1. Motivation
Correlated probabilistic streams commonly occur in a large variety of applications.
A Habitat Monitoring Application

Problem: Execute continuous queries over probabilistic streams
Example queries:
• What is the weekly average of the bird count at location A?
• What is the likelihood that A has more birds than B for the past week?
• List all days in which A has more than 100 birds
Issues:
• Correlations significantly influence results of query evaluation
• SQL query semantics are ambiguous when results are probabilistic sequences

2. Previous work
• Probabilistic databases (Mystiq, Trio, Orion, MayBMS) focus on static data and not streams, also cannot handle these complex correlations
• Luban, Calypso are applicable to probabilistic streams, but focus on pattern identification queries

3. Our Approach
Observation: Many naturally occurring probabilistic streams are both structured and Markovian in nature.
• Structured: Same set of correlations and independences repeat across time
• Markovian: The values of variables at time \( t+1 \) is independent of those at times \( t \) and \( t-1 \) given their values at time \( t \)
We exploit this to:
• Design compact, lightweight data structures to represent and modify them
• Enable incremental query processing using the iterator framework

4. Markov sequence
• Special case of a probabilistic sequence: Completely determined by specifying successive joint distributions for all time instants \( p(V_t, V_{t+1}) \)
• Efficient Representation: We represent Markov sequences using a combination of the schema graph and the actual data

\[
\begin{align*}
X_t &\sim p(X_t | X_{t-1}) \\
Y_t &\sim p(Y_t | X_t) \\
V_t &\sim p(V_t | X_t)
\end{align*}
\]

(a) Schema graph (b) Distributions representing the Markov sequence

• Efficient Query Processing: Only transmit these numbers between operators. Only operators with the knowledge of the schema can interpret these numbers

5. Formal Semantics
Possible World Semantics (+ small modification for sequences)
Operators: Select, Project, Joins, Aggregate, Windowing
• The set of Markov sequences is not closed under these operators, i.e., some operators return non-Markovian sequences (Projection, Windowing)

6. Operator Design
• Schema routine: Output schema is computed based on the input schema
• get_next() routine: Operates on each of the input data tuples to compute the output tuples.
Selection: Add new boolean random variable to each slice; Always safe
Projection: Eliminate all variables not of interest; Can be unsafe for some schemas
Aggregation: Maintain the joint distribution of the aggregate variable along with other variables in the time slice; Always safe

7. Query Planning Algorithm
• Unsafe Operators: Projection, Window operator reduced to a projection
• Query Planning = Determining the correct position for Projection
• Strategy: Pull up Projection operator until it is safe; If no safe position for projection, check its parent. ML + we can determine a safe plan, MAP + we cannot determine a safe plan

8. Results
Markov sequence generator: Generate Markov sequences for different schemas with correlations ranging from (indep., perfectly correlated)