GQS: Graph Query System

Thomas Fannes & Jan Ramon

[firstname.lastname]@cs.kuleuven.be
GQS

- Overview

- Target value algebras
  - Homomorphism
  - Isomorphism
  - Support measure & embedding significance

- Extras & current status
GQS

- Overview

- Target value algebras
  - Homomorphism
  - Isomorphism
  - Support measure & embedding significance

- Extras & current status
Many different flavors of graph databases:
- Single network vs multiple networks
- Standard graph vs hypergraph
- Query type:
  - Pattern
  - Embedding
  - Aggregated value
  - ... Embeddings under [iso|homo]-morphism
GQS overview

• Why a new graph query system?

• Use state-of-the-art graph algorithms
  – More efficient mining (special subclasses)
  – Results applicable for data mining

• GQS settings:
  – Single, large network database
  – Rooted, bounded treewidth pattern graphs
GQS overview

- Bounded treewidth decomposition
GQS overview

- Rooted bounded treewidth graph
GQS overview

- query operators:
  - List
  - Extend
  - Project
  - Join
  - Select

- Per embedding, calculate a target value:

  \[ T(e) = \prod_{v \in e} T(v) \]
GQS overview

Pattern

Network

\[ x_1 x_2^2 + x_1 x_2 x_3 + x_1 x_2 x_4 + \]
\[ x_1 x_3 x_2 + x_1 x_3^2 + x_1 x_3 x_4 + x_1 x_4 x_2 + x_1 x_4 x_3 + x_1 x_4^2 \]
GQS overview

- Target value (per embedding):
  
  Product of values associated with network vertices

- List: \( T( \ L(v) \ ) = T(v) \)
- Extend: \( T( \ E(e,v) \ ) = T(e) \cdot T(v) \)
- Join: \( T( \ J(e_1, e_2) \ ) = T(e_1) \cdot T( e_2 \setminus e_1) \)
- Project: \( T( \ P(e_1 \cup ... , e_1) \ ) = \Sigma T(e_1 \cup ...) \)
GQS

- Overview

- Target value algebras
  - Homomorphism
  - Isomorphism
  - Support measure & embedding significance

- Extras & current status
GQS: default

Task: List **homomorphic** root embeddings of a pattern

evaluation of (a projection of) a conjunctive query

- No target value (or count embeddings per root)
- Runtime:
  - Polynomial in network size
  - Polynomial in pattern size
  - Exponential in pattern treewidth
GQS

- Overview

- Target value algebras
  - Homomorphism
  - Isomorphism
  - Support measure & embedding significance

- Extras & current status
GQS: $GF_2$

Task: List **isomorphic** root embeddings of a pattern

$$x_1 x_2^2 + x_1 x_2 x_3 + x_1 x_2 x_4 +$$

$$x_1 x_3 x_2 + x_1 x_3^2 + x_1 x_3 x_4 + x_1 x_4 x_2 + x_1 x_4 x_3 + x_1 x_4^2$$
Task: List isomorphic root embeddings of a pattern
Task: List **isomorphic** root embeddings of a pattern

\[x_1 x_2^2 + x_1 x_2 x_3 + x_1 x_2 x_4 +
\]
\[x_1 x_3 x_2 + x_1 x_3^2 + x_1 x_3 x_4 + x_1 x_4 x_2 + x_1 x_4 x_3 + x_1 x_4^2\]

- Use $GF(2^l) \mathbb{Z}_2^k$ as target value algebra:
  - One embedding $\leftrightarrow$ one term
  - Squares or higher are evaluated to zero
  - Randomized approach:
    - $T(e) \neq 0 \rightarrow$ isomorphic embedding of pattern
    - $T(e) = 0 \rightarrow$ Pr[non-isomorphic embedding of pattern] > $(1-\varepsilon)$
Task: List **isomorphic** root embeddings of a pattern

- Target value in randomized $GF(2^l) \mathbb{Z}_2^k$ algebra
- Runtime:
  - Polynomial in network size
  - Mildly exponential in pattern size $O(2^{|V(P)|})$
  - Exponential in pattern treewidth

- (Kibriya & Ramon, DMKD 2013)
Overview

Target value algebras
- Homomorphism
- Isomorphism
- Support measure & embedding significance

Extras & current status
GQS: s-measure

- Vertices: objects
- Hyperedges: examples
- Hyperedge overlap on shared objects.
  - these examples are not independent

- How do we learn from these examples?

![Diagram of GQS: s-measure](attachment:diagram.png)
• Measuring effective sample size
  – Given a networked training set, can we get information out of it equivalent to \( n \) i.i.d. examples? What is \( n \)?
  – What is the weight for each embedding w.r.t. \( n \)?

• For a pattern \( P \), the s-measure gives a anti-monotonic support measure
  \( \rightarrow \) e.g. finding frequent patterns

• (Wang & Ramon, DMKD 2013)
GQS: s-measure

- Influence of each embedding is at most 1:
- \( \max s \)
  \[ s = w_1 + w_2 + w_3 + w_4 + w_5 + w_6 \]

Subject to
\[
\begin{align*}
  w_1 + w_2 &\leq 1, \quad w_1 + w_3 \leq 1, \\
  w_4 + w_5 &\leq 1, \quad w_5 + w_6 \leq 1, \\
  w_4 + w_6 &\leq 1 \\
  w_1, w_2, w_3, w_4, w_5, w_6 &\geq 0
\end{align*}
\]

\[ s = 3.5 \]
GQS: s-measure

- Linear program \(\rightarrow\) efficient
  - Support measure for each pattern
  - Statistical significance of each embedding \((w_1, w_2, w_3, w_4, w_5, w_6)\)

- GQS target value?
  - Writes out program!

\(s = 3.5\)
GQS

- Overview

- Target value algebras
  - Homomorphism
  - Isomorphism
  - Support measure & embedding significance

- Extras & current status
GQS: extras

- Different query plan optimizations:
  - Memory footprint reducing
  - Optimizations specific for target value algebra

- 2-step approach:

  C++ runtime polymorphic system

  C++ template-based query program
GQS: Current status

- Query system
- Query optimization
- GF₂ isomorphism integration
- GF₂ extended tests
- s-measure integration
- s-measure extended tests
Thanks!

Any questions?