P]-RFPT)

Class of languages for which there exist f(k)poly(n) time bounded randomized algorithm accepting languages with bounded one-sided error probability and $g(k)\log(n)$ probabilistic moves.

Problem Description

Problem (p-acit (Müller'08))

Input: *Arithmetic circuit* C, $\text{syntdeg}(C) \leq k$.
Parameter: k.
Task: *Test if the polynomial computed by* C *is identically zero.*

- Classical randomized algorithm using Schwartz-Zipple Lemma problem will be in RFPT as it will use $O(n \log k)$ probabilistic moves.

Problem Description

Problem (p-acit (Müller'08))

Input: *Arithmetic circuit* C, $\text{syntdeg}(C) \leq k$.
Parameter: k.
Task: *Test if the polynomial computed by* C *is identically zero.*

- Classical randomized algorithm using Schwartz-Zipple Lemma problem will be in RFPT as it will use $O(n \log k)$ probabilistic moves.
- Challenge is to solve this problem using W[P]-RFPT algorithm.

Theorem

Theorem

p-acit *is in* W[P]-RFPT

Idea.

Theorem

Theorem

p-acit *is in* W[P]-RFPT

Idea.

- Modify Circuit using Black Box

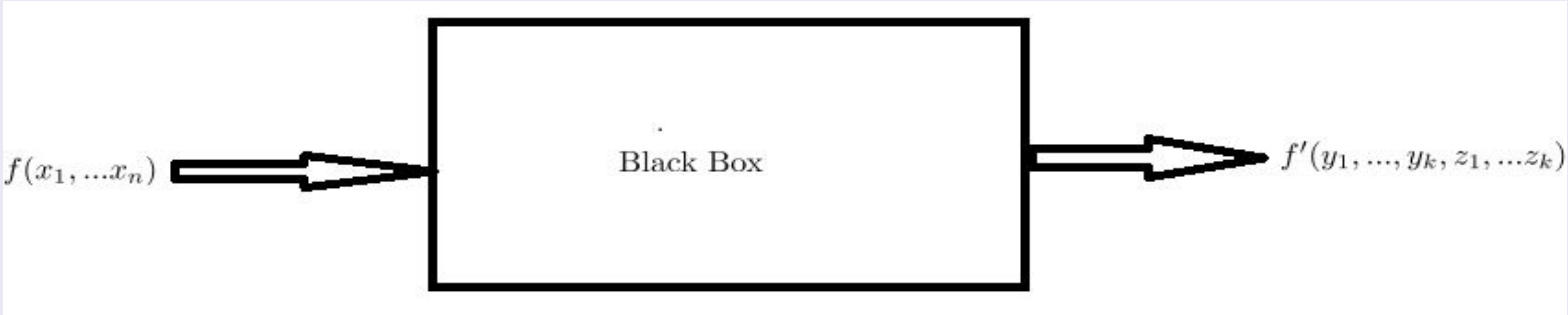
Theorem

Theorem

p-acit *is in* W[P]-RFPT

Idea.

- Modify Circuit using Black Box



- Now apply Schwartz-Zipple Lemma on modified Circuit.

Function of BlackBox

- Let f be an n-variate polynomial of degree at most k. Then

$$f \equiv 0 \iff \forall a \in W_n^k(S), \quad f(a) = 0,$$

where $S \subset \mathbb{K}$ has at least $k+1$ elements and let $W_n^k(S)$ the set of all vectors of Hamming weight at most k.

Function of BlackBox

Definition (Shpilka-Volkovich Hitting set generator)

Let $a_1, \ldots, a_n$ be distinct elements in $\mathbb{K}$. Let $G_k^i \in \mathbb{K}[y_1, \ldots, y_k, z_1, \ldots, z_k]$ be the polynomial defined as follows:

$$G^i(y_1, \ldots, y_k, z_1, \ldots, z_k) = \sum_{j=1}^{k} L_i(y_j) z_j, \quad \text{where } L_i(x) = \frac{\prod_{j \neq i}(x - a_j)}{\prod_{j \neq i}(a_i - a_j)}.$$

The generator G is defined as $G \stackrel{\triangle}{=} (G^1, \ldots, G^n)$.

Function of BlackBox

- Let f be an n-variate polynomial of degree at most k. Then

$$f \equiv 0 \iff \forall a \in W_n^k(S) \quad f(a) = 0,$$

  where $S \subset \mathbb{K}$ has at least $k+1$ elements and let $W_n^k(S)$ the set of all vectors of Hamming weight at most k.

- For any finite set $S \subset \mathbb{K}$, then $W_n^k(S) \subseteq \{(G_k^1(a), \ldots, G_k^n(a)) \mid a \in (S \cup \{a_1, \ldots, a_n\})^{2k}\}$.

Function of BlackBox

- Let f be an n-variate polynomial of degree at most k. Then

$$f \equiv 0 \iff \forall a \in W_n^k(S) \quad f(a) = 0,$$

  where $S \subset \mathbb{K}$ has at least $k+1$ elements and let $W_n^k(S)$ the set of all vectors of Hamming weight at most k.

- For any finite set $S \subset \mathbb{K}$, then $W_n^k(S) \subseteq \{(G_k^1(a), \ldots, G_k^n(a)) \mid a \in (S \cup \{a_1, \ldots, a_n\})^{2k}\}$.

- Let f be a polynomial of degree at most k. Then $f \equiv 0 \iff f(G_k) \equiv 0$.

Outline

Parameterized Permanant vs Parameterized Determinent

Definition (Permanant , Determinant)

$$\det(A) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)} \quad (1)$$

$$\operatorname{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}, \quad (2)$$

Definition

A permutation $\sigma \in S_n$ is said to be a *k-permutation*, if $|\{i \mid \sigma(i) \neq i\}| = k$. Let $S_{n,k}$ denote the set of all k-permutations on n symbols.

Parameterized Permanant vs Parameterized Determinent

Definition

Let k be a parameter. The parametrized determinent function and parametrized permanent function p-perm of a matrix $A \in \mathbb{Z}^{n \times n}$ is defined as follows:

$$\text{p-det}(A) = \sum_{\sigma \in S_n \text{is a } k\text{-permutation}} \text{sgn}(\sigma) \prod_{i \neq \sigma(i)} A_{i\sigma(i)}$$

$$\text{p-perm}(A) = \sum_{\sigma \in S_n \text{is a } k\text{-permutation}} \prod_{i \neq \sigma(i)} A_{i\sigma(i)}$$

Parameterized Permanant vs Parameterized Determinent

Theorem

p-perm *is* $\#W[1]$ *complete. The hardness holds even in the case of 0-1 matrices.*

Theorem

p-det *on integer matrices is fixed parameter tractable*

Conclusion

- Definition of W[P]-PFPT leads to further developments in the structural aspects of probabilistic and counting complexities in the parameterized world.

Conclusion

- Definition of W[P]-PFPT leads to further developments in the structural aspects of probabilistic and counting complexities in the parameterized world.
- W[P]-PFPT might be useful in defining a parameterized variant of the Counting Hierarchy (CH).

Conclusion

- Definition of W[P]-PFPT leads to further developments in the structural aspects of probabilistic and counting complexities in the parameterized world.
- W[P]-PFPT might be useful in defining a parameterized variant of the Counting Hierarchy (CH).
- Is W[P]-PFPTclosed under intersection?

Conclusion

- Definition of W[P]-PFPT leads to further developments in the structural aspects of probabilistic and counting complexities in the parameterized world.
- W[P]-PFPT might be useful in defining a parameterized variant of the Counting Hierarchy (CH).
- Is W[P]-PFPTclosed under intersection?
- W[P]-RFPTalgorithm for p-acit opens a new question whether all RFPT algorithms lies in W[P]-RFPT.

Thank You !!!

Combinatorial Nullstellensatz

Lemma (Alon '99)

Let $P \in \mathbb{K}[x_1, \ldots, x_n]$ be a polynomial where for every $i \in [n]$, the degree of x_i is bounded by t. Let $S \subseteq \mathbb{K}$ be a finite set of size at least $t+1$, and $A = S^n$. Then $P \equiv 0 \iff P(a) = 0,\ \forall a \in A$.