Repairs and Consistent Answers for Inconsistent Probabilistic Spatio-Temporal Databases

Francesco Parisi[1] John Grant[2]

[1]Department of Informatics, Modeling, Electronics and System Engineering,
University of Calabria, Italy,
email: fparisi@dimes.unical.it
[2]Department of Computer Science and UMIACS,
University of Maryland, College Park, USA,
email: grant@cs.umd.edu

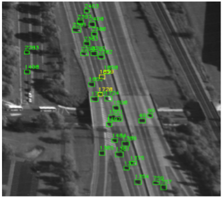
8th International Conference on Scalable Uncertainty Management
St. Anne's College, Oxford, UK,
September 15-17, 2014

Tracking moving objects (1/2)

- Tracking moving objects is fundamental in several application contexts (e.g. environment protection, product traceability, traffic monitoring, mobile tourist guides, analysis of animal behavior, etc.)

http://www.merl.com/publications/TR2008-010

http://www.edimax.com/au/



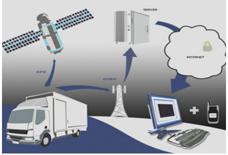
http://iris.usc.edu/people/medioni/current_research.html

http://www.i3b.org/content/wildlife-behavior

http://www.science20.com/news_articles/german_research_center_artificial_intelligence_smart_eye_tracking_glass

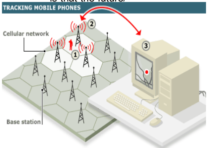
Tracking moving objects (2/2)

- Location estimation techniques have limited accuracy and precision
  - limitations of technologies used (e.g. GPS, Cellular networks, WiFi, Bluetooth, RFID, etc.)
  - limitations of the estimation methods (e.g., proximity to antennas, triangulation, signal strength sample map, dead reckoning, etc.)

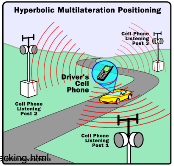


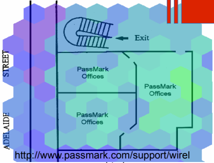
http://www.nitrobahn.com/conceptz/self-driving-cars-is-that-the-future/

http://www.ayantra.com/traffic-control-monitoring.html



http://www.gksoft.in/2014/07/mobile-phone-tracking.html





http://www.passmark.com/support/wireless_coverage_map.html

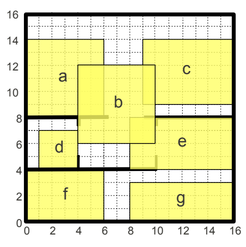
object inside a region at a time with (uncertain) probability

SPOT framework

- SPOT : a declarative framework for the representation and processing of probabilistic spatio-temporal data with uncertain probabilities [Parker, Subrahmanian, Grant. TKDE '07]
- A SPOT database is a set of atoms $loc(id, r, t)[\ell, u]$
- $loc(id, r, t)[\ell, u]$ means that "object *id* is/was/will be inside region r at time t with probability in the interval $[\ell, u]$".

Example

$$A_1 = loc(id_1, a, 3)[.5, .9]$$
$$A_2 = loc(id_1, b, 3)[.6, 1]$$
$$A_3 = loc(id_1, c, 3)[.7, .8]$$

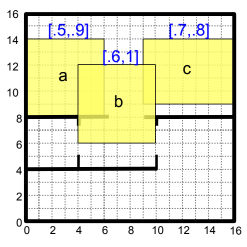


Inconsistency in probabilistic spatio-temporal data

- Recognizing process getting SPOT atoms not error-free
- Data coming from different sensors may be inconsistent (i.e., entailing that an object is in two places at the same time)

Example

$A_1 = loc(id_1, a, 3)[.5, .9]$
$A_2 = loc(id_1, b, 3)[.6, 1]$
$A_3 = loc(id_1, c, 3)[.7, .8]$



- Can we extract reliable information from inconsistent probabilistic spatio-temporal databases?

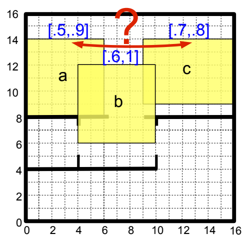
Inconsistency in probabilistic spatio-temporal data

- Recognizing process getting SPOT atoms not error-free
- Data coming from different sensors may be inconsistent (i.e., entailing that an object is in two places at the same time)

Example

$A_1 = loc(id_1, a, 3)[.5, .9]$
$A_2 = loc(id_1, b, 3)[.6, 1]$
$A_3 = loc(id_1, c, 3)[.7, .8]$

object id_1 cannot be at the same time with probability greater than .5 in region *a* and with probability greater than .7 in region *c* (disjoint from *a*)



- Can we extract reliable information from inconsistent probabilistic spatio-temporal databases?

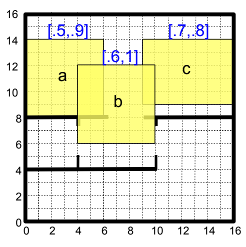
Repairs

- Two strategies for restoring consistency of a SPOT database D:
  - S-repairs are maximal consistent subsets of D
  - PU-repairs “minimally” update the probability bounds of the atoms in D

Example

$D = \{A_1, A_2, A_3\}$, where:

$A_1 = loc(id_1, a, 3)[.5, .9]$
$A_2 = loc(id_1, b, 3)[.6, 1]$
$A_3 = loc(id_1, c, 3)[.7, .8]$



- PTIME algorithms for computing S- and PU-repairs

Repairs

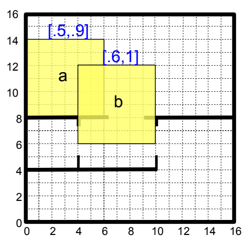
- Two strategies for restoring consistency of a SPOT database D:
  - S-repairs are maximal consistent subsets of D
  - PU-repairs "minimally" update the probability bounds of the atoms in D

Example

$D = \{A_1, A_2, A_3\}$, where:

$A_1 = loc(id_1, a, 3)[.5, .9]$
$A_2 = loc(id_1, b, 3)[.6, 1]$
$A_3 = loc(id_1, c, 3)[.7, .8]$

$D' = \{A_1, A_2\}$ is an S-repair for D



- PTIME algorithms for computing S- and PU-repairs

Repairs

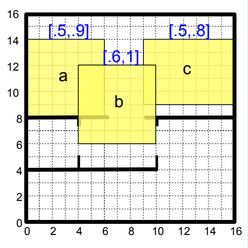
- Two strategies for restoring consistency of a SPOT database D:
  - S-repairs are maximal consistent subsets of D
  - PU-repairs "minimally" update the probability bounds of the atoms in D

Example

$D = \{A_1, A_2, A_3\}$, where:

$A_1 = loc(id_1, a, 3)[.5, .9]$
$A_2 = loc(id_1, b, 3)[.6, 1]$
$A_3 = loc(id_1, c, 3)[.7, .8]$

$D' = \{A_1, A_2\}$ is an S-repair for D
$D'' = \{A_1, A_2, A_3'\}$ with $A_3' = loc(id_1, c, 3)[.5, .8]$ is a PU-repair for D



- PTIME algorithms for computing S- and PU-repairs

Consistent Answers

- Selection query $(?id, r, ?t, [\ell, u])$: find all objects *id* and times *t* such that *id* is inside region *r* at time *t* with a probability in the interval $[\ell, u]$
- An *S*-consistent (resp. *PU*-consistent) answer to a selection query is an answer that can be obtained by every *S*-repair (resp. *PU*-repair)

Example

$$D = \{loc(id_1, a, 3)[.5, .9], \\ loc(id_1, b, 3)[.6, 1], \\ loc(id_1, c, 3)[.7, .8]\}$$

$$Q = (?id, r, ?t, [0.5, 1])$$

$\langle id_1, 3\rangle$ is a *PU*-consistent answer to *Q*, but not an *S*-consistent answer

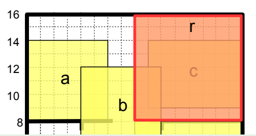
- Deciding whether $\langle id, t\rangle$ is a consistent answer is
  - *coNP*-complete for *S*-repair semantics
  - PTIME for *PU*-repair semantics
- Experimental evaluation of algorithms for computing *PU*-repairs and *PU*-consistent answers.

Consistent Answers

- Selection query $(?id, r, ?t, [\ell, u])$: find all objects *id* and times *t* such that *id* is inside region *r* at time *t* with a probability in the interval $[\ell, u]$
- An *S*-consistent (resp. *PU*-consistent) answer to a selection query is an answer that can be obtained by every *S*-repair (resp. *PU*-repair)

Example

$$D = \{loc(id_1, a, 3)[.5, .9], loc(id_1, b, 3)[.6, 1], loc(id_1, c, 3)[.7, .8]\}$$
$$Q = (?id, r, ?t, [0.5, 1])$$



$\langle id_1, 3\rangle$ is a *PU*-consistent answer to *Q*, but not an *S*-consistent answer

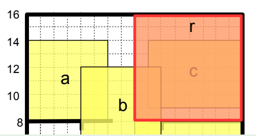
- Deciding whether $\langle id, t\rangle$ is a consistent answer is
  - *coNP*-complete for *S*-repair semantics
  - PTIME for *PU*-repair semantics
- Experimental evaluation of algorithms for computing *PU*-repairs and *PU*-consistent answers.

Consistent Answers

- Selection query $(?id, r, ?t, [\ell, u])$: find all objects *id* and times *t* such that *id* is inside region *r* at time *t* with a probability in the interval $[\ell, u]$
- An *S*-consistent (resp. *PU*-consistent) answer to a selection query is an answer that can be obtained by every *S*-repair (resp. *PU*-repair)

Example

$$D = \{loc(id_1, a, 3)[.5, .9], loc(id_1, b, 3)[.6, 1], loc(id_1, c, 3)[.7, .8]\}$$
$$Q = (?id, r, ?t, [0.5, 1])$$



$\langle id_1, 3\rangle$ is a *PU*-consistent answer to *Q*, but not an *S*-consistent answer

- Deciding whether $\langle id, t\rangle$ is a consistent answer is
  - *coNP*-complete for *S*-repair semantics
  - PTIME for *PU*-repair semantics
- Experimental evaluation of algorithms for computing *PU*-repairs and *PU*-consistent answers.

Outline

SPOT atom

- Notation:
  - *ID* is the set of objects identifiers
  - *Space* is a grid of $N \times N$ points
  - T is the set of time points
- An object can be in only one location at a time
- A single location may contain more than one object

Definition (SPOT atom)

A SPOT atom is of the form $loc(id, r, t)[\ell, u]$, where

- $id \in ID$ is an object id
- $r \subseteq Space$ is a region in the space
- $t \in T$ is a time point,
- $[\ell, u] \subseteq [0, 1]$ is a probability interval

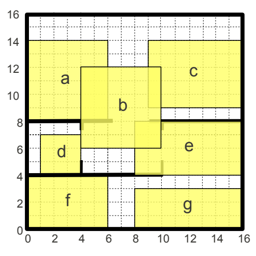
SPOT database

Definition (SPOT database)

A SPOT *database* is a finite set of SPOT atoms

Example

$$D = \{loc(id_1, d, 1)[.9, 1]$$
$$loc(id_1, b, 3)[.6, 1]$$
$$loc(id_1, c, 3)[.7, .8]$$
$$loc(id_2, b, 1)[.5, .9]$$
$$loc(id_2, e, 2)[.2, .5]$$
$$loc(id_3, e, 1)[.6, .9]\}$$



SPOT interpretation

Definition (Interpretation)

A SPOT *interpretation* is a function $I : ID \times Space \times T \to [0,1]$ such that for each $id \in ID$ and $t \in T$, $\sum_{p \in Space} I(id, p, t) = 1$

- For id and t, $I^{id,t}(p) = I(id, p, t)$ is a PDF over *Space*

Example

$I(id_1, (3,6), 1) = 0.4$
$I(id_1, (3,5), 1) = 0.3$
$I(id_1, (2,5), 1) = 0.2$
$I(id_1, (7,7), 1) = 0.1$
$I(id_2, (5,7), 1) = 0.7$
$I(id_2, (12,12), 1) = 0.3$
$I(id_3, (10,5), 1) = 0.8$
$I(id_3, (5,6), 1) = 0.2$

$I(id_1, (7,5), 2) = 0.5$
$I(id_1, (4,2), 2) = 0.5$
$I(id_2, (9,7), 2) = 0.3$
$I(id_2, (12,13), 2) = 0.7$
$I(id_3, (5,5), 2) = 0.5$
$I(id_3, (6,5), 2) = 0.5$

$I(id_1, (10,10), 3) = 0.7$
$I(id_1, (7,5), 3) = 0.3$
$I(id_2, (8,7), 3) = 0.9$
$I(id_2, (11,15), 3) = 0.1$
$I(id_3, (5,3), 3) = 0.6$
$I(id_3, (5,6), 3) = 0.4$

$I(id, p, t) = 0$ for all triplets (id, p, t) not mentioned above

SPOT interpretation

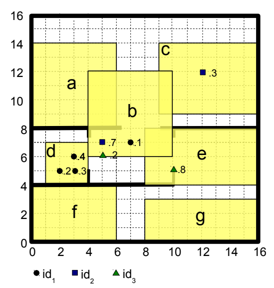
Definition (Interpretation)

A SPOT *interpretation* is a function $I : ID \times Space \times T \rightarrow [0, 1]$ such that for each $id \in ID$ and $t \in T$, $\sum_{p \in Space} I(id, p, t) = 1$

- For id and t, $I^{id,t}(p) = I(id, p, t)$ is a PDF over *Space*

Example (Time point 1)

$I(id_1, (3, 6), 1) = 0.4$
$I(id_1, (3, 5), 1) = 0.3$
$I(id_1, (2, 5), 1) = 0.2$
$I(id_1, (7, 7), 1) = 0.1$
$I(id_2, (5, 7), 1) = 0.7$
$I(id_2, (12, 12), 1) = 0.3$
$I(id_3, (10, 5), 1) = 0.8$
$I(id_3, (5, 6), 1) = 0.2$
$I(id, p, 1) = 0$ for all triplets $(id, p, 1)$ not mentioned above



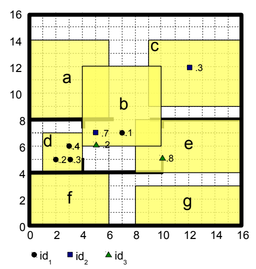
SPOT model

Definition (Satisfaction)

A SPOT interpretation I *satisfies* SPOT atom $loc(id, r, t)[\ell, u]$ (denoted as $I \models loc(id, r, t)[\ell, u]$) iff $\sum_{p \in r} I(id, p, t) \in [\ell, u]$

Example (Time point 1)

$I(id_1, (3, 6), 1) = 0.4$
$I(id_1, (3, 5), 1) = 0.3$
$I(id_1, (2, 5), 1) = 0.2$
$I(id_1, (7, 7), 1) = 0.1$
$I(id_2, (5, 7), 1) = 0.7$
$I(id_2, (12, 12), 1) = 0.3$
$I(id_3, (10, 5), 1) = 0.8$
$I(id_3, (5, 6), 1) = 0.2$



$D = \{loc(id_1, d, 1)[.9, 1]$
$loc(id_1, b, 3)[.6, 1]$
$loc(id_1, c, 3)[.7, .8]$
$loc(id_2, b, 1)[.5, .9]$
$loc(id_2, e, 2)[.2, .5]$
$loc(id_3, e, 1)[.6, .9]\}$

Definition (SPOT model)

An interpretation I is a *model* for a database D iff I satisfies every atom in D

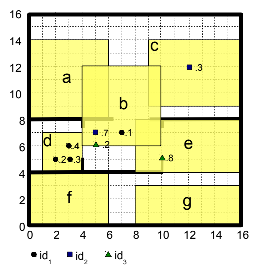
SPOT model

Definition (Satisfaction)

A SPOT interpretation I *satisfies* SPOT atom $loc(id, r, t)[\ell, u]$ (denoted as $I \models loc(id, r, t)[\ell, u]$) iff $\sum_{p \in r} I(id, p, t) \in [\ell, u]$

Example (Time point 1)

$I(id_1, (3,6), 1) = 0.4$
$I(id_1, (3,5), 1) = 0.3$
$I(id_1, (2,5), 1) = 0.2$
$I(id_1, (7,7), 1) = 0.1$
$I(id_2, (5,7), 1) = 0.7$
$I(id_2, (12,12), 1) = 0.3$
$I(id_3, (10,5), 1) = 0.8$
$I(id_3, (5,6), 1) = 0.2$



$D = \{loc(id_1, d, 1)[.9, 1]$
$loc(id_1, b, 3)[.6, 1]$
$loc(id_1, c, 3)[.7, .8]$
$loc(id_2, b, 1)[.5, .9]$
$loc(id_2, e, 2)[.2, .5]$
$loc(id_3, e, 1)[.6, .9]\}$

Definition (SPOT model)

An interpretation I is a *model* for a database D iff I satisfies every atom in D

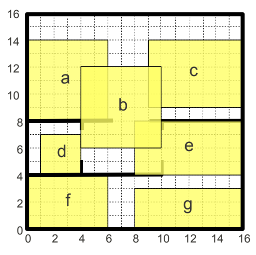
Consistency

Definition (Consistency)

A SPOT database is *consistent* iff it there is a model for it

Example

Interpretation I is a model



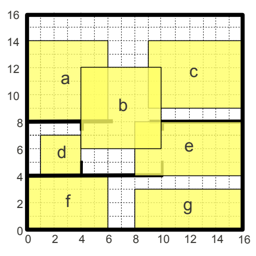
$$D = \{loc(id_1, d, 1)[.9, 1]$$
$$loc(id_1, b, 3)[.6, 1]$$
$$loc(id_1, c, 3)[.7, .8]$$
$$loc(id_2, b, 1)[.5, .9]$$
$$loc(id_2, e, 2)[.2, .5]$$
$$loc(id_3, e, 1)[.6, .9]\}$$

- It can be checked in PTIME [Parker, Subrahmanian, Grant. TKDE '07]

Example of Inconsistent Database

Example

$at_1 = loc(id_1, d, 1)[.9, 1]$
$at_2 = loc(id_1, a, 3)[.5, .9]$
$at_3 = loc(id_1, b, 3)[.6, 1]$
$at_4 = loc(id_1, c, 3)[.7, .8]$
$at_5 = loc(id_2, b, 1)[.5, .9]$
$at_6 = loc(id_2, e, 2)[.3, .5]$
$at_7 = loc(id_2, f, 2)[.5, .7]$
$at_8 = loc(id_2, g, 2)[.9, 1]$
$at_9 = loc(id_3, c, 1)[.5, .8]$
$at_{10} = loc(id_3, e, 1)[.6, .9]$



There is an inconsistency for object id_1 at time 3, for object id_2 at time 2, and for object id_3 at time 1

Outline

S-repairs

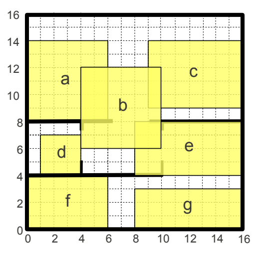
- Minimally modify the original database in order to restore consistency

Definition (S-repair)

An S-repair for a SPOT database D is a maximal consistent subset of D

Example

$at_1 = loc(id_1, d, 1)[.9, 1]$
$at_2 = loc(id_1, a, 3)[.5, .9]$
$at_3 = loc(id_1, b, 3)[.6, 1]$
$at_4 = loc(id_1, c, 3)[.7, .8]$
$at_5 = loc(id_2, b, 1)[.5, .9]$
$at_6 = loc(id_2, e, 2)[.3, .5]$
$at_7 = loc(id_2, f, 2)[.5, .7]$
$at_8 = loc(id_2, g, 2)[.9, 1]$
$at_9 = loc(id_3, c, 1)[.5, .8]$
$at_{10} = loc(id_3, e, 1)[.6, .9]$



Each S-repairs consists of
-at_1
-either at_2 and at_3 or at_3 and at_4
-at_5
-either at_6 and at_7 or at_8
-either at_9 or at_{10}

e.g. $\{at_1, at_2, at_3, at_5, at_8, at_9\}$

PU-repairs

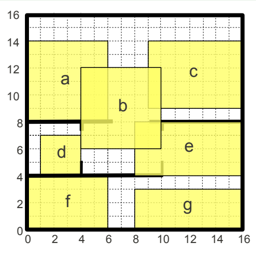
- A *probability-interval updated atom* for $a = loc(id, r, t)[\ell, u]$ is $a' = loc(id, r, t)[\ell', u']$ where $[\ell', u'] \supseteq [\ell, u]$

Definition (*PU*-repair)

A *PU*-repair for a SPOT database D is a consistent SPOT database D' consisting of a probability-interval update atom a' for each $a \in D$ and s.t. $\sum_{a \in D}(\ell - \ell') + (u' - u)$ is minimum.

Example

$at_1 = loc(id_1, d, 1)[.9, 1]$
$at_2 = loc(id_1, a, 3)[.5, .9]$
$at_3 = loc(id_1, b, 3)[.6, 1]$
$at_4 = loc(id_1, c, 3)[.7, .8]$
$at_5 = loc(id_2, b, 1)[.5, .9]$
$at_6 = loc(id_2, e, 2)[.3, .5]$
$at_7 = loc(id_2, f, 2)[.5, .7]$
$at_8 = loc(id_2, g, 2)[.9, 1]$
$at_9 = loc(id_3, c, 1)[.5, .8]$
$at_{10} = loc(id_3, e, 1)[.6, .9]$



$at'_1 = at_1$
$at'_2 = loc(id_1, a, 3)[.3, .9]$
$at'_3 = at_3$
$at'_4 = at_4$
$at'_5 = at_5$
$at'_6 = loc(id_2, e, 2)[.1, .5]$
$at'_7 = loc(id_2, f, 2)[.1, .7]$
$at'_8 = loc(id_2, g, 2)[.8, 1]$
$at'_9 = loc(id_3, c, 1)[.45, .8]$
$at'_{10} = loc(id_3, e, 1)[.55, .9]$

Properties of *S*- and *PU*-repairs

- There are exponentially many *S*-repairs for a SPOT database
- There are infinitely many *PU*-repairs for a SPOT database
- An *S*-repair and a *PU*-repair for a SPOT database always exist
- A repair for a SPOT database can be obtained by looking at one $\langle id, t\rangle$ pair at a time

Given a SPOT database D, $D_{id,t} = \{loc(id', r', t')[\ell', u'] \in D \mid id' = id \wedge t' = t\}$ be the set of atoms referring to $\langle id, t\rangle$

Proposition (*Repair modularity*)

A SPOT *database D' is an S-repair (resp. PU-repair) for D, iff $D' = \bigcup_{id \in ID, t \in T} D'_{id,t}$, where $D'_{id,t}$ is an S-repair (resp. PU-repair) for $D_{id,t}$.*

maximal Subset semantics

Theorem (Complexity of checking S-repairs)

Let D, D' be SPOT *DBs. Deciding whether D' is an S-repair for D is in PTIME.*

- Proof Hint: $\forall a \in D \setminus D'$ check that $D' \cup \{a\}$ is not consistent.

Corollary (Complexity of computing S-repairs)

An S-repair for SPOT *database can be computed in PTIME.*

- An S-repair for D can be computed as the union of S-repairs for $D_{id,t}$

1) $D'_{id,t} = \emptyset$
2) scan $D_{id,t}$ according to any total ordering, and for each $a \in D_{id,t}$
2) if $D'_{id,t} \cup \{a\}$ is consistent, then $D'_{id,t} = D'_{id,t} \cup \{a\}$

maximal Subset semantics

Theorem (Complexity of checking S-repairs)

Let D, D' be SPOT *DBs. Deciding whether D' is an S-repair for D is in PTIME.*

- Proof Hint: $\forall a \in D \setminus D'$ check that $D' \cup \{a\}$ is not consistent.

Corollary (Complexity of computing S-repairs)

An S-repair for SPOT *database can be computed in PTIME.*

- An S-repair for D can be computed as the union of S-repairs for $D_{id,t}$

1) $D'_{id,t} = \emptyset$
2) scan $D_{id,t}$ according to any total ordering, and for each $a \in D_{id,t}$
2) if $D'_{id,t} \cup \{a\}$ is consistent, then $D'_{id,t} = D'_{id,t} \cup \{a\}$

Probability-interval Update semantics

- Linear programming problem $PULP(D, id, t)$ whose optimal solutions encode PU-repairs for $D_{id,t}$

Definition ($PULP(D, id, t)$)

minimize $\sum_{a_i \in D_{id,t}} low_i + up_i$ ***subject to:***

$$\begin{cases} 1) \quad \forall a_i = loc(id, r_i, t)[\ell_i, u_i] \in D_{id,t} \\ \qquad \ell_i - low_i \leq \sum_{p \in r_i} v_p \leq u_i + up_i \\ \qquad 0 \leq low_i \leq \ell_i \\ \qquad 0 \leq up_i \leq 1 - u_i \\ 2) \quad \sum_{p \in Space} v_p = 1 \\ 3) \quad \forall p \in Space \quad v_p \geq 0 \end{cases}$$

- v_p represents the probability that id is at point $p \in Space$ at time t
- low_i and up_i represent the probability-interval update of atom a_i

Probability-interval Update semantics

- Let σ be a solution of $PULP(D, id, t)$.
  Let $D_{id,t}(\sigma)$ be the SPOT database obtained from $D_{id,t}$ by replacing each atom $a_i = loc(id, r_i, t)[\ell_i, u_i] \in D_{id,t}$ with the probability-interval updated atom $a'_i = loc(id, r_i, t)[\ell_i - \sigma[low_i],\ u_i + \sigma[up_i]]$

Theorem (Relationship between PULP solutions and PU-repairs)

For each optimal solution σ of $PULP(D, id, t)$, $D_{id,t}(\sigma)$ is a PU-repair for $D_{id,t}$. Moreover, every optimal solution σ for $PULP(D, id, t)$ one-to-one corresponds to a model for PU-repair $D_{id,t}(\sigma)$ for $D_{id,t}$, and vice versa.

Corollary (Complexity of checking PU-repairs)

Deciding whether D' is a PU-repair for SPOT DB D is in PTIME.

Corollary (Complexity of computing PU-repairs)

A PU-repair for SPOT database D can be computed in PTIME.

Probability-interval Update semantics

- Let σ be a solution of $PULP(D, id, t)$.
  Let $D_{id,t}(\sigma)$ be the SPOT database obtained from $D_{id,t}$ by replacing each atom $a_i = loc(id, r_i, t)[\ell_i, u_i] \in D_{id,t}$ with the probability-interval updated atom $a_i' = loc(id, r_i, t)[\ell_i - \sigma[low_i],\ u_i + \sigma[up_i]]$

Theorem (Relationship between PULP solutions and PU-repairs)

For each optimal solution σ of $PULP(D, id, t)$, $D_{id,t}(\sigma)$ is a PU-repair for $D_{id,t}$. Moreover, every optimal solution σ for $PULP(D, id, t)$ one-to-one corresponds to a model for PU-repair $D_{id,t}(\sigma)$ for $D_{id,t}$, and vice versa.

Corollary (Complexity of checking PU-repairs)

Deciding whether D' is a PU-repair for SPOT *DB D is in PTIME.*

Corollary (Complexity of computing PU-repairs)

A PU-repair for SPOT *database D can be computed in PTIME.*

Outline

Selection queries

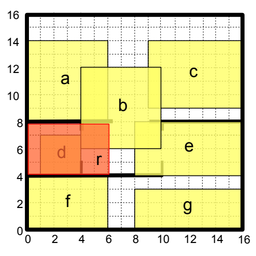
- A selection query asks to find all objects *id* and times *t* such that *id* is inside a given region *r* at time *t* with a probability in a given interval $[\ell, u]$

Definition (Selection Query Answers)

$\langle id, t\rangle$ is an answer to selection query $(?id, r, ?t, [\ell, u])$ w.r.t. SPOT database D iff for every model M for D, $M \models loc(id, r, t)[\ell, u]$.

Example

$$D = \{loc(id_1, d, 1)[.9, 1]$$
$$loc(id_1, b, 3)[.6, 1]$$
$$loc(id_1, c, 3)[.7, .8]$$
$$loc(id_2, b, 1)[.5, .9]$$
$$loc(id_2, e, 2)[.2, .5]$$
$$loc(id_3, e, 1)[.6, .9]\}$$



$$Q = (?id, r, ?t, [.75, 1])$$

Answer: $\langle id_1, 1\rangle$

Who and when was in *r* with probability greater than .75?

S- and PU-consistent queries

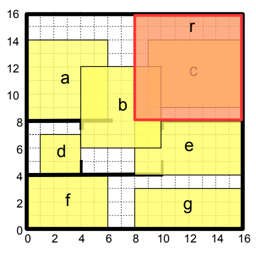
- X stands for either S or PU

Definition (X-Consistent Selection Query Answers)

Given a SPOT database D and a selection query $Q = (?id, r, ?t, [\ell, u])$, $\langle id, t\rangle$ is an X-consistent answer to Q w.r.t. D iff for each X-repair D' for D, $\langle id, t\rangle$ is an answer to Q w.r.t. D'.

Example

$$at_1 = loc(id_1, d, 1)[.9, 1]$$
$$at_2 = loc(id_1, a, 3)[.5, .9]$$
$$at_3 = loc(id_1, b, 3)[.6, 1]$$
$$at_4 = loc(id_1, c, 3)[.7, .8]$$
$$at_5 = loc(id_2, b, 1)[.5, .9]$$
$$at_6 = loc(id_2, e, 2)[.3, .5]$$
$$at_7 = loc(id_2, f, 2)[.5, .7]$$
$$at_8 = loc(id_2, g, 2)[.9, 1]$$
$$at_9 = loc(id_3, c, 1)[.5, .8]$$
$$at_{10} = loc(id_3, e, 1)[.6, .9]$$



$Q = (?id, r, ?t, [0.5, 1])$

$\langle id_1, 3\rangle$ is a PU-consistent answer to Q

The set of S-consistent answers is empty

maximal Subset semantics

Theorem (Complexity S-CQA)

Deciding whether $\langle id, t\rangle$ is an S-consistent answer to selection query Q w.r.t. SPOT *database D is coNP-complete.*

- Membership: Use that S-repair checking, and checking whether $\langle id, t\rangle$ is not an answer to Q, is in PTIME
- Hardness: Reduction from SUBSET SUM

Example

Set $S = \{2, 4, 5\}$
Constant $C = 7$

$Space = \{p_1, p_2, p_3, p_4\}$
$ID = \{id\}$
$T = 0$

$Q = (?id, \{p_4\}, ?t, [\frac{1}{7}, 1])$

$$\begin{aligned} D = \{at_1 &= loc(id, \{p_1\}, 0)[0, 0] \\ at_1' &= loc(id, \{p_1\}, 0)[\tfrac{2}{10}, \tfrac{2}{7}] \\ at_2 &= loc(id, \{p_2\}, 0)[0, 0] \\ at_2' &= loc(id, \{p_2\}, 0)[\tfrac{4}{10}, \tfrac{4}{7}] \\ at_3 &= loc(id, \{p_3\}, 0)[0, 0] \\ at_3' &= loc(id, \{p_3\}, 0)[\tfrac{5}{10}, \tfrac{5}{7}]\} \end{aligned}$$

- $\langle id, 0\rangle$ is not an S-CQA to Q w.r.t. D iff there is $S' \subseteq S$ s.t. $\sum_{s_i \in S'} s_i = C$
- $R = \{at_1', at_2, at_3'\}$ is a repair for D s.t. $\langle id, 0\rangle$ is not an answer to Q w.r.t. R
- $R = \{at_1', at_2, at_3'\}$ corresponds to subset $\{2, 5\}$ s.t. $2 + 5 = 7$

Probability-interval Update semantics

- PTIME algorithm PU-CQAs

1) Compute the minimum cost o^* of PU-repairs for $D_{id,t}$, by $PULP(D, id, t)$
2) Use o^* to construct and solve two additional LP problems (defined below)
3) Decide whether $\langle id, t \rangle$ is a PU-consistent answer by using the optimal values of these LP problems

Definition ($PU\text{-}CQA^{\ell}(D, Q, id, t)$ (resp. $PU\text{-}CQA^{u}(D, Q, id, t)$))

Let o^* be the optimal value of $PULP(D, id, t)$, and $Q = (?id, r, ?t, [\ell, u])$.
$PU\text{-}CQA^{\ell}(D, Q, id, t)$ (resp. $PU\text{-}CQA^{u}(D, Q, id, t)$) is as follows:

minimize (resp., ***maximize***) $\sum_{p \in r} v_p$ ***subject to:***

$$\begin{cases} 1) \quad \forall a_i = loc(id, r_i, t)[\ell_i, u_i] \in D_{id,t} \\ \qquad \ell_i - low_i \leq \sum_{p \in r_i} v_p \leq u_i + up_i \\ \qquad 0 \leq low_i \leq \ell_i \\ \qquad 0 \leq up_i \leq 1 - u_i \\ 2) \quad \sum_{p \in Space} v_p = 1 \\ 3) \quad \forall p \in Space \quad v_p \geq 0 \\ 4) \sum_{a_i \in D_{id,t}} low_i + up_i = o^* \end{cases}$$

(in)equalities of PULP *plus* equality 4)

Probability-interval Update semantics

Lemma (Relationship between solutions of *PU-CQA*$^{\ell/u}$ and PU-repairs)

Every optimal solution σ for PU-CQA$^{\ell}(D, Q, id, t)$ (resp., PU-CQA$^{u}(D, Q, id, t)$) one-to-one corresponds to a model for PU-repair $D_{id,t}(\sigma)$ such that the probability that id is in the query region r at time t is minimum (resp., maximum), and vice versa.

Theorem (Exploiting *PU-CQA*$^{\ell}$/*PU-CQA*^{u} to compute PU-CQAa)

Let $D_{id,t}$ be a SPOT *database, and $Q = (?id, r, ?t, [\ell, u])$.*
Let ℓ^* and u^* be the optimal values returned by PU-CQA$^{\ell}(D, Q, id, t)$ and PU-CQA$^{u}(D, Q, id, t)$, respectively.
$\langle id, t \rangle$ is a PU-CQA to Q w.r.t. $D_{id,t}$ iff $\ell \leq \ell^*$ and $u^* \leq u$

- $\langle id, t \rangle$ is a *PU*-CQA to *Q* w.r.t. *D* iff $\langle id, t \rangle$ is a *PU*-CQA *Q* w.r.t. $D_{id,t}$

Theorem (Complexity of PU-CQA)

Deciding whether $\langle id, t \rangle$ is a PU-CQA to Q w.r.t. SPOT *DB D is in PTIME*

Probability-interval Update semantics

Lemma (Relationship between solutions of *PU-CQA*$^{\ell/u}$ and PU-repairs)

Every optimal solution σ for PU-CQA$^{\ell}(D, Q, id, t)$ (resp., PU-CQA$^{u}(D, Q, id, t)$) one-to-one corresponds to a model for PU-repair $D_{id,t}(\sigma)$ such that the probability that id is in the query region r at time t is minimum (resp., maximum), and vice versa.

Theorem (Exploiting *PU-CQA*$^{\ell}$/*PU-CQA*^{u} to compute PU-CQAa)

Let $D_{id,t}$ be a SPOT *database, and $Q = (?id, r, ?t, [\ell, u])$.*
Let ℓ^* and u^* be the optimal values returned by PU-CQA$^{\ell}(D, Q, id, t)$ and PU-CQA$^{u}(D, Q, id, t)$, respectively.
$\langle id, t\rangle$ is a PU-CQA to Q w.r.t. $D_{id,t}$ iff $\ell \leq \ell^*$ and $u^* \leq u$

- $\langle id, t\rangle$ is a *PU*-CQA to *Q* w.r.t. *D* iff $\langle id, t\rangle$ is a *PU*-CQA *Q* w.r.t. $D_{id,t}$

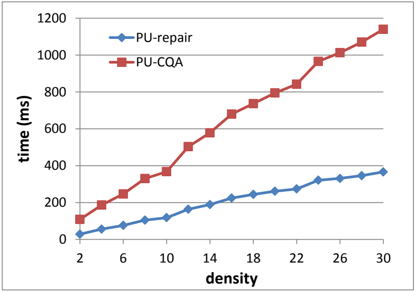
Theorem (Complexity of PU-CQA)

Deciding whether $\langle id, t\rangle$ is a PU-CQA to Q w.r.t. SPOT *DB D is in PTIME*

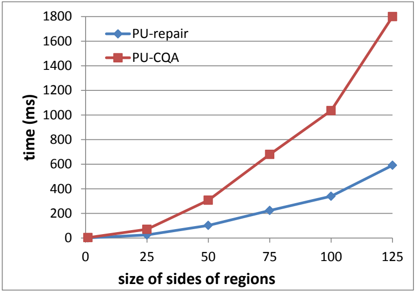
Outline

Experimental evaluation for PU semantics

- the *density* is the average cardinality of $D_{id,t}$ ($id \in ID$, $t \in T$) (i.e., the average number of times that an object was detected at a time point)
- ω is the average size of one side of the atom's rectangles (i.e., ω^2 is the average number of points in the detection's regions)



Repair and CQA time vs. the density ($\omega = 75$, $|Space| = 1000 \times 1000$).



Repair and CQA time vs. ω ($d = 16$, $|Space| = 1000 \times 1000$).

Outline

Other types of consistent answers

- We interpreted selection queries under a cautious semantics: $\langle id, t\rangle$ is an answer to query $Q = (?id, r, ?t, [\ell, u])$ w.r.t. D iff *for each* model M for D, $M \models loc(id, r, t)[\ell, u]$
- Optimistic semantics: just ensure that *there exists* a model M for D s.t. $M \models loc(id, r, t)[\ell, u]$.
- Let us denote the type of consistent answers introduced so far as *S-consistent universal cautious answers*.
- Given a SPOT database D and a selection query Q, we say that $\langle id, t\rangle$ is an *X-consistent*
  - *existential cautious answer* to Q w.r.t. D iff there exists $D' \in Rep_X(D)$, such that $\langle id, t\rangle$ is a cautious answer to Q w.r.t. D'.
  - *universal optimistic answer* to Q w.r.t. D iff for each $D' \in Rep_X(D)$, $\langle id, t\rangle$ is an optimistic answer to Q w.r.t. D'.
  - *existential optimistic answer* to Q w.r.t. D iff there exists $D' \in Rep_X(D)$, such that $\langle id, t\rangle$ is an optimistic answer to Q w.r.t. D'.

Other types of consistent answers

- We interpreted selection queries under a cautious semantics: $\langle id, t\rangle$ is an answer to query $Q = (?id, r, ?t, [\ell, u])$ w.r.t. D iff *for each* model M for D, $M \models loc(id, r, t)[\ell, u]$
- Optimistic semantics: just ensure that *there exists* a model M for D s.t. $M \models loc(id, r, t)[\ell, u]$.
- Let us denote the type of consistent answers introduced so far as *S-consistent universal cautious answers*.
- Given a SPOT database D and a selection query Q, we say that $\langle id, t\rangle$ is an X*-consistent*
  - *existential cautious answer* to Q w.r.t. D iff there exists $D' \in Rep_X(D)$, such that $\langle id, t\rangle$ is a cautious answer to Q w.r.t. D'.
  - *universal optimistic answer* to Q w.r.t. D iff for each $D' \in Rep_X(D)$, $\langle id, t\rangle$ is an optimistic answer to Q w.r.t. D'.
  - *existential optimistic answer* to Q w.r.t. D iff there exists $D' \in Rep_X(D)$, such that $\langle id, t\rangle$ is an optimistic answer to Q w.r.t. D'.

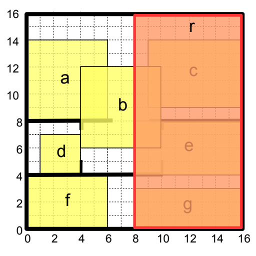
Example

Example

$$
\begin{aligned}
at_1 &= loc(id_1, d, 1)[.9, 1] \\
at_2 &= loc(id_1, a, 3)[.5, .9] \\
at_3 &= loc(id_1, b, 3)[.6, 1] \\
at_4 &= loc(id_1, c, 3)[.7, .8] \\
at_5 &= loc(id_2, b, 1)[.5, .9] \\
at_6 &= loc(id_2, e, 2)[.3, .5] \\
at_7 &= loc(id_2, f, 2)[.5, .7]
\end{aligned}
$$

$$
\begin{aligned}
at_7 &= loc(id_2, f, 2)[.5, .7] \\
at_8 &= loc(id_2, g, 2)[.9, 1] \\
at_9 &= loc(id_3, c, 1)[.5, .8] \\
at_{10} &= loc(id_3, e, 1)[.6, .9]
\end{aligned}
$$

$$Q = (?id, r, ?t, [0.6, 1])$$



Answers:

S-consistent universal cautious: $\emptyset$

S-consistent existential cautious: $\{(id_3, 1), (id_2, 2), (id_1, 3)\}$

S-consistent universal optimistic: $\{(id_3, 1), (id_2, 1)\}$

S-consistent existential optimistic: $\{(id_3, 1), (id_2, 1), (id_2, 2), (id_1, 3)\}$

PU-consistent universal cautious: $\{(id_3, 1)\}$

PU-consistent existential cautious: $\{(id_3, 1)\}$

PU-consistent universal optimistic: $\{(id_3, 1), (id_2, 2), (id_1, 3), (id_2, 1)\}$

PU-consistent existential optimistic: $\{(id_3, 1), (id_2, 1), (id_2, 2), (id_1, 3)\}$

Outline

Conclusions and future work

- All the previous works on probabilistic spatio-temporal DBs assume that the database is consistent
- We introduced database repairs and consistent selection query answers for SPOT DBs
- We shown that some cases can be solved in PTIME

	S semantics	*PU* semantics
Repair Checking	PTIME	PTIME
Consistent Answer	*coNP*-complete	PTIME

- We experimentally shown the feasibility of our approach
- Interesting directions for future work are:
  - Investigation of the complexity of different types of repairs (id/region/time update)
  - Complexity of checking different types of query answers
  - Repairs and consistent answers in the presence of the spatio-temporal integrity constraints (that we'll present tomorrow morning)

Conclusions and future work

- All the previous works on probabilistic spatio-temporal DBs assume that the database is consistent
- We introduced database repairs and consistent selection query answers for SPOT DBs
- We shown that some cases can be solved in PTIME

	S semantics	*PU* semantics
Repair Checking	PTIME	PTIME
Consistent Answer	*coNP*-complete	PTIME

- We experimentally shown the feasibility of our approach
- Interesting directions for future work are:
  - Investigation of the complexity of different types of repairs (id/region/time update)
  - Complexity of checking different types of query answers
  - Repairs and consistent answers in the presence of the spatio-temporal integrity constraints (that we'll present tomorrow morning)

Thank you!

... any question?

Selected References

Parker, A., Subrahmanian, V.S., Grant, J.
A logical formulation of probabilistic spatial databases.
IEEE TKDE, pp. 1541–1556, 2007.