Approximate Categories for the Graph Isomorphism Problem

Harm Derksen

CMS Summer Meeting
June 5, 2010

The Orbit Problem

k a field with algebraic closure $\overline{k}$
G a linear algebraic group defined over k
V a representation of G

The Orbit Problem

k a field with algebraic closure $\overline{k}$
G a linear algebraic group defined over k
V a representation of G

Orbit Problem

Given $v, w \in V$, do v, w lie in the same $G(\overline{k})$-orbit?

The Orbit Problem

k a field with algebraic closure $\overline{k}$
G a linear algebraic group defined over k
V a representation of G

Orbit Problem

Given $v, w \in V$, do v, w lie in the same $G(\overline{k})$-orbit?

Isomorphism problems can be translated to orbit problems.

Example: The Graph Isomorphism Problem

Γ_1, Γ_2 graphs with vertex set $\{1, 2, \ldots, n\}$
$A_1, A_2 \in \mathrm{Mat}_{n,n}(k)$ the adjacency matrices of Γ_1, Γ_2 respectively

Example: The Graph Isomorphism Problem

Γ_1, Γ_2 graphs with vertex set $\{1, 2, \ldots, n\}$
$A_1, A_2 \in \mathrm{Mat}_{n,n}(k)$ the adjacency matrices of Γ_1, Γ_2 respectively

G set of $n \times n$ permutation matrices
G acts on $\mathrm{Mat}_{n,n}(k)$ ($n \times n$ matrices) by conjugation:
$P \cdot A := PAP^{-1}$, $P \in G$, $A \in \mathrm{Mat}_{n,n}(k)$

Example: The Graph Isomorphism Problem

Γ_1, Γ_2 graphs with vertex set $\{1, 2, \dots, n\}$
$A_1, A_2 \in \mathrm{Mat}_{n,n}(k)$ the adjacency matrices of Γ_1, Γ_2 respectively

G set of $n \times n$ permutation matrices
G acts on $\mathrm{Mat}_{n,n}(k)$ ($n \times n$ matrices) by conjugation:
$P \cdot A := PAP^{-1}$, $P \in G$, $A \in \mathrm{Mat}_{n,n}(k)$

Translation of Isomorphism Problem into Orbit Problem

$$\Gamma_1 \cong \Gamma_2 \Leftrightarrow A_1, A_2 \text{ in same } G\text{-orbit}$$

Example: The Graph Isomorphism Problem

Γ_1, Γ_2 graphs with vertex set $\{1, 2, \ldots, n\}$
$A_1, A_2 \in \mathrm{Mat}_{n,n}(k)$ the adjacency matrices of Γ_1, Γ_2 respectively

G set of $n \times n$ permutation matrices
G acts on $\mathrm{Mat}_{n,n}(k)$ ($n \times n$ matrices) by conjugation:
$P \cdot A := PAP^{-1}$, $P \in G$, $A \in \mathrm{Mat}_{n,n}(k)$

Translation of Isomorphism Problem into Orbit Problem

$$\Gamma_1 \cong \Gamma_2 \Leftrightarrow A_1, A_2 \text{ in same } G\text{-orbit}$$

We'll get back to graphs later.

Example: Isomorphism of Modules

$T = k\langle x_1, \ldots, x_r\rangle / I$ associative algebra over k (with 1)
M, N n-dimensional T-modules

Example: Isomorphism of Modules

$T = k\langle x_1, \dots, x_r\rangle / I$ associative algebra over k (with 1)
M, N n-dimensional T-modules

$x_i \cdot : M \to M$ given by matrix $A_i \in \mathrm{Mat}_{n,n}(k)$
$x_i \cdot : N \to N$ given by matrix $B_i \in \mathrm{Mat}_{n,n}(k)$
$A = (A_1, \dots, A_r), B = (B_1, \dots, B_r) \in \mathrm{Mat}_{n,n}(k)^r$

Example: Isomorphism of Modules

$T = k\langle x_1, \dots, x_r\rangle / I$ associative algebra over k (with 1)
M, N n-dimensional T-modules

$x_i \cdot : M \to M$ given by matrix $A_i \in \mathrm{Mat}_{n,n}(k)$
$x_i \cdot : N \to N$ given by matrix $B_i \in \mathrm{Mat}_{n,n}(k)$
$A = (A_1, \dots, A_r), B = (B_1, \dots, B_r) \in \mathrm{Mat}_{n,n}(k)^r$

$G = \mathrm{GL}_n(k)$ acts on $\mathrm{Mat}_{n,n}(k)$ by conjugation:

$$U \cdot (C_1, \dots, C_r) = (UC_1U^{-1}, \dots, UC_rU^{-1})$$

Example: Isomorphism of Modules

$T = k\langle x_1, \ldots, x_r\rangle / I$ associative algebra over k (with 1)
M, N n-dimensional T-modules

$x_i \cdot : M \to M$ given by matrix $A_i \in \mathrm{Mat}_{n,n}(k)$
$x_i \cdot : N \to N$ given by matrix $B_i \in \mathrm{Mat}_{n,n}(k)$
$A = (A_1, \ldots, A_r), B = (B_1, \ldots, B_r) \in \mathrm{Mat}_{n,n}(k)^r$
$G = \mathrm{GL}_n(k)$ acts on $\mathrm{Mat}_{n,n}(k)$ by conjugation:

$$U \cdot (C_1, \ldots, C_r) = (UC_1U^{-1}, \ldots, UC_rU^{-1})$$

Isomorphism Test

$$M \cong N \Leftrightarrow A, B \text{ in same } G\text{-orbit}$$

Example: Isomorphism of Modules

$T = k\langle x_1, \dots, x_r\rangle / I$ associative algebra over k (with 1)
M, N n-dimensional T-modules

$x_i \cdot : M \to M$ given by matrix $A_i \in \mathrm{Mat}_{n,n}(k)$
$x_i \cdot : N \to N$ given by matrix $B_i \in \mathrm{Mat}_{n,n}(k)$
$A = (A_1, \dots, A_r), B = (B_1, \dots, B_r) \in \mathrm{Mat}_{n,n}(k)^r$
$G = \mathrm{GL}_n(k)$ acts on $\mathrm{Mat}_{n,n}(k)$ by conjugation:

$$U \cdot (C_1, \dots, C_r) = (UC_1U^{-1}, \dots, UC_rU^{-1})$$

Isomorphism Test

$$M \cong N \Leftrightarrow A, B \text{ in same } G\text{-orbit}$$

Remark

$$A, B \text{ in same } G(k)\text{-orbit} \Leftrightarrow A, B \text{ in same } G(\overline{k})\text{-orbit}$$

Complexity of the Module Isiomorphism Problem

Theorem (Chistov–Invanyos–Karpinski '97, Brooksbank–Luks '08)

There exists a T-module isomorphism test that requires only a polynomial number (in the dimension of the modules) of arithmetic operations in the field k.

Isomorphism Test using Ideals

G linear algebraic group
$k[G]$ coordinate ring of G over k
V representation of G
$v, w \in V$

Isomorphism Test using Ideals

G linear algebraic group
$k[G]$ coordinate ring of G over k
V representation of G
$v, w \in V$

$g \cdot v = w$ gives a system of polynomial equations for $g \in G$
Let $I \subseteq k[G]$ be the ideal generated by these polynomials

Isomorphism Test using Ideals

G linear algebraic group
$k[G]$ coordinate ring of G over k
V representation of G
$v, w \in V$

$g \cdot v = w$ gives a system of polynomial equations for $g \in G$
Let $I \subseteq k[G]$ be the ideal generated by these polynomials

Isomorphism Test

v, w in the distinct G-orbits $\Leftrightarrow 1 \in I$

Isomorphism Test using Gröbner Bases

If G is fixed, then one can test whether $1 \in I$ efficiently: the number of arithmetic operations in k required is polynomial in n and the degrees of the polynomials defining the representation V.

Isomorphism Test using Gröbner Bases

If G is fixed, then one can test whether $1 \in I$ efficiently: the number of arithmetic operations in k required is polynomial in n and the degrees of the polynomials defining the representation V.

In many interesting examples, such as the graph isomorphism problem, G is not fixed.

Isomorphism Test using Gröbner Bases

If G is fixed, then one can test whether $1 \in I$ efficiently: the number of arithmetic operations in k required is polynomial in n and the degrees of the polynomials defining the representation V.

In many interesting examples, such as the graph isomorphism problem, G is not fixed.

One can use Buchberger's algorithm to test whether $1 \in I$, but this may not be efficient.

Approximate Categories

(k, G, V as before)
For every d will construct an “approximate” k-category $\mathcal{C}_d(V)$ with the following properties:

Approximate Categories

(k, G, V as before)
For every d will construct an “approximate” k-category $\mathcal{C}_d(V)$ with the following properties:

1. Every element $v \in V$ is an object in $\mathcal{C}_d(V)$

Approximate Categories

(k, G, V as before)
For every d will construct an “approximate” k-category $\mathcal{C}_d(V)$ with the following properties:

1. Every element $v \in V$ is an object in $\mathcal{C}_d(V)$
2. If v, w lie in the same G-orbit, then v and w are isomorphic in $\mathcal{C}_d(V)$

Approximate Categories

(k, G, V as before)
For every d will construct an “approximate” k-category $\mathcal{C}_d(V)$ with the following properties:

1. Every element $v \in V$ is an object in $\mathcal{C}_d(V)$
2. If v, w lie in the same G-orbit, then v and w are isomorphic in $\mathcal{C}_d(V)$
3. If v, w are isomorphic in $\mathcal{C}_{d+1}(V)$, then they are isomorphic in $\mathcal{C}_d(V)$

Approximate Categories

(k, G, V as before)
For every d will construct an "approximate" k-category $\mathcal{C}_d(V)$ with the following properties:

1. Every element $v \in V$ is an object in $\mathcal{C}_d(V)$
2. If v, w lie in the same G-orbit, then v and w are isomorphic in $\mathcal{C}_d(V)$
3. If v, w are isomorphic in $\mathcal{C}_{d+1}(V)$, then they are isomorphic in $\mathcal{C}_d(V)$
4. If v, w are isomorphic in $\mathcal{C}_d(V)$ for all $d \geq 1$, then v and w are in the same G-orbit

Approximate Categories

(k, G, V as before)
For every d will construct an "approximate" k-category $\mathcal{C}_d(V)$ with the following properties:

1. Every element $v \in V$ is an object in $\mathcal{C}_d(V)$
2. If v, w lie in the same G-orbit, then v and w are isomorphic in $\mathcal{C}_d(V)$
3. If v, w are isomorphic in $\mathcal{C}_{d+1}(V)$, then they are isomorphic in $\mathcal{C}_d(V)$
4. If v, w are isomorphic in $\mathcal{C}_d(V)$ for all $d \geq 1$, then v and w are in the same G-orbit
5. There exists an efficient algorithm to determine if v and w are isomorphic in $\mathcal{C}_d(V)$

Truncated Ideals

Suppose that R is a finitely generated commutative k-algebra (with 1) with a filtration

$$R_0 = k \subseteq R_1 \subseteq R_2 \subseteq \cdots$$

Truncated Ideals

Suppose that R is a finitely generated commutative k-algebra (with 1) with a filtration

$$R_0 = k \subseteq R_1 \subseteq R_2 \subseteq \cdots$$

If $S \subseteq R_d$ then we define

$$(S)_d = \sum_{e=0}^{d} (S \cap R_e) R_{d-e}.$$

We call $S \subseteq R_d$ a *d-truncated ideal* if $(S)_d = S$.

Truncated Ideals

Suppose that R is a finitely generated commutative k-algebra (with 1) with a filtration

$$R_0 = k \subseteq R_1 \subseteq R_2 \subseteq \cdots$$

If $S \subseteq R_d$ then we define

$$(S)_d = \sum_{e=0}^{d} (S \cap R_e) R_{d-e}.$$

We call $S \subseteq R_d$ a *d-truncated ideal* if $(S)_d = S$.

The sequence

$$(S)_d \subseteq ((S)_d)_d \subseteq (((S)_d)_d)_d \subseteq \cdots$$

stabilizes to a d-truncated ideal which will be denoted by $((S))_d$.

A nice Filtration for $k[G]$

Let G be a linear algebraic group over k
$G \times G$ acts on $R = k[G]$ by

$$((g,h)\cdot f)(u) = f(g^{-1}uh), \quad f \in R,\ g,h,u \in G$$

A nice Filtration for $k[G]$

Let G be a linear algebraic group over k
$G \times G$ acts on $R = k[G]$ by

$$((g,h)\cdot f)(u) = f(g^{-1}uh), \quad f \in R,\ g,h,u \in G$$

Fix a finite dimensional subspace $W \subseteq R$ such that

A nice Filtration for $k[G]$

Let G be a linear algebraic group over k
$G \times G$ acts on $R = k[G]$ by

$$((g,h)\cdot f)(u) = f(g^{-1}uh), \quad f \in R,\ g,h,u \in G$$

Fix a finite dimensional subspace $W \subseteq R$ such that

1. $k \subseteq W$

A nice Filtration for $k[G]$

Let G be a linear algebraic group over k
$G \times G$ acts on $R = k[G]$ by

$$((g,h)\cdot f)(u) = f(g^{-1}uh), \quad f \in R,\ g,h,u \in G$$

Fix a finite dimensional subspace $W \subseteq R$ such that

1. $k \subseteq W$
2. W is $G \times G$-stable

A nice Filtration for $k[G]$

Let G be a linear algebraic group over k
$G \times G$ acts on $R = k[G]$ by

$$((g, h) \cdot f)(u) = f(g^{-1}uh), \quad f \in R,\ g, h, u \in G$$

Fix a finite dimensional subspace $W \subseteq R$ such that

1. $k \subseteq W$
2. W is $G \times G$-stable
3. W generates R

A nice Filtration for $k[G]$

Let G be a linear algebraic group over k
$G \times G$ acts on $R = k[G]$ by

$$((g,h)\cdot f)(u) = f(g^{-1}uh), \quad f \in R,\ g,h,u \in G$$

Fix a finite dimensional subspace $W \subseteq R$ such that

1. $k \subseteq W$
2. W is $G \times G$-stable
3. W generates R

Define a filtration by $R = \bigcup_d R_d$, where $R_d = W^d$

The Algebra Structure for $R_d^\star$

Let $\Delta : K[G] \to K[G] \otimes K[G]$ be the co-multiplication of $K[G]$

The Algebra Structure for $R_d^\star$

Let $\Delta : K[G] \to K[G] \otimes K[G]$ be the co-multiplication of $K[G]$

Then $\Delta(R_d) \subseteq R_d \otimes R_d$

The Algebra Structure for $R_d^\star$

Let $\Delta : K[G] \to K[G] \otimes K[G]$ be the co-multiplication of $K[G]$

Then $\Delta(R_d) \subseteq R_d \otimes R_d$

So $R_d^\star$ is an associative algebra

The category $\mathcal{C}_d(V)$

Objects

Objects in $\mathcal{C}_d(V)$ are affine subspaces of the form $v + Z$ with $v \in V$ and $Z \subseteq V$ a subspace.

The category $\mathcal{C}_d(V)$

Objects

Objects in $\mathcal{C}_d(V)$ are affine subspaces of the form $v+Z$ with $v \in V$ and $Z \subseteq V$ a subspace.

Suppose that $X_1 = v_1 + Z_1$ and $X_2 = v_2 + Z_2$ are objects. The equation

$$g \cdot X_1 \subseteq X_2$$

gives a system of polynomials $S(X_1, X_2) \subset R_d$
Define $I_d(X_1, X_2) = ((S(X_1, X_2)))_d$

The category $\mathcal{C}_d(V)$

Objects

Objects in $\mathcal{C}_d(V)$ are affine subspaces of the form $v + Z$ with $v \in V$ and $Z \subseteq V$ a subspace.

Suppose that $X_1 = v_1 + Z_1$ and $X_2 = v_2 + Z_2$ are objects. The equation

$$g \cdot X_1 \subseteq X_2$$

gives a system of polynomials $S(X_1, X_2) \subset R_d$
Define $I_d(X_1, X_2) = ((S(X_1, X_2)))_d$

Morphisms

We define $\mathrm{Hom}_d(X_1, X_2) = (R_d/I_d(X_1, X_2))^\star$. The bilinear map $\mathrm{Hom}_d(X_1, X_2) \times \mathrm{Hom}_d(X_2, X_3) \to \mathrm{Hom}_d(X_1, X_3)$ is the restriction of the multiplication $R_d^\star \times R_d^\star \to R_d^\star$.

Isomorphism Testing in $\mathcal{C}_d(V)$

Suppose that X_1, X_2 are objects in $\mathcal{C}_d(V)$ We can test whether X_1 and X_2 are isomorphic as follows:

Isomorphism Testing in $\mathcal{C}_d(V)$

Suppose that X_1, X_2 are objects in $\mathcal{C}_d(V)$ We can test whether X_1 and X_2 are isomorphic as follows:

- $T = \mathrm{Hom}_d(X_1, X_1)$ is a finite dim. associative algebra
  If T and $\mathrm{Hom}_d(X_2, X_1)$ are not isomorphic as T-modules, then X_1 and X_2 are not isomorphic

Isomorphism Testing in $\mathcal{C}_d(V)$

Suppose that X_1, X_2 are objects in $\mathcal{C}_d(V)$ We can test whether X_1 and X_2 are isomorphic as follows:

- $T = \mathrm{Hom}_d(X_1, X_1)$ is a finite dim. associative algebra
  If T and $\mathrm{Hom}_d(X_2, X_1)$ are not isomorphic as T-modules, then X_1 and X_2 are not isomorphic
- We can test whether two T-modules are isomorphic efficiently, and if T and $\mathrm{Hom}_d(X_2, X_1)$ are isomorphic, we can compute an isomorphism $\varphi : \mathrm{Hom}_d(X_1, X_1) \to \mathrm{Hom}_d(X_2, X_1)$

Isomorphism Testing in $\mathcal{C}_d(V)$

Suppose that X_1, X_2 are objects in $\mathcal{C}_d(V)$ We can test whether X_1 and X_2 are isomorphic as follows:

- $T = \mathrm{Hom}_d(X_1, X_1)$ is a finite dim. associative algebra If T and $\mathrm{Hom}_d(X_2, X_1)$ are not isomorphic as T-modules, then X_1 and X_2 are not isomorphic
- We can test whether two T-modules are isomorphic efficiently, and if T and $\mathrm{Hom}_d(X_2, X_1)$ are isomorphic, we can compute an isomorphism $\varphi : \mathrm{Hom}_d(X_1, X_1) \to \mathrm{Hom}_d(X_2, X_1)$
- Let $f = \varphi(\mathrm{id})$. Then X_1 and X_2 are isomorphic if and only if f is an isomorphism. This is easy to test.

The Graph Isomorphism Problem

The Graph isomorphism is in **NP**, but it is not known whether it is in **P**. In other words, it is not known whether there exists an algorithm that can determine if two graphs with n vertices are isomorphic in $O(n^m)$ time, for some fixed m.

The Graph Isomorphism Problem

The Graph isomorphism is in **NP**, but it is not known whether it is in **P**. In other words, it is not known whether there exists an algorithm that can determine if two graphs with n vertices are isomorphic in $O(n^m)$ time, for some fixed m.

If the graphs have bounded valence, then there exists a polynomial time algorithm (Luks '82).

The Graph Isomorphism Problem

The Graph isomorphism is in **NP**, but it is not known whether it is in **P**. In other words, it is not known whether there exists an algorithm that can determine if two graphs with n vertices are isomorphic in $O(n^m)$ time, for some fixed m.

If the graphs have bounded valence, then there exists a polynomial time algorithm (Luks '82).

Another well-known algorithm is the d-dimensional Weisfeiler-Lehman algorithm (60's).

The d-dimensional Weisfeiler-Lehman algorithm

$\Gamma = (X, E)$ Graph, X set with n elements
$E \subseteq X \times X$ symmetric relation

The d-dimensional Weisfeiler-Lehman algorithm

$\Gamma = (X, E)$ Graph, X set with n elements
$E \subseteq X \times X$ symmetric relation

Idea: color i tuples in X^i for $i \leq d$ recursively until a stable coloring is obtained.

The d-dimensional Weisfeiler-Lehman algorithm

$\Gamma = (X, E)$ Graph, X set with n elements
$E \subseteq X \times X$ symmetric relation

Idea: color i tuples in X^i for $i \leq d$ recursively until a stable coloring is obtained.

For fixed d, this algorithm is polynomial time in n.

The d-dimensional Weisfeiler-Lehman algorithm

$\Gamma = (X, E)$ Graph, X set with n elements
$E \subseteq X \times X$ symmetric relation

Idea: color i tuples in X^i for $i \leq d$ recursively until a stable coloring is obtained.

For fixed d, this algorithm is polynomial time in n.

The stable coloring is invariant under $\mathrm{Aut}(X)$. If Γ_1, Γ_2 are distinct graphs, then we can take Γ as the disjoint union. If a vertex of Γ_1 get a color that does not appear in Γ_2, then Γ_1 and Γ_2 are not isomorphic.

d-Variable first order language $\mathbf{L}_d$

We can think of a graph $\Gamma = (X, E)$ as a structure, and to this structure we can associate the first order logic. In the d-variable language $\mathbf{L}_d$, we only allow d variables to be used (but one may re-use variables)

d-Variable first order language $\mathbf{L}_d$

We can think of a graph $\Gamma = (X, E)$ as a structure, and to this structure we can associate the first order logic. In the d-variable language $\mathbf{L}_d$, we only allow d variables to be used (but one may re-use variables)

For example:

$$\varphi(x_1, x_2) = \exists x_3 \, [\exists x_2 \, E(x_1, x_2) \wedge E(x_2, x_3)] \wedge E(x_3, x_2)$$

says "x_1 and x_2 are connected by a path of length 3". The formula uses 3 variables (x_2 has been re-used).

d-Variable Logic with Counting $\mathbf{C}_d$

In the d-variable first order language with counting $\mathbf{C}_d$, we allow also quantors that can count.

d-Variable Logic with Counting $\mathbf{C}_d$

In the d-variable first order language with counting $\mathbf{C}_d$, we allow also quantors that can count.

$\exists_l\, x$ means "there exist exactly l values for x such that …"

d-Variable Logic with Counting $\mathbf{C}_d$

In the *d*-variable first order language with counting $\mathbf{C}_d$, we allow also quantors that can count.

$\exists_l\, x$ means "there exist exactly l values for x such that ..."

For example

$$\psi(x_1) = \exists_{37} x_2\, \varphi(x_1, x_2)$$

means " there are exactly 37 vertices that can be connected to x_1 by a path of length 3".

Cai–Fürer–Immerman ’92

Theorem

The *d*-dimensional Weisfeiler-Lehman algorithm can distinguish two graphs Γ_1, Γ_2 if and only if there exists a closed formula ψ in the $(d + 1)$-variable logic with counting such that ψ is true for Γ_1 but not for Γ_2.

Cai–Fürer–Immerman '92

Theorem

The d-dimensional Weisfeiler-Lehman algorithm can distinguish two graphs Γ_1, Γ_2 if and only if there exists a closed formula ψ in the $(d+1)$-variable logic with counting such that ψ is true for Γ_1 but not for Γ_2.

Theorem (CFI)

For every d there exists two non-isomorphic graphs Γ_1 and Γ_2 such that for every formula ψ in $\mathbf{C}_{d+1}$, ψ is true for Γ_1 if and only if ψ is true for Γ_2. So the d-dimensional Weisfeiler-Lehman algorithm cannot distinguish Γ_1 and Γ_2.

Distinguishing Graphs using the category $\mathcal{C}_d(V)$

Γ_1, Γ_2 two graphs with n vertices
A_1, A_2 corresponding adjacency matrices

Distinguishing Graphs using the category $\mathcal{C}_d(V)$

Γ_1, Γ_2 two graphs with n vertices
A_1, A_2 corresponding adjacency matrices

$G \subseteq \mathrm{Mat}_{n,n}(k)$ set of $n \times n$ permutation matrices,
W the space of linear functions on $G \subseteq \mathrm{Mat}_{n,n}(k)$
$V = \mathrm{Mat}_{n,n}(k)$, G acts on V by conjugation

Distinguishing Graphs using the category $\mathcal{C}_d(V)$

Γ_1, Γ_2 two graphs with n vertices
A_1, A_2 corresponding adjacency matrices

$G \subseteq \mathrm{Mat}_{n,n}(k)$ set of $n \times n$ permutation matrices,
W the space of linear functions on $G \subseteq \mathrm{Mat}_{n,n}(k)$
$V = \mathrm{Mat}_{n,n}(k)$, G acts on V by conjugation

Theorem

Assume that k has characteristic 0 or $> n$. If A_1, A_2 are isomorphic in $\mathcal{C}_d(V)$, then the $(d-1)$-dimensional Weisfeiler-Lehman algorithm cannot distinguish the graphs Γ_1, Γ_2.

Distinguishing Graphs using the category $\mathcal{C}_d(V)$

Γ_1, Γ_2 two graphs with n vertices
A_1, A_2 corresponding adjacency matrices

$G \subseteq \mathrm{Mat}_{n,n}(k)$ set of $n \times n$ permutation matrices,
W the space of linear functions on $G \subseteq \mathrm{Mat}_{n,n}(k)$
$V = \mathrm{Mat}_{n,n}(k)$, G acts on V by conjugation

Theorem

Assume that k has characteristic 0 or $> n$. If A_1, A_2 are isomorphic in $\mathcal{C}_d(V)$, then the $(d-1)$-dimensional Weisfeiler-Lehman algorithm cannot distinguish the graphs Γ_1, Γ_2.

For fixed d, isomorphisms in $\mathcal{C}_d(V)$ can be checked using a polynomial number of arithmetic operations in k. If $k = \mathbb{F}_p$ and $p = O(n)$ then isomorphism can be checked in polynomial time.

Distinguishing Graphs using the category $\mathcal{C}_d(V)$

Γ_1, Γ_2 two graphs with n vertices
A_1, A_2 corresponding adjacency matrices

$G \subseteq \mathrm{Mat}_{n,n}(k)$ set of $n \times n$ permutation matrices,
W the space of linear functions on $G \subseteq \mathrm{Mat}_{n,n}(k)$
$V = \mathrm{Mat}_{n,n}(k)$, G acts on V by conjugation

Theorem

Assume that k has characteristic 0 or $> n$. If A_1, A_2 are isomorphic in $\mathcal{C}_d(V)$, then the $(d-1)$-dimensional Weisfeiler-Lehman algorithm cannot distinguish the graphs Γ_1, Γ_2.

For fixed d, isomorphisms in $\mathcal{C}_d(V)$ can be checked using a polynomial number of arithmetic operations in k. If $k = \mathbb{F}_p$ and $p = O(n)$ then isomorphism can be checked in polynomial time.

So our algorithm is at least as powerful as the Weisfeiler-Lehman algorithm.

Distinguishing the CFI graphs in polynomial time

Suppose that Γ_1, Γ_2 is a pair of non-isomorphic graphs in the Cai-Fürer-Immerman family.

Distinguishing the CFI graphs in polynomial time

Suppose that Γ_1, Γ_2 is a pair of non-isomorphic graphs in the Cai-Fürer-Immerman family.

Theorem

If $k = \mathbb{F}_2$ then A_1, A_2 are *not* isomorphic in $\mathcal{C}_3(V)$.

Distinguishing the CFI graphs in polynomial time

Suppose that Γ_1, Γ_2 is a pair of non-isomorphic graphs in the Cai-Fürer-Immerman family.

Theorem

If $k = \mathbb{F}_2$ then A_1, A_2 are *not* isomorphic in $\mathcal{C}_3(V)$.

So using our algorithm distinguishes these graphs in polynomial time, but the Weisfeiler-Lehman algorithm cannot distinguish these graphs in polynomial time.

Why is our algorithm more powerful?

It is hard to say “the rank of the adjacency matrix (over the field $\mathbb{F}_p$) of Γ has rank r. One cannot express such a sentence in $\mathbf{C}_d$ for small d.

Why is our algorithm more powerful?

It is hard to say "the rank of the adjacency matrix (over the field $\mathbb{F}_p$) of Γ has rank r. One cannot express such a sentence in $\mathbf{C}_d$ for small d.

Our algorithm captures a "logic" that is more powerful. For $d = 3$ one can already express that the adjacency matrix has rank r.

Why is our algorithm more powerful?

It is hard to say "the rank of the adjacency matrix (over the field $\mathbb{F}_p$) of Γ has rank r. One cannot express such a sentence in $\mathbf{C}_d$ for small d.

Our algorithm captures a "logic" that is more powerful. For $d = 3$ one can already express that the adjacency matrix has rank r.

The CFI graphs can easily be distinguished, because their adjacency matrices have canonical submatrices with distinct ranks (when working over $\mathbb{F}_2$).

Questions

Can our algorithm distinguish the CFI graphs in polynomial time if we work over fields of characteristic $\neq 2$?

Questions

Can our algorithm distinguish the CFI graphs in polynomial time if we work over fields of characteristic $\neq 2$?

Can our algorithm distinguish graphs of bounded valence in polynomial time?

Questions

Can our algorithm distinguish the CFI graphs in polynomial time if we work over fields of characteristic $\neq 2$?

Can our algorithm distinguish graphs of bounded valence in polynomial time?

(Wishful thinking)

Can our algorithm distinguish all graphs in polynomial time?