Distance oracles in edge-labeled graphs

F. Bonchi * A. Gionis † F. Gullo * A. Ukkonen †

* Yahoo Labs
Barcelona, Spain

YAHOO! LABS

† HIIT
Aalto University, Finland

17th International Conference on Extending Database Technology (EDBT)
March 24-28, 2014
Athens, Greece

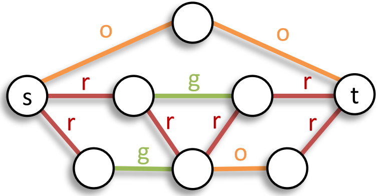
Motivations

- Fast approximation of *shortest-path (SP) distance* queries is an extremely well-studied problem arising in a plethora of today's applications
  - route planning, GIS systems, computer games, server selection, XML indexing, packet routing, web-search ranking, recommender systems

- *Edge-labeled graphs* have become common nowadays
  - *Social circles* in *social networks* (e.g., "circles" in Google+ or "lists" in Facebook or Twitter)
  - *RDF resources* (e.g., *Google Knowledge Graph*, *Yago*): each relationship between two entities is labeled with the type of the property (predicate)
  - *Co-authorship networks* (e.g., DBLP): a link between two authors is labeled with the topic(s) of the collaboration
  - In *protein-interaction networks* edge labels correspond to different type of interaction between proteins
  - *Multi-dimensional networks*, *metabolic networks*, *recommendation networks*, etc.

Our problem

- Approximation of SP distance queries + edge-labeled graphs
  $\Rightarrow$ *label-constrained point-to-point shortest-path distance* (LC-PPSPD) queries on edge-labeled graphs
  - Given two vertices s and t and a set of labels C, find the length of a shortest path between s and t, using only edges whose label belongs to C



$\langle s, t, \{r\} \rangle = 4$ $\quad$ $\langle s, t, \{r, g\} \rangle = 3$ $\quad$ $\langle s, t, \{r, g, o\} \rangle = 2$

LC-PPSPD queries: applications

- Real-time queries on RDF resources, such as Google Knowledge Graph or Facebook Graph Search
  - LC-PPSPD queries as primitive for complex machine-learned ranking function used for answering "*How related are entities A and B, contextualized to additional user information C?*"
- Edge-label prediction
  - LC-PPSPD queries as feature for machine-learning-based edge-label prediction systems: both offline model learning and online prediction need to rely on a set of example LC-PPSPD queries
- Network alignment in protein-interaction networks
  - LC-PPSPD queries as pruning condition to speed-up complex subgraph-isomorphism-based queries, such as finding all pathways that match an input pathway

Related Work

- Non-labeled graphs: the literature is huge!
- Edge-labeled graphs: some studies on *subset-constrained reachability* (a special case of LC-PPSPD queries)
  [Jin et al., SIGMOD'10; Xu et al., CIKM'11; Fan et al., ICDE'11]
- Work by Rice and Tsotras, "*Graph indexing of road networks for shortest path queries with label restrictions*", PVLDB'10
  - Road networks
  - Exact methods
- Work by Likhyani and Bedathur, "*Label Constrained Shortest Path Estimation*", CIKM'13
  - Concurrent submission

Contributions

- Answering LC-PPSPD queries is poly-time: we aim at faster approximate answers
- We design two indexes that trade-off between storage vs. accuracy/efficiency:
  - PowCov, faster and more accurate query processing
  - ChromLand, less storage space and indexing time

Problem definition

- Input: an edge-labeled graph $G = (V, E, L, \ell)$
  - V vertices, $E \subseteq V \times V$ edges, L labels, $\ell : E \to L$ edge labeling function

- Given $C \subseteq L$ and $u, v \in V$, a *C-constrained path* $p_C(u, v)$ is a path between u and v whose edges e have all $\ell(e) \in C$

- The *C-constrained distance* $d_C(u, v)$ is the length of a $p_C(u, v)$ ($d_C(u, v) = \infty$ if no $p_C(u, v)$ exists)

- An LC-PPSPD query is a triple $\langle s, t, C \rangle$, whose answer is the C-constrained distance $d_C(u, v)$

Challenges

- *Landmark* approach for traditional SP distance queries:
  1. For an input graph $G = (V, E)$, select k landmark vertices $X \subseteq V$
  2. Compute the exact distances $d(u, x)$ for all $u \in V$ and $x \in X$
  3. The distance $d(s, t)$ is approximated by triangle inequality:
     $$\max_{x \in X} |d(x, s) - d(x, t)| \ \leq \ d(s, t) \ \leq \ \min_{x \in X} (d(x, s) + d(x, t))$$
  4. $\mathcal{O}(kn)$ space, $\mathcal{O}(km)$ indexing time, $\mathcal{O}(k)$ query-processing time

- Naïve adaptation for LC-PPSPD queries:
  1. For an input edge-labeled graph $G = (V, E, L, \ell)$, select k landmark vertices $X \subseteq V$
  2. Compute the exact distances $d_C(u, x)$ for all $u \in V$, all $x \in X$, and all $C \subseteq L$
  3. The distance $d_C(s, t)$ is approximated by triangle inequality:
  4. $\mathcal{O}(k)$ query-processing time, but $\mathcal{O}(k2^{|L|}n)$ space and $\mathcal{O}(k2^{|L|}m)$ indexing time!

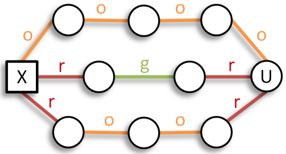
PowCov index

PowCov index: overview

Key observation

In real-world graphs, different constraint label sets yield the same distances between graph vertices.

- Given $S, T \subseteq L$, S *subsumes* T w.r.t. $x \in X$ and $u \in V$ iff $S \subseteq T$ and $d_S(x, u) = d_T(x, u)$
- $S \subseteq L$ is *shortest-path (SP) minimal* w.r.t. $x \in X$ and $u \in V$ iff is not subsumed by any other label set



- $\{$o$\}$ and $\{$r, g$\}$ are SP-minimal w.r.t. x and u, while $\{$r, o$\}$ is not
- The $\{$r, o$\}$-constrained distance can implicitly be derived from $\{$o$\}$ that subsumes $\{$r, o$\}$

PowCov index: overview

- Storing all SP-minimal label sets for a vertex-landmark pair (x, u) is sufficient for retrieving the exact distance between x and u, for each subset $C \subseteq L$:

Theorem

Given a landmark-vertex pair (x, u), let $\mathcal{SP}_{xu}$ be the set of $\langle S, d_S \rangle$ pairs containing all SP-minimal label sets S with respect to x and u along with the corresponding S-constrained shortest-path distance d_S. Then, for any label set $C \subseteq L$, the C-constrained distance $d_C(x, u)$ can be retrieved from $\mathcal{SP}_{xu}$ as

$$d_C(x,u) = \begin{cases} \infty, & \textit{if there is no } \langle S, d_S \rangle \in \mathcal{SP}_{xu} \textit{ s.t. } S \subseteq C \\ \min\{d_S \mid \langle S, d_S \rangle \in \mathcal{SP}_{xu}, S \subseteq C\}, & \textit{otherwise.} \end{cases}$$

PowCov index: structure and query processing

- The structure of PowCov corresponds to all $\mathcal{SP}_{xu}$, partitioned based on d_S, and organized in a prefix-tree
  - Index storage space: $\mathcal{O}(kHn)$ $(H \ll 2^{|L|}$ in practice)
- Answering a query $\langle s, t, C \rangle$: retrieve $d_C(x, s)$, $d_C(x, t)$, $\forall x \in X$, and approximate the answer by triangle inequality
  - Query-processing time: $\mathcal{O}(kH|L|)$

PowCov index: building the index

A brute-force algorithm ($\mathcal{O}(2^{|L|}k(m+n|L|))$ time):

Input: an edge-labeled graph $G=(V,E,L,\ell)$, a set of landmarks X

Output: for each pair (x,u), where $x \in X$ and $u \in V$, a set $\mathcal{SP}_{xu}$ of $\langle C, d\rangle$ pairs storing all SP-minimal label sets C with respect to x and u along with the corresponding C-constrained shortest path distance d

$\mathcal{SP}_{xu} \leftarrow \emptyset$, $\forall x \in X, u \in V$
for all $x \in X$ **do**
    $\mathbf{D} \leftarrow \emptyset$
    for all $C \subseteq L$ **do**
        $\mathbf{D}[C] \leftarrow$ ConstrainedSSSP(G, x, C)
    end for
    for all $C \subseteq L$, $u \in V$ s.t. $\mathbf{D}[C,u] < \infty$ **do**
        if C is SP-minimal w.r.t. x and u **then**
            $\mathcal{SP}_{xu} \leftarrow \mathcal{SP}_{xu} \cup \{\langle C, \mathbf{D}[C,u]\rangle\}$
        end if
    end for
end for

PowCov index: building the index

Pruning the search space:

- *Skipping unnecessary label sets*, i.e., recognize early the label sets C for which there exists no vertex u s.t. C is SP-minimal w.r.t. x and u.
- *Skipping unnecessary SP-minimality tests*, i.e., once a C-constrained SSSP with source x has been computed, identify a set of vertices for which C cannot be SP-minimal and skip the corresponding SP-minimality test.
- *Speeding-up SP-minimality tests*, i.e., for some vertices u, recognize if a label set C is SP-minimal or not w.r.t x and u more efficiently than $\mathcal{O}(|C|)$ time.

$\Rightarrow$ **the** TraversePowerset **algorithm**

ChromLand index

ChromLand index

Motivation: in the worst case, the construction time of PowCov is still exponential, it might be not affordable for large label sets

Key observation

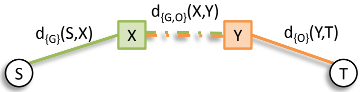
Assign each of the k landmarks $x \in X$ to a *single* label $c(x) \in L$

- *chromatic landmarks*
- *chromatic distances* $cd(x, u) = d_{\{c(x)\}}(x, u)$ (vertex-to-landmark) and $cd(x, y) = d_{\{c(x), c(y)\}}(x, y)$ (landmark-to-landmark)

$\Rightarrow$ **Structure of ChromLand** ($\mathcal{O}(km)$ time, $\mathcal{O}(kn)$ space):

- For each vertex $u \in V \setminus X$, keep $cd(x, u)$, $\forall x \in X$
- For each landmark $x \in X$, keep $cd(x, y)$, $\forall y \in X \setminus \{x\}$

ChromLand index: query processing

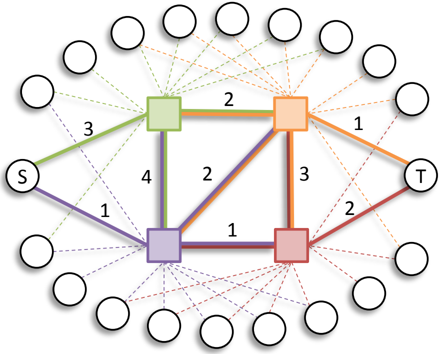


- ChromLand query-processing strategy for $\langle s, t, \{\text{g,o}\}\rangle$: $d_{\{\text{g,o}\}}(s, t)$ is approximated by a path passing trough two landmarks, x and y
- The SP distance from s to t is upper bounded by $d_{\{\text{g}\}}(s, x) + d_{\{\text{g,o}\}}(x, y) + d_{\{\text{o}\}}(y, t)$: this might improve using only x

Theorem

Let $G = (V, E, L, \ell)$ be an edge-labeled graph and $X \subseteq V$ a set of landmarks. Let $G_X = (V, X, E_X, c, w)$ be the auxiliary graph of G defined over G and X. Given a label set C and two vertices $u, v \in V$, let $G_X[u, v, C]$ denote the subgraph of G_X induced by the set of vertices $\{u, v\} \cup \{x \in X \mid c(x) \in C\}$. For any query $\langle s, t, C\rangle$ the shortest path distance $\delta_C(s, t)$ between s and t computed on $G_X[s, t, C]$ is the tightest upper bound to $d_C(s, t)$ that can be computed from the information stored by ChromLand index.

ChromLand index: query processing



ChromLand query-processing strategy for $\langle s, t, \{\mathrm{b, g, o, r}\}\rangle$:

- take the subgraph induced by s and t and all landmarks whose label is in $\{\mathrm{b, g, o, r}\}$
- approximate $d_{\{\mathrm{b,g,o,r}\}}(s, t)$ by the SP distance between s and t on that subgraph
- $\mathcal{O}(k^2)$ query-processing time

Selecting landmarks

Selecting landmarks for PowCov

Definition

Given a set of landmarks X and a query $Q = \langle s, t, C \rangle$, let $\tilde{d}_{\mathrm{PC}}(Q, X)$ denote the approximate answer to Q provided by the PowCov index using the landmarks X. A set of landmarks X is called PowCov-exact if and only if $\tilde{d}_{\mathrm{PC}}(Q, X) = d_C(s, t)$, for all queries $Q = \langle s, t, C \rangle$.

Problem (POWCOV-LANDMARK-SELECTION)

Given an edge-labeled graph $G = (V, E, L, \ell)$, find a minimum-sized set of landmarks $X \subseteq V$ such that X is PowCov-exact.

Theorem

A set of landmarks X is a solution for the POWCOV-LANDMARK-SELECTION *problem if and only if X is a minimum vertex cover of the input graph.*

$\Rightarrow$ We relax exactness and we ask for the k landmarks that *maximize the number of queries that can be answered exactly*

- the problem is **NP**-hard and we design a $\max\left\{1 - \frac{1}{e}, \frac{k}{n}\right\}$-greedy approximation algorithm

Selecting landmarks for ChromLand

Landmark selection for ChromLand is more complex than PowCov:

Theorem

Given an edge-labeled graph $G = (V, E, L, \ell)$, a set of landmarks $X \subseteq V$ allows the ChromLand index to provide exact answers only if for all pairs $u, v \in V$ and all label sets $C \subseteq L$, there exists a shortest path $p^*_C(u, v)$ such that $|X \cap \{i \mid (i, j) \in p^*_C(u, v)\}| \geq |\mathrm{labels}(p^*_C(u, v))|$.

$\Rightarrow$ We select landmarks so that *any vertex of the graph is close to at least one landmark for any given label*:

Problem (ChromLand-Landmark-Selection)

Given an edge-labeled graph $G = (V, E, L, \ell)$ and an integer k, find a set of k landmarks $X \subseteq V$ and a landmark-labeling function $c : X \to L$ so as to maximize the objective function

$$J(G, X, c) = \sum_{u \in V} \max_{x \in X} \mathrm{sim}_c(x, u).$$

- the problem is **NP**-hard and we design a k-Median-based heuristic

Experiments

Experiments: datasets

Characteristics of the selected datasets

dataset	# *vertices*	# *edges*	# *labels*	*diameter*	# *queries*
BioGrid	26 806	298 957	7	18	19 037
BioMine	943 510	5 727 448	7	16	20 799
String	1 490 098	8 886 639	6	19	18 149
DBLP	47 598	252 881	8	19	18 611
Youtube	15 088	19 923 067	5	6	23 499
synthetic	500 000	2 500 000	4–100	[5, 20]	$\sim$ [15K, 100K]

Experiments: index size

Average number of distances stored per landmark-vertex pair

real datasets

index	BioGrid ($\lvert L\rvert=7$)	BioMine ($\lvert L\rvert=7$)	String ($\lvert L\rvert=6$)	DBLP ($\lvert L\rvert=8$)	YouTube ($\lvert L\rvert=5$)
PowCov	5.79	3.88	2.01	8.63	4.72
Naïve	84.24	74.43	34.66	116.3	29.21
	93.1%	94.8%	94.2%	92.6%	83.8%

synthetic datasets (varying $\lvert L\rvert$)

index	4	5	6	7	8	9	10
PowCov	9.12	14.73	24.35	39.09	60.36	92.19	123.7
Naïve	13.39	27.69	56.59	115.1	233.3	470.68	950.7
	31.9%	46.8%	57%	66%	74.1%	80.4%	87%

Experiments: indexing time

Average time (secs) per single landmark

real datasets

index	BioGrid	BioMine	String	DBLP	YouTube
ChromLand	0.2	4.43	0.04	0.18	2.5
PowCov	5.8	156.2	0.59	14.6	20.2
BruteForce	11.3	269.8	1.09	38	29.4
	48.9%	42.1%	45.9%	61.7%	31.1%

synthetic datasets (varying $|L|$)

index	4	5	6	7	8	9	10	30	50	100
ChromLand	4.1	4.8	5.7	5.6	6	6.6	6.4	2.7	2.02	1.2
PowCov	20.1	41.6	90.8	192.4	398	833.1	1783	—	—	—
BruteForce	33.2	76.2	179.5	409.4	963	2124	5631	—	—	—
	39.5%	45.4%	49.4%	53%	58.7%	60.8%	68.3%			

Experiments: query processing

PowCov, BioMine

#landmarks	100	200	300	400	500
absolute error (avg)	1.07	0.91	0.81	0.78	0.58
relative error (avg)	0.31	0.27	0.25	0.24	0.18
exact answers (%)	33.2	41.0	46.8	48.3	62.3
false negatives (%)	0.004	0.003	0.003	0.003	0.003
speed-up factor	3 696	1 952	1 382	999	982

PowCov, YouTube

#landmarks	40	80	120	160	200
absolute error (avg)	0.46	0.38	0.34	0.32	0.3
relative error (avg)	0.28	0.24	0.22	0.21	0.2
exact answers (%)	56.6	63.2	67.1	69.4	70.1
false negatives (%)	0	0	0	0	0
speed-up factor	1 649	1 173	966	899	882

ChromLand, BioMine

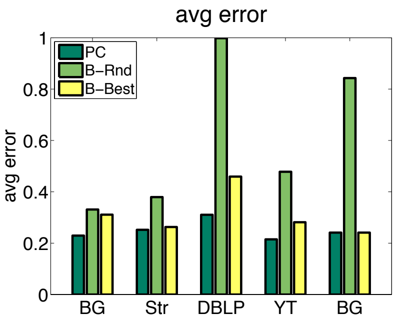
#landmarks	100	200	300	400	500
absolute error (avg)	2.34	1.94	1.84	1.8	1.76
relative error (avg)	0.63	0.52	0.5	0.49	0.48
exact answers (%)	9	12	13.9	15	16
false negatives (%)	0.002	0.001	0.001	0.001	0.001
speed-up factor	4 073	1 429	616	435	270

ChromLand, YouTube

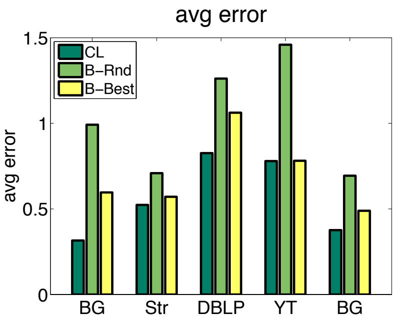
#landmarks	40	80	120	160	200
absolute error (avg)	0.86	0.63	0.55	0.49	0.46
relative error (avg)	0.49	0.37	0.33	0.3	0.28
exact answers (%)	26.2	41.9	47.6	53.1	56
false negatives (%)	0.04	0.04	0.04	0.04	0.04
speed-up factor	2 257	881	466	295	220

Experiments: landmark selection

PowCov



ChromLand



Conclusions

- We addressed the problem of fast online approximation of label-constrained shortest-path distance queries in edge-labeled graphs
- We devised two landmark-based indexes that trade-off between storage vs. accuracy/efficiency
- We developed rigorous landmark-selection strategies for each one of the proposed indexes
- Experiments on synthetic and real datasets revealed the high performance of our indexes

Thanks!