# Aggregate Count Queries in Probabilistic Spatio-Temporal Databases

John Grant [1] Cristian Molinaro [2] Francesco Parisi [2]

[1]Towson University and University of Maryland at College Park, USA

[2]DIMES, Università della Calabria, Italy

# SPOT databases

**SPOT (Spatial PrObabilistic Temporal) databases**
[Parker, Subrahmanian, Grant. TKDE'07]

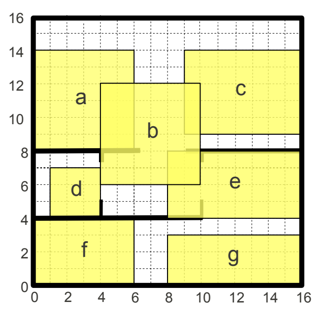
- Declarative framework for the representation and processing of probabilistic spatio-temporal databases with uncertain probabilities.

# SPOT databases

**SPOT (Spatial PrObabilistic Temporal) databases**
[Parker, Subrahmanian, Grant. TKDE'07]

- Declarative framework for the representation and processing of probabilistic spatio-temporal databases with uncertain probabilities.

## Example




A SPOT database:

$(id_1, d, 1, [0.9, 1])$
$(id_1, b, 2, [0.6, 1])$
$(id_1, c, 2, [0.7, 0.8])$
$(id_2, b, 1, [0.5, 0.9])$
$(id_2, e, 2, [0.2, 0.5])$

# SPOT databases

Several problems have been investigated:

- **Consistency Checking** [Parker et al. TKDE'09]:
  Does a given SPOT database have a model?
  - Efficient algortihms.

# SPOT databases

Several problems have been investigated:

- **Consistency Checking** [Parker et al. TKDE'09]:
  Does a given SPOT database have a model?
  - Efficient algortihms.
- **Selection Queries** [Parker et al. TKDE'09]:
  Given a region $r$ and a probability interval $[\ell, u]$, find all pairs $(id, t)$ s.t. object $id$ is at time $t$ inside region $r$ with a probability in $[\ell, u]$.
  - Two semantics: optimistic and cautious.
  - Efficient evaluation.

# SPOT databases

Several problems have been investigated:

- **Consistency Checking** [Parker et al. TKDE'09]:
  Does a given SPOT database have a model?
  - Efficient algortihms.
- **Selection Queries** [Parker et al. TKDE'09]:
  Given a region $r$ and a probability interval $[\ell, u]$, find all pairs $(id, t)$ s.t. object $id$ is at time $t$ inside region $r$ with a probability in $[\ell, u]$.
  - Two semantics: optimistic and cautious.
  - Efficient evaluation.
- **Belief revision** [Grant et al. AIJ'10]:
  Given a SPOT database $D$ and a new SPOT atom $A$ (to be added to $D$), if $D \cup \{A\}$ is inconsistent, then "revise" $D$ into a new database $D'$ so that $D' \cup \{A\}$ is consistent.

# SPOT databases

Several problems have been investigated:

- **Consistency Checking** [Parker et al. TKDE'09]:
  Does a given SPOT database have a model?
  - Efficient algortihms.
- **Selection Queries** [Parker et al. TKDE'09]:
  Given a region $r$ and a probability interval $[\ell, u]$, find all pairs $(id, t)$ s.t. object $id$ is at time $t$ inside region $r$ with a probability in $[\ell, u]$.
  - Two semantics: optimistic and cautious.
  - Efficient evaluation.
- **Belief revision** [Grant et al. AIJ'10]:
  Given a SPOT database $D$ and a new SPOT atom $A$ (to be added to $D$), if $D \cup \{A\}$ is inconsistent, then "revise" $D$ into a new database $D'$ so that $D' \cup \{A\}$ is consistent.
- **Full logic** [Doder, Grant, Ognjanović. J. Log. Comput.'13]:
  More expressive language with with negation, disjunction, and quantifiers.

# Contribution

**Count Queries** in the SPOT framework:
how many objects are in a certain region at a given time point?

- Syntax and three alternative semantics
  - *Expected value semantics*
  - *Extreme values semantics*
  - *Ranking semantics*
- Properties
- Algorithms
- Complexity

# SPOT databases

Notation

- *ID* is the set of all object ids.
- *Space* is a grid of $N \times N$ points.
- $T$ is the set of time points.

Assumptions:

- An object can be in only one location at a time.
- A location may contain more than one object.

# SPOT databases - **Syntax**

### Definition

A **SPOT atom** is a tuple $(id, r, t, [\ell, u])$ where

- *id* is an object id,
- $r$ is a region,
- $t$ is a time point,
- $[\ell, u] \subseteq [0, 1]$ is a probability interval.

# SPOT databases - **Syntax**

### Definition

A **SPOT atom** is a tuple $(id, r, t, [\ell, u])$ where

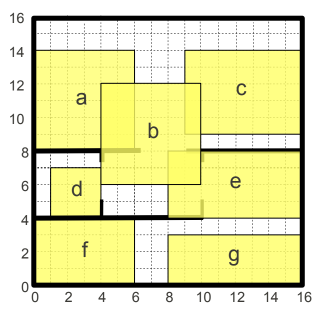
- *id* is an object id,
- $r$ is a region,
- $t$ is a time point,
- $[\ell, u] \subseteq [0, 1]$ is a probability interval.

### Definition

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

# SPOT databases - **Syntax**

## Example




A SPOT database:

$(id_1, d, 1, [0.9, 1])$
$(id_1, b, 2, [0.6, 1])$
$(id_1, c, 2, [0.7, 0.8])$
$(id_2, b, 1, [0.5, 0.9])$
$(id_2, e, 2, [0.2, 0.5])$

# SPOT databases - **Semantics**

## Definition

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

$$\sum_{p \in Space} I(id, p, t) = 1$$

# SPOT databases - **Semantics**

## Definition

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

$$\sum_{p \in Space} I(id, p, t) = 1$$

## Definition

A SPOT interpretation $I$ **satisfies** a SPOT atom $(id, r, t, [\ell, u])$ iff

$$\sum_{p \in r} I(id, p, t) \in [\ell, u]$$

# SPOT databases - **Semantics**

## Definition

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

$$\sum_{p \in Space} I(id, p, t) = 1$$

## Definition

A SPOT interpretation $I$ **satisfies** a SPOT atom $(id, r, t, [\ell, u])$ iff

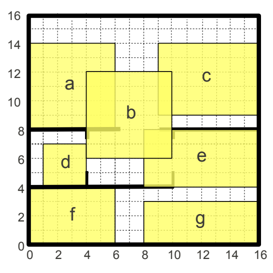
$$\sum_{p \in r} I(id, p, t) \in [\ell, u]$$

## Definition

A SPOT interpretation $I$ is a **model** for a SPOT database $D$ iff $I$ satisfies every SPOT atom in $D$.

# SPOT databases - **Semantics**

## Example

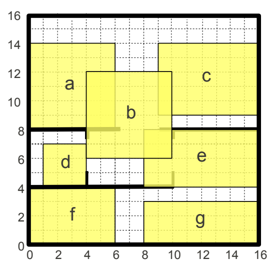



SPOT database $D$:

$(id_1, d, 1, [0.9, 1])$
$(id_1, b, 2, [0.6, 1])$
$(id_1, c, 2, [0.7, 0.8])$
$(id_2, b, 1, [0.5, 0.9])$
$(id_2, e, 2, [0.2, 0.5])$

# SPOT databases - **Semantics**

## Example




SPOT database $D$:

$(id_1, d, 1, [0.9, 1])$
$(id_1, b, 2, [0.6, 1])$
$(id_1, c, 2, [0.7, 0.8])$
$(id_2, b, 1, [0.5, 0.9])$
$(id_2, e, 2, [0.2, 0.5])$

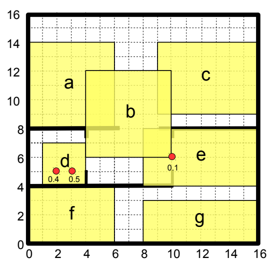
Interpretation $I$ defined as follows is a model of $D$

$I(id_1, (2,5), 1) = 0.4$ $\quad$ $I(id_1, (3,5), 1) = 0.5$ $\quad$ $I(id_1, (10,6), 1) = 0.1$
$I(id_1, (10,10), 2) = 0.7$ $\quad$ $I(id_1, (1,1), 2) = 0.3$
$I(id_2, (7,8), 1) = 0.7$ $\quad$ $I(id_2, (11,12), 1) = 0.3$
$I(id_2, (9,7), 2) = 0.3$ $\quad$ $I(id_2, (12,15), 2) = 0.7$
$I(id, p, t) = 0$ for all triplets $(id, p, t)$ not mentioned above.

# SPOT databases - **Semantics**

## Example

object $id_1$
at time point 1




SPOT database $D$:

$(id_1, d, 1, [0.9, 1])$
$(id_1, b, 2, [0.6, 1])$
$(id_1, c, 2, [0.7, 0.8])$
$(id_2, b, 1, [0.5, 0.9])$
$(id_2, e, 2, [0.2, 0.5])$

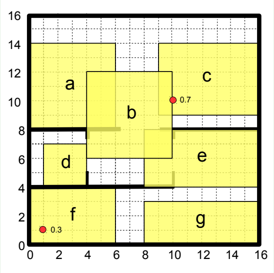
Interpretation $I$ defined as follows is a model of $D$

$\mathbf{I(id_1, (2,5), 1) = 0.4}$ $\mathbf{I(id_1, (3,5), 1) = 0.5}$ $\mathbf{I(id_1, (10,6), 1) = 0.1}$
$I(id_1, (10,10), 2) = 0.7$ $I(id_1, (1,1), 2) = 0.3$
$I(id_2, (7,8), 1) = 0.7$ $I(id_2, (11,12), 1) = 0.3$
$I(id_2, (9,7), 2) = 0.3$ $I(id_2, (12,15), 2) = 0.7$
$I(id, p, t) = 0$ for all triplets $(id, p, t)$ not mentioned above.

# SPOT databases - **Semantics**

## Example

object $id_1$
at time point 2




SPOT database $D$:

$(id_1, d, 1, [0.9, 1])$
$(id_1, b, 2, [0.6, 1])$
$(id_1, c, 2, [0.7, 0.8])$
$(id_2, b, 1, [0.5, 0.9])$
$(id_2, e, 2, [0.2, 0.5])$

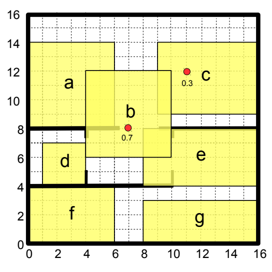
Interpretation $I$ defined as follows is a model of $D$

$I(id_1, (2,5), 1) = 0.4$ $\quad$ $I(id_1, (3,5), 1) = 0.5$ $\quad$ $I(id_1, (10,6), 1) = 0.1$
$\mathbf{I(id_1, (10,10), 2) = 0.7}$ $\quad$ $\mathbf{I(id_1, (1,1), 2) = 0.3}$
$I(id_2, (7,8), 1) = 0.7$ $\quad$ $I(id_2, (11,12), 1) = 0.3$
$I(id_2, (9,7), 2) = 0.3$ $\quad$ $I(id_2, (12,15), 2) = 0.7$
$I(id, p, t) = 0$ for all triplets $(id, p, t)$ not mentioned above.

# SPOT databases - **Semantics**

## Example

object $id_2$
at time point 1




SPOT database $D$:

$(id_1, d, 1, [0.9, 1])$
$(id_1, b, 2, [0.6, 1])$
$(id_1, c, 2, [0.7, 0.8])$
$(id_2, b, 1, [0.5, 0.9])$
$(id_2, e, 2, [0.2, 0.5])$

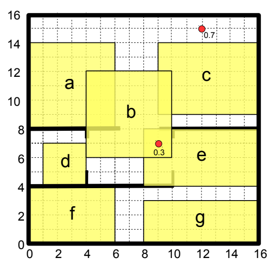
Interpretation $I$ defined as follows is a model of $D$

$I(id_1, (2, 5), 1) = 0.4$ $\quad$ $I(id_1, (3, 5), 1) = 0.5$ $\quad$ $I(id_1, (10, 6), 1) = 0.1$
$I(id_1, (10, 10), 2) = 0.7$ $\quad$ $I(id_1, (1, 1), 2) = 0.3$
$\mathbf{I(id_2, (7, 8), 1) = 0.7}$ $\quad$ $\mathbf{I(id_2, (11, 12), 1) = 0.3}$
$I(id_2, (9, 7), 2) = 0.3$ $\quad$ $I(id_2, (12, 15), 2) = 0.7$
$I(id, p, t) = 0$ for all triplets $(id, p, t)$ not mentioned above.

# SPOT databases - **Semantics**

## Example

object $id_2$
at time point 2




SPOT database $D$:

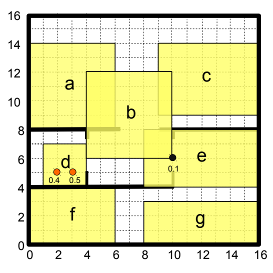
$(id_1, d, 1, [0.9, 1])$
$(id_1, b, 2, [0.6, 1])$
$(id_1, c, 2, [0.7, 0.8])$
$(id_2, b, 1, [0.5, 0.9])$
$(id_2, e, 2, [0.2, 0.5])$

Interpretation $I$ defined as follows is a model of $D$

$I(id_1, (2,5), 1) = 0.4$ $\quad$ $I(id_1, (3,5), 1) = 0.5$ $\quad$ $I(id_1, (10,6), 1) = 0.1$
$I(id_1, (10,10), 2) = 0.7$ $\quad$ $I(id_1, (1,1), 2) = 0.3$
$I(id_2, (7,8), 1) = 0.7$ $\quad$ $I(id_2, (11,12), 1) = 0.3$
$\mathbf{I(id_2, (9,7), 2) = 0.3}$ $\quad$ $\mathbf{I(id_2, (12,15), 2) = 0.7}$
$I(id, p, t) = 0$ for all triplets $(id, p, t)$ not mentioned above.

# SPOT databases - **Semantics**

## Example




SPOT database $D$:

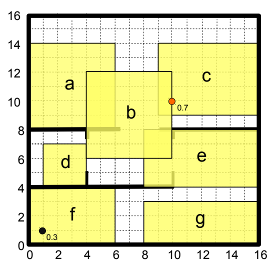
$\mathbf{(id_1, d, 1, [0.9, 1])}$
$(id_1, b, 2, [0.6, 1])$
$(id_1, c, 2, [0.7, 0.8])$
$(id_2, b, 1, [0.5, 0.9])$
$(id_2, e, 2, [0.2, 0.5])$

Interpretation $I$ defined as follows is a model of $D$

$\mathbf{I(id_1, (2, 5), 1) = 0.4}$ $\mathbf{I(id_1, (3, 5), 1) = 0.5}$ $I(id_1, (10, 6), 1) = 0.1$
$I(id_1, (10, 10), 2) = 0.7$ $I(id_1, (1, 1), 2) = 0.3$
$I(id_2, (7, 8), 1) = 0.7$ $I(id_2, (11, 12), 1) = 0.3$
$I(id_2, (9, 7), 2) = 0.3$ $I(id_2, (12, 15), 2) = 0.7$
$I(id, p, t) = 0$ for all triplets $(id, p, t)$ not mentioned above.

# SPOT databases - **Semantics**

## Example




SPOT database $D$:

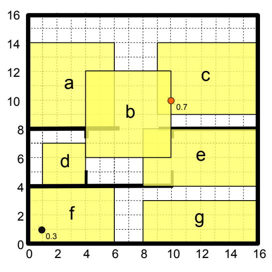
$(id_1, d, 1, [0.9, 1])$
$(\mathbf{id_1}, \mathbf{b}, \mathbf{2}, [\mathbf{0.6}, \mathbf{1}])$
$(id_1, c, 2, [0.7, 0.8])$
$(id_2, b, 1, [0.5, 0.9])$
$(id_2, e, 2, [0.2, 0.5])$

Interpretation $I$ defined as follows is a model of $D$

$I(id_1, (2,5), 1) = 0.4$ $\quad I(id_1, (3,5), 1) = 0.5$ $\quad I(id_1, (10,6), 1) = 0.1$
$\mathbf{I(id_1, (10, 10), 2) = 0.7}$ $\quad I(id_1, (1,1), 2) = 0.3$
$I(id_2, (7,8), 1) = 0.7$ $\quad I(id_2, (11,12), 1) = 0.3$
$I(id_2, (9,7), 2) = 0.3$ $\quad I(id_2, (12,15), 2) = 0.7$
$I(id, p, t) = 0$ for all triplets $(id, p, t)$ not mentioned above.

# SPOT databases - **Semantics**

## Example




SPOT database $D$:

$(id_1, d, 1, [0.9, 1])$
$(id_1, b, 2, [0.6, 1])$
$\mathbf{(id_1, c, 2, [0.7, 0.8])}$
$(id_2, b, 1, [0.5, 0.9])$
$(id_2, e, 2, [0.2, 0.5])$

Interpretation $I$ defined as follows is a model of $D$

$I(id_1, (2,5), 1) = 0.4$ $\quad$ $I(id_1, (3,5), 1) = 0.5$ $\quad$ $I(id_1, (10,6), 1) = 0.1$
$\mathbf{I(id_1, (10,10), 2) = 0.7}$ $\quad$ $I(id_1, (1,1), 2) = 0.3$
$I(id_2, (7,8), 1) = 0.7$ $\quad$ $I(id_2, (11,12), 1) = 0.3$
$I(id_2, (9,7), 2) = 0.3$ $\quad$ $I(id_2, (12,15), 2) = 0.7$
$I(id, p, t) = 0$ for all triplets $(id, p, t)$ not mentioned above.

# Count Queries in SPOT databases - **Syntax**

**Definition**

A **count query** is an expression of the form

$$Count(r, t)$$

where $r$ is a region (i.e., a subset of *Space*) and $t$ is a time point.

Intuitively, $Count(r, t)$ asks:
"How many objects are inside region $r$ at time $t$?".

# Count Queries in SPOT databases - **Semantics**

We propose three alternative semantics for interpreting count queries:

1. the *expected value semantics*,
2. the *extreme values semantics*,
3. the *ranking semantics*.

# Expected Value Semantics

Basic idea: Given a count query $Count(r, t)$ and a SPOT database $D$,

- Define the **expected number of objects** in $r$ at time $t$ **w.r.t. to a model** $M$.
- Take the **minimum and maximum expected number of objects across all models** of $D$.

# Expected Value Semantics

Consider a count query $Count(r, t)$ and a SPOT database $D$ with $n$ objects.

## Definition

Let $M$ be a model for $D$ and $X_M$ a random variable representing the number of objects in region $r$ at time $t$ according to $M$.
The expected number of objects in $r$ at time $t$ w.r.t. $M$ is:

$$Q^{exp}(M) = \mathbb{E}[X_M] = \sum_{i=0}^{n} i \cdot \Pr(X_M = i)$$

## Definition (Expected value semantics)

The **expected value answer** is $[c, C]$ where:

$$c = \min_{M \text{ is a model of } D} Q^{exp}(M) \quad \text{and} \quad C = \max_{M \text{ is a model of } D} Q^{exp}(M)$$

# Expected Value Semantics

Consider a count query $Count(r, t)$ and a SPOT database $D$.

**Proposition**

*If* $[c, C]$ *is the expected value answer, then* $\forall v \in [c, C]$ *there exists a model* $M$ *of* $D$ *s.t.* $Q^{exp}(M) = v$.

# Extreme Values Semantics

Basic idea: Given a count query $Count(r, t)$ and a SPOT database $D$, return **the lowest and the highest numbers of objects** that can be inside region $r$ at time $t$ (according to the different models of $D$).

# Extreme Values Semantics

Consider a count query $Count(r, t)$ and a SPOT database $D$.

## Definition (Extreme values semantics)

The **extreme values answer** is $[z, Z]$ where:

$$z = \min_{M \text{ is a model of } D} |\{id \mid M(id, r, t) = 1\}|$$

$$Z = \max_{M \text{ is a model of } D} |\{id \mid M(id, r, t) \neq 0\}|$$

# Extreme Values Semantics

Consider a count query $Count(r, t)$ and a SPOT database $D$.

## Definition (Extreme values semantics)

The **extreme values answer** is $[z, Z]$ where:

$$z = \min_{M \text{ is a model of } D} |\{id \mid M(id, r, t) = 1\}|$$

$$Z = \max_{M \text{ is a model of } D} |\{id \mid M(id, r, t) \neq 0\}|$$

## Proposition

*If $[c, C]$ is the expected value answer and $[z, Z]$ is the extreme value answer, then $z \leq c \leq C \leq Z$.*

# Ranking Semantics

<u>Basic idea:</u> Given a count query $Count(r, t)$ and a SPOT database $D$ with $n$ objects, return a set of pairs

$$\begin{array}{c} \langle 0, [\ell_0, u_0] \rangle \\ \langle 1, [\ell_1, u_1] \rangle \\ \langle 2, [\ell_2, u_2] \rangle \\ \langle 3, [\ell_3, u_3] \rangle \\ \vdots \\ \langle n, [\ell_n, u_n] \rangle \end{array}$$

where $[\ell_i, u_i]$ is a probability interval for exactly $i$ objects being in region $r$ at time $t$.

# Ranking Semantics

We assume independence of events involving the locations of different objects.
Consider a count query $Count(r, t)$ and a SPOT database $D$ with $n$ objects.

## Definition

$$Prob^{min}(r,t,i) = \min_{M \text{ is a model of } D} Prob_M(r,t,i) \qquad \text{for } 0 \leq i \leq n$$

$$Prob^{max}(r,t,i) = \max_{M \text{ is a model of } D} Prob_M(r,t,i) \qquad \text{for } 0 \leq i \leq n$$

where $Prob_M(r,t,i)$ is the probability of having exactly $i$ objects in $r$ at time $t$ w.r.t. model $M$, i.e.,

$$\sum_{\substack{S \text{ is a set of ids} \\ \text{and } |S| = i}} \left( \prod_{id \in S} M(id,r,t) \cdot \prod_{id \in \{\text{all ids}\} \setminus S} (1 - M(id,r,t)) \right)$$

# Ranking Semantics

Consider a count query $Count(r, t)$ and a SPOT database $D$ with $n$ objects.

## Definition

The **ranking answer** is

$$
\begin{array}{ll}
\langle 0, [\ell_0, u_0] \rangle & \text{where } \ell_0 = Prob^{min}(r, t, 0) \text{ and } u_0 = Prob^{max}(r, t, 0) \\
\langle 1, [\ell_1, u_1] \rangle & \text{where } \ell_1 = Prob^{min}(r, t, 1) \text{ and } u_1 = Prob^{max}(r, t, 1) \\
 & \vdots \\
\langle n, [\ell_n, u_n] \rangle & \text{where } \ell_n = Prob^{min}(r, t, n) \text{ and } u_n = Prob^{max}(r, t, n)
\end{array}
$$

# Ranking Semantics

Consider a count query $Count(r, t)$ and a SPOT database $D$ with $n$ objects.

## Definition

The **ranking answer** is

$$
\begin{array}{ll}
\langle 0, [\ell_0, u_0] \rangle & \text{where } \ell_0 = Prob^{min}(r, t, 0) \text{ and } u_0 = Prob^{max}(r, t, 0) \\
\langle 1, [\ell_1, u_1] \rangle & \text{where } \ell_1 = Prob^{min}(r, t, 1) \text{ and } u_1 = Prob^{max}(r, t, 1) \\
 & \vdots \\
\langle n, [\ell_n, u_n] \rangle & \text{where } \ell_n = Prob^{min}(r, t, n) \text{ and } u_n = Prob^{max}(r, t, n)
\end{array}
$$

## Proposition

*For a* **simple** *SPOT database (i.e., with a single model)*

- *The expected answer is* $[\sum_{i=0}^{n} i \cdot \ell_i, \ \ \sum_{i=0}^{n} i \cdot u_i]$
- *The extreme answer is* $[\min\{i \mid 0 \leq i \leq n \wedge \ell_i = 1\}, \ \max\{i \mid 0 \leq i \leq n \wedge \ell_i \neq 0\}]$

# Algorithms

1. Algorithm to compute the expected value semantics
   - It leverages a linear program derived from the SPOT database
   - Polynomial time
2. Algorithm to compute the extreme values semantics
   - It leverages a linear program derived from the SPOT database
   - Polynomial time
3. Algorithm to compute the ranking semantics
   - Exponential time algorithm
   - Polynomial time algorithm for *simple* SPOT database (i.e., admitting a single model)

# Algorithms

## Definition

Given a SPOT database $D$, an object $id$, and a time point $t$, $LC(D, id, t)$ is the linear program consisting of the following linear constraints:

$$\ell \leq \sum_{p \in r} v_p \leq u \qquad \text{for each } (id, r, t, [\ell, u]) \in D$$

$$v_p \geq 0 \qquad \text{for each location } p \in Space$$

$$\sum_{p \in Space} v_p = 1$$

# Computing expected value semantics

Consider a count query $Count(r, t)$ and a SPOT database $D$ with $n$ objects.

**Theorem**

*The expected value answer* $[c, C]$ *can be computed as*

$$c = \sum_{id=1}^{n} \left( \textbf{minimize } \sum_{p \in r} v_p \textbf{ subject to } LC(D, id, t) \right)$$

$$C = \sum_{id=1}^{n} \left( \textbf{maximize } \sum_{p \in r} v_p \textbf{ subject to } LC(D, id, t) \right)$$

# Computing expected value semantics

Consider a count query $Count(r, t)$ and a SPOT database $D$ with $n$ objects.

## Theorem

*The expected value answer* $[c, C]$ *can be computed as*

$$c = \sum_{id=1}^{n} \Big( \textbf{minimize } \textstyle\sum_{p \in r} v_p \textbf{ subject to } LC(D, id, t) \Big)$$

$$C = \sum_{id=1}^{n} \Big( \textbf{maximize } \textstyle\sum_{p \in r} v_p \textbf{ subject to } LC(D, id, t) \Big)$$

## Corollary

*The expected value answer can be computed in time* $O(n \cdot (|Space| \cdot |D|)^3)$.

# Computing extreme value semantics

Consider a count query $Count(r, t)$ and a SPOT database $D$ with $n$ objects.

**Theorem**

*The extreme value answer $[z, Z]$ can be computed as*

$$z = |\{id \text{ appears in } D \text{ and } (\textbf{minimize } \sum_{p \in r} v_p \textbf{ subject to } LC(D, id, t)) = 1\}|$$
$$Z = |\{id \text{ appears in } D \text{ and } (\textbf{maximize } \sum_{p \in r} v_p \textbf{ subject to } LC(D, id, t)) \neq 0\}|$$

# Computing extreme value semantics

Consider a count query $Count(r,t)$ and a SPOT database $D$ with $n$ objects.

## Theorem

*The extreme value answer $[z,Z]$ can be computed as*

$$z = |\{id \text{ appears in } D \text{ and } (\textbf{minimize} \sum_{p\in r} v_p \textbf{ subject to } LC(D,id,t)) = 1\}|$$
$$Z = |\{id \text{ appears in } D \text{ and } (\textbf{maximize} \sum_{p\in r} v_p \textbf{ subject to } LC(D,id,t)) \neq 0\}|$$

## Corollary

*The extreme value answer can be computed in time $O(n\cdot(|Space|\cdot|D|)^3)$.*

# Computing ranking semantics

## Proposition

*The ranking answer* $\langle 0, [\ell_0, u_0]\rangle, \ldots, \langle n, [\ell_n, u_n]\rangle$ *can be computed as*

$$\ell_i = \textbf{minimize} \sum_{\substack{S \text{ is a set of ids} \\ \text{and } |S| = i}} \left( \prod_{id \in S} \sum_{p \in r} v_p^{id} \cdot \prod_{id \in \{\text{all ids}\} \setminus S} \left(1 - \sum_{p \in r} v_p^{id}\right) \right)$$

**subject to**

$$LC(D, id_1, t) \cup \cdots \cup LC(D, id_n, t)$$

$$u_i = \textbf{maximize} \sum_{\substack{S \text{ is a set of ids} \\ \text{and } |S| = i}} \left( \prod_{id \in S} \sum_{p \in r} v_p^{id} \cdot \prod_{id \in \{\text{all ids}\} \setminus S} \left(1 - \sum_{p \in r} v_p^{id}\right) \right)$$

**subject to**

$$LC(D, id_1, t) \cup \cdots \cup LC(D, id_n, t)$$

# Computing ranking semantics

For **simple** SPOT databases (i.e., with one single model) we have a polynomial-time dynamic programming algorithm.

## Definition

$$
\begin{array}{lll}
Prob_M(r,t,0,j) = & \prod_{k=1}^{j}(1 - M(id_k,r,t)) & 1 \leq j \leq n \\
Prob_M(r,t,j,j) = & \prod_{k=1}^{j} M(id_k,r,t) & 1 \leq j \leq n \\
Prob_M(r,t,i,j) = & M(id_j,r,t) \cdot Prob_M(r,t,i-1,j-1) + & \\
 & (1 - M(id_j,r,t)) \cdot Prob_M(r,t,i,j-1) & 2 \leq j \leq n, 1 \leq i \leq j-1
\end{array}
$$

# Computing ranking semantics

For **simple** SPOT databases (i.e., with one single model) we have a polynomial-time dynamic programming algorithm.

## Definition

$$
\begin{array}{lll}
Prob_M(r,t,0,j) = & \prod_{k=1}^{j}(1 - M(id_k, r, t)) & 1 \leq j \leq n \\
Prob_M(r,t,j,j) = & \prod_{k=1}^{j} M(id_k, r, t) & 1 \leq j \leq n \\
Prob_M(r,t,i,j) = & M(id_j, r, t) \cdot Prob_M(r,t,i-1,j-1) + & \\
 & (1 - M(id_j, r, t)) \cdot Prob_M(r,t,i,j-1) & 2 \leq j \leq n, 1 \leq i \leq j-1
\end{array}
$$

## Theorem

*The ranking answer* $\langle 0, [\ell_0, u_0]\rangle, \ldots, \langle n, [\ell_n, u_n]\rangle$ *can be computed as*

$$\ell_i = u_i = Prob_M(r,t,i,n)$$

*for simple SPOT databases.*

# Computing ranking semantics

**Corollary**

*The ranking answer can be computed in time* $O(n \cdot (|Space| \cdot |D|)^3)$.

# Conclusion

**Count queries** in the SPOT framework

- Three alternative semantics
  1. Expected value semantics
  2. Extreme values semantics
  3. Ranking semantics
- Properties, Algorithms, Complexity

# Conclusion

**Count queries** in the SPOT framework

- Three alternative semantics
  1. Expected value semantics
  2. Extreme values semantics
  3. Ranking semantics
- Properties, Algorithms, Complexity

**Future work**

- No independence assumption for the ranking semantics
- Count queries over time intervals
- Other kinds of count queries

THANKS!
QUESTIONS?