LBL-27435
UC-405

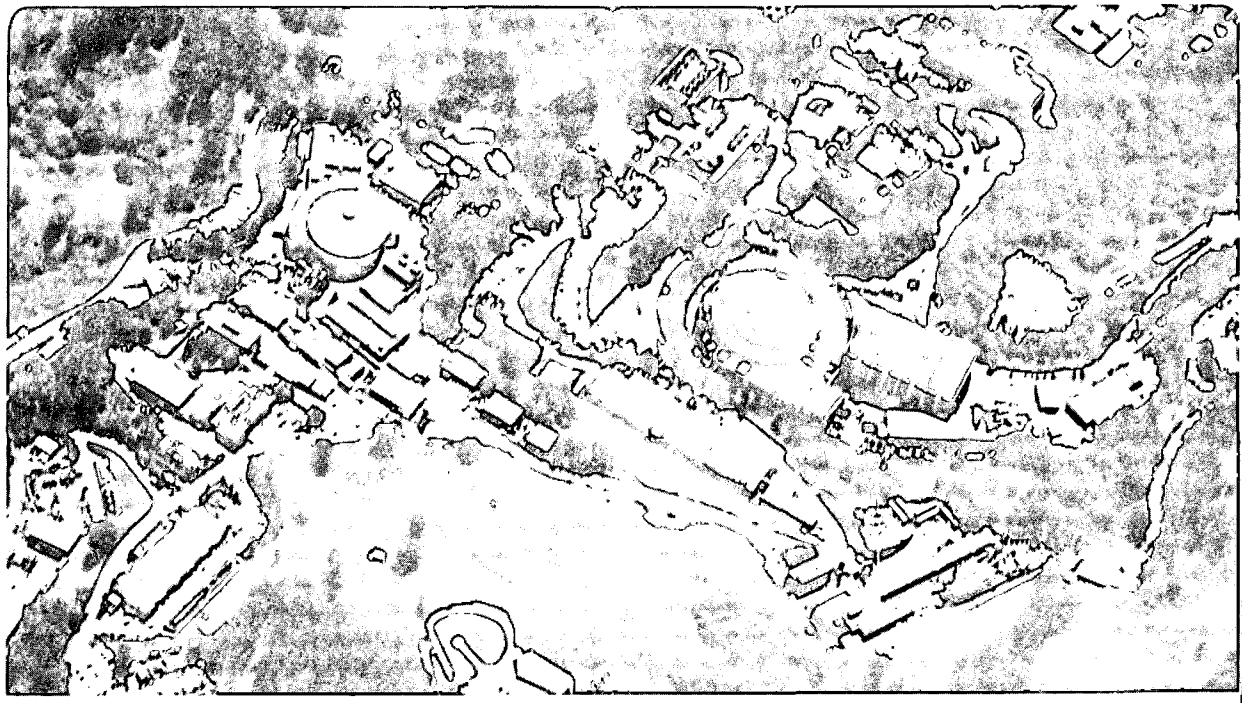
Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Information and Computing Sciences Division

Selectivity Estimation in Temporal Databases

H. Gunadhi, A. Segev, and J.G. Shanthikumar

February 1990



Prepared for the U.S. Department of Energy under Contract Number DE-AC03-76SF00098.

DISCLAIMER

LBL-27435

SELECTIVITY ESTIMATION IN TEMPORAL DATABASES

Himawan Gunadhi, Arie Segev and J. George Shanthikumar

Computing Science Research & Development
Information & Computing Sciences Division
Lawrence Berkeley Laboratory
1 Cyclotron Road
Berkeley, California 94720

and

Walter A. Haas School of Business
The University of California, Berkeley
Berkeley, California 94720

February 1990

This work was supported by the Director, Office of Energy Research, Applied Mathematical Sciences Research Program, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

LBL-27435
June 1989

SELECTIVITY ESTIMATION IN TEMPORAL DATABASES

Himawan Gunadhi, Arie Segev and J. George Shanthikumar

Walter A. Haas School of Business
The University of California and
Computing Sciences Research and Development Department
Lawrence Berkeley Laboratory
Berkeley, California 94720

Revised Feb. 1990

Abstract

Temporal relations possess several characteristics that distinguish them from conventional snapshot relations. First, for each instance of the surrogate (entity) there is a set of time-ordered tuples. Second, surrogate instances may arrive and depart in some time-dependent manner. Third, the surrogate instance may arrive and depart more than once, thus creating gaps (null values) within its history. Lastly, the value of the temporal attribute may be also be time-dependent. Conventional methods of estimation are incapable of providing good approximations of the cost of various temporal operations, even for those involving selections on a single relation. The problem is more acute in the case of join operations, since selectivities on time interval intersections have to be estimated. We propose a practical, yet theoretically sound model to characterize the behavior of temporal relations. Estimates of the outcome for various unary and binary operations are derived from this model. Preliminary results on the accuracy of selected estimates are provided.

This work was supported by the Applied Mathematical Sciences Research Program of the Office of Energy Research, U.S. Department of Energy under Contract DE-AC03-76SF00098.

1. INTRODUCTION

Accurate cost estimation of relational operations is crucial to query optimization. A substantial amount of literature exists on selectivity estimation, among them by [Yao 77, Selinger et al 79, Christodoulakis 83, Piatetsky-Shapiro & Connell 84, Graefe 87, Lynch 88, Mularikrishna & Dewitt 88, Ahad et al 89]. However, estimation techniques for snapshot relations cannot be readily applied to the context of temporal relations, due to certain distinguishing properties of the latter. First, each relation consists of time-ordered histories for instances of the surrogate (entity). Second, histories of the surrogate instances may begin and end at different points in time. Third, some of the histories may contain disjoint intervals, i.e., there are gaps in the history for which no data exist. Fourth, the temporal attributes themselves may also be time-dependent in their behavior. Conventional methods of estimation are based on the assumption that tuples within a relation are independent of one another, which is reasonable in the case of snapshot relations; the focus of research is on the likely distribution of the attribute values. Clearly, without modeling some or all of the temporal properties explicitly, simple extension of existing methods to the temporal context would yield poor results. Furthermore, many temporal operations are based on intersections between time intervals, e.g., joins between two relations over the time domain or selection of tuples in a relation over a query interval. Any estimates of such operations would necessitate explicit consideration of the temporal behavior of relations.

The optimization of temporal operations [Snodgrass & Ahn 87, Gunadhi & Segev 88, Leung & Muntz 89, Segev & Gunadhi 89a, 89b] is more critical than that for snapshot relations, as the size of data and complexity of operations are greater. Yet no study has been carried out on selectivity estimation. In this paper we introduce a model that deals with (1) the arrival process for the surrogate class, (2) the arrival process of tuples for each instance of the surrogate, (3) the existence of disjoint histories, (4) the distribution of temporal attribute values and (5) the length of a surrogate instance's lifespan (history). We will then derive the unary and binary estimates from the model. This technique is practical enough to implement, yet has sound theoretical foundations. The contributions of this paper are the following.

- A detailed framework for discussing the issues involved in selectivity estimation of temporal relational operations.
- Development of a mathematical model to characterize the general behavior of a temporal relation.
- Derivation of estimates for the sizes resulting from both unary and binary temporal operations.
- Tests on the accuracy of selected estimates and a comparison with conventional estimates.

The rest of the paper is organized as follows. Section 2 introduces a framework to help understand the relational representation of temporal data and issues involved with their mathematical modeling. Section 3 discusses the model we develop and its underlying assumptions. Section 4 provides the derivations of unary estimates, while binary estimates are derived in Section 5. In Section 6 we present test results on the accuracy of some of the unary estimates, and Section 7 offers concluding remarks and an outline of future research.

2. REPRESENTATION AND MODELING OF TEMPORAL DATA

In this section we develop the framework needed to understand the representation of temporal data as relations, basic terminology, characteristics of such relations, and the estimation measures required for relational operations, illustrated by examples.

2.1. Relational Representation of Temporal Data

A convenient way to look at temporal data is through the concepts of *Time Sequence* (*TS*) and *Time Sequence Collection* (*TSC*) [Segev & Shoshani 87]. A *TS* represents a history of a temporal attribute(s) associated with a particular instance of an entity or a relationship. The entity or relationship is identified by a *surrogate* (or equivalently, the *time–invariant key*). For example, the salary history of an employee is a *TS*. A *TS* is characterized by several properties, such as the time granularity, lifespan, type, and interpolation rule to derive data values for non-stored time points. In this paper, we focus on a common type of data -- *stepwise constant* . Stepwise constant (*SWC*) data represents a state variable whose value is determined by events and remains the same between events; the salary attribute represents *SWC* data. Time sequences of the same surrogate and attribute types can be

grouped into a time sequence collection (*TSC*), e.g. the salary history of all employees forms a *TSC*. There are various ways to represent temporal data in the relational model; detailed discussion can be found in [Segev & Shoshani 88]. In this paper we assume a representation as shown in Fig. 1, which illustrates two temporal relations, representing the MANAGER and COMMISSION histories of employees.

MANAGER

E#	MGR	T_S	T_E
E1	TOM	1	5
E1	MARK	9	12
E1	JAY	13	20
E2	RON	1	18
E3	RON	1	20

COMMISSION

E#	C_RATE	T_S	T_E
E1	10%	2	7
E1	12%	8	20
E2	8%	2	7
E2	10%	8	20

Figure 1. Representing SWC Data with Lifespan = [1, 20]

We use the terms *surrogate* (*S*), *temporal attribute* (*A*), and *time attribute* (T_S or T_E) when referring to attributes of a relation. For example, in Fig. 1, the surrogate of the MANAGER relation† is E#, MGR is the temporal attribute, and T_S and T_E are time start and time end attributes respectively. E1 is an instance of E#, and tuples (E1, TOM, 1, 5), (E1, MARK, 9, 12) and (E1, JAY, 13, 20) represent the tuples in its history. Note that there is a discontinuity In E1's history between time 6 and 8. Thus, there were actually 4 changes in the manager status of E1. We assume that all relations are in first temporal normal form (1TNF) [Segev & Shoshani 88]. In the simplest case, the temporal relation has one temporal attribute; due to normalization reasons, this is likely to be a common manifestation of temporal relations [Navathe & Ahmed 86]. Each relation has a lifespan, which is defined by the first

† We refer to the data construct as a 'relation', but we mean a 'temporal relation'. It is different from a standard relation because of the associated meta-data.

surrogate instance arrival and last instance departure, or current time, whichever is applicable. The lifespan of a surrogate instance is the length of its history, defined by the starting time associated with the first tuple and ending time of the last tuple.

It should be emphasized that the representation of time for the *SWC* data type is dependent on the level of granularity required to capture the behavior of the temporal attribute. For the temporal attribute 'commission' in Fig. 1, we may use various levels of Gregorian calendar representation, such as year, month-year, month-day-year, and so on, depending on the requirements. In all our examples and models, we adopt an integer representation for convenience. This does not in any way imply that the underlying processes are discrete; in fact, most *SWC* data reflect continuous time processes.

2.2. Characteristics of Temporal Relations

The following are the fundamental characteristics that describe the behavior of a temporal relation.

Arrival of surrogate instances. The arrival of a new surrogate instance adds a new history to the relation. Surrogate instances arrive according to some probability distribution; for example, a company may hire 120 new employees a year, at a uniform rate of 10 a month.

Departure and re-entry of surrogate instances. After arriving, a surrogate instance may remain active for the duration of the relation's lifespan, leave permanently at some point, or leave and then re-enter later. All these may be modeled by a single stochastic process, or by separate processes. If the instance is allowed to return, we assume that no new history is generated, instead the old one is reactivated and extended, with a resulting discontinuity in its history. This is exemplified by the example of E1's history in the MANAGER relation of Fig. 1.

Arrival of tuples for a surrogate instance. The arrival process of tuples for a given surrogate instance follows some probability distribution representing the behavior of *changes* in the temporal attribute value for that instance. For a given surrogate class, each instance may have its own distribution or may share an identical distribution with other instances.

Distribution of temporal attribute values. Two consecutive tuples for a given instance's history must have different values unless there is a discontinuity in their associated time intervals. Attribute values may be time-dependent, in which case they can either be dependent on the event time itself, e.g., salaries paid based on seniority, or dependent on the value in one or more prior period(s), e.g., the value of a fixed deposit. On the other hand they may be independent of time, e.g., manager or project name.

2.3. Unary Estimates

We now look at the cost estimates needed for unary operations on a temporal relation. Cost here is represented by the number of tuples resulting from a selection or projection operation. We can characterize a temporal query as being qualified on some interval, which we call the *query interval*, $[t_s, t_e]$, a special case of which is the singular time point where $t_e = t_s$. The following measures are conditioned on the non-null *intersection* between the tuples of the relation and the query interval; a tuple x intersects the query interval when $x(T_S) \leq t_e$ and $x(T_E) \geq t_s$. There are other interval predicates that may substitute for intersection. For example, the 'equal' predicate, which is true if the tuple's time stamps match those of the query interval's. There are other relationships representing 'contained-in', 'containment', and 'overlap'; yet all these predicates are merely subsets of 'intersection' and as such will not be considered separately.

(1). *Number of surrogate instances.*

Example: "How many employees were in the company between time 1 and 12?"

(2). *Number of tuples for a surrogate instance.*

Example: "Get all the manager records for E1 between time 2 and 10."

(3). *Number of tuples that intersect with the query interval.*

Example: "Find all commission records between time 4 and 10."

(4). *Number of tuples with a given temporal attribute value for surrogate instance or relation.*

Example: "How many tuples in MANAGER have MGR = TOM between time 1 and 12?"

(5). *Range selectivities for surrogate instances and attribute values.* These are queries specifying a range of values over the surrogate or temporal attribute domains.

Example: "How many employees earned between 10K and 20K between time 1 and 20?"

2.4. Binary Estimates

Binary estimates are associated with operations that involve two relations. The operand relations may have identical or different lifespans. The following are the join sizes we are interested in.

(1). *Number of intersecting intervals for two histories.* Let H_1 and H_2 be two arbitrary histories from relations r_1 and r_2 respectively. The basic measure for all temporal joins is the number of tuples that intersect over time between any pair of histories.

(2). *The size of a temporal equijoin.* In a *temporal equijoin* [Clifford & Croker 87, Gunadhi & Segev 88], which we call *TE−join*, the result is made up of concatenated tuples that have (i) identical values on a *non−time* join attribute, and (ii) intersecting time-intervals. In other words, this join is the temporal equivalent of the snapshot equijoin. The estimation procedure would depend on whether the non-temporal join predicate involves the surrogate or the temporal attribute.

(3). *The size of an event-join.* An *event−join* [Segev & Shoshani 88, Segev & Gunadhi 89b] is used to group several temporal attributes of an entity into a single relation. As stated earlier, temporal attributes for a surrogate that change values at different times (i.e. asynchronously), are likely to be stored in separate relations for normalization purposes, but need to be composed into one for many queries. Differences in the two attributes' temporal behavior and lifespans bring the possibility that *outerjoins* are needed to compose the result. The procedure for executing an event-join is the following [Segev & Gunadhi 89b]: (1) temp1 $\leftarrow r_1$ TE-JOIN r_2 on S; (2) temp2 $\leftarrow r_1$ OUTERJOIN r_2 on S; (3) temp3 $\leftarrow r_2$ OUTERJOIN r_1 on S; (4) result $\leftarrow$ temp1 $\cup$ temp2 $\cup$ temp3. Fig. 2 shows the result of an event-join between the MANAGER and COMMISSION relations previously shown in Fig. 1.

MANAGER EVENT-JOIN COMMISSION

result	E#	MGR	C_RATE	T_S	T_E
	E1	TOM	Ø	1	1
	E1	TOM	10%	2	5
	E1	Ø	10%	6	7
	E1	Ø	12%	8	8
	E1	MARK	12%	9	12
	E1	JAY	12%	13	20
	E2	RON	Ø	1	1
	E2	RON	8%	2	7
	E2	RON	10%	8	18
	E2	Ø	10%	19	20
	E3	RON	Ø	1	20

Figure 2. Event-Join Result

2.5. Multi-Attribute Temporal Modeling

A more complex scheme for a temporal relation is one involving multiple temporal attributes $A_1, A_2, ..., A_m$. We have to consider the interdependence amongst attributes in terms of both the timing of events and value changes in temporal attributes. In general, it would not be desirable to maintain relations where the temporal attributes are not synchronous, as previously explained. If such relations are nonetheless maintained, then each new tuple indicates that at least one attribute has changed its value, but not necessarily all attributes have undergone changes. If the attributes indeed form a synchronous set, we can model them as if they form a single attribute A; in this case, the preceding discussions on modeling and measurement parameters directly apply. In this paper, we concentrate primarily on the case of a single or synchronous set of temporal attributes, and for the sake of convenience, will refer to them as the *one–attribute model*.

3. ONE-ATTRIBUTE MODEL AND ASSUMPTIONS

We propose a model with the following parameters: A surrogate instance arrival process, tuple arrival process for each instance in the surrogate class, probability distribution of the temporal

attributes, two different distributions for the lifespan of a surrogate instance and a treatment of possible discontinuities in histories. The following basic assumptions are made: (1) Lifespan information is maintained for each relation, call it LS_{r_i}; the start and end points of the lifespan are represented by $LS_{r_i}.START$ and $LS_{r_i}.END$; (2) The time domains of all temporal relations can be represented by the set of non-negative integers $\{0, 1, \cdots\}$; and (3) Granularities of the time attributes in two joining relations are identical. Additional assumptions and explanations will be provided as we proceed.

Arrival of Surrogate Instances

Let $\{N^s_{r_i}(p)\}$, $p = 0, 1, \cdots$, be the number of surrogate instances that arrive in period p for relation r_i. Assume that $\{N^s_{r_i}(p)\}$ is a Poisson counting process with arrival rate $\lambda^s_{r_i}$.

Tuple Arrivals for a Surrogate Instance

Let $\{N^h_{r_i}(p)\}$, $p = 0, 1, \cdots$, be the number of tuples that arrive in period p for an arbitrary surrogate instance in relation r_i. Then assume that $\{N^h_{r_i}(p)\}$ is a Poisson process with rate of arrival $\lambda^h_{r_i}$. Further, we assume that the counting processes for surrogate instances $s_1, s_2, \cdots$ are independent and identically distributed.

Distribution of Temporal Attribute Values

We model the value of the temporal attribute during the surrogate instance's lifespan by an i.i.d. sequence of uniform random variables over the temporal attribute domain. Although it would be incorrect to assume that for a given surrogate instance and say the temporal attribute 'salary', the value can remain the same for two successive tuples, the impact should not be significant if the domain size is large. This approach is taken to simplify estimation, since time-dependent characterization requires knowledge of the actual behavior of the temporal attribute.

Life-span of Each Surrogate Instance

There are two possibilities with respect to the length of a surrogate instance's history in a relation, which we denote as $LS^s_{r_i}$. For simplicity in deriving approximations, we first assume that the lifespan is deterministic in length. We then relax this assumption and model the length of history as a random variable with a general distribution.

Treatment of Null Values

The null values will be handled by using a parameter, called the existence density-- $ED_{r_i} = \frac{number\ of\ data\ points}{number\ of\ time\ points}$. Therefore $1 - ED_{r_i}$ gives us the proportion of *changes* within a given history that will generate nulls. By assuming that null values are uniformly distributed throughout each history, we can apply this constant factor to a relation to determine the total number of null changes. When this measure is applicable, then the measure of tuple arrivals per surrogate instance has to incorporate the 'arrival' of null values. In the derivations that follow, we do not consider the existence of nulls, since only the final results will be affected by a constant factor.

Discussion

The choice of Poisson characterization for the arrival processes is not incompatible with our representation of the time domain. If the temporal attribute values are recorded only at time points t and $t + 1$, this does not imply that no value exists in between them; it merely reflects the selected granularity of representation. Secondly, the Poisson property that no two arrivals occur simultaneously, does not mean that for a given time point, no two surrogate instance arrivals can be recorded; again, the time point t is assumed to capture information within the interval $[t, t + 1)$.

4. UNARY ESTIMATES

We derive selectivity estimates for unary operations on a relation in this section. The following symbols are used throughout. $|r_i|$ is the number of tuples (cardinality) of r_i. $|r_i(S)|$ is the number of unique surrogate instances in r_i. $|r_i(A)|$ is the number of unique attribute values in r_i. The query interval is the interval $[t_s, t_e]$, with $t_s \geq 0$, which we will refer to as QI; the length of the interval is $|QI| = t_e - t_s + 1$. *MTUP* is the expected number of tuples of r_i contained within QI. *MSUR* is the expected number of unique surrogate instances of r_i found within QI. *MHIS* is the expected number of tuples per surrogate instance of r_i found within QI. *MATT* is the expected number of tuples per unique temporal attribute value of r_i found within QI.

4.1. Deterministic Surrogate Instance Lifespans

We first derive unary estimates under the assumption that the lifespan of each surrogate instance is a constant k in length. In the next subsection, we relax this assumption for the case where the lifespan can take an arbitrary random distribution.

Number of Surrogate Instances

The number of valid histories is at most equal to those arrivals that took place from time $t_s - k + 1$ to time t_e. Surrogate instances that were active before $t_s - k + 1$ would have become inactive by time t_s. Given the Poisson nature of arrivals, the required measure is given by $(|QI| + k)\ \lambda^s_{r_i}$. This assumes that $t_s - k + 1$ is greater than or equal to 0, the start of the relation's lifespan. If this is not satisfied, then all histories will intersect with the query interval, and the measure is given by $(t_e + 1)\ \lambda^s_{r_i}$.

$$MSUR = \min\{|QI| + k, t_e + 1\}\ \lambda^s_{r_i}. \tag{4.1}$$

Number of Tuples in Relation and per Surrogate Instance

The total number of tuples for the relation that are found within the query interval, can be found by finding the total length of all surrogate instance lifespans that intersect with the query interval, then multiplying it by the arrival rate of tuples per surrogate. We divide arrivals into two types -- those that arrive before QI, and those that arrive within it. For the first type of arrival, if $t_s - k + 1 \geq 0$, then the number of surrogate instance arrivals still active at time t_s is $k\ \lambda^s_{r_i}$. The length of the intersecting lifespan of such a surrogate instance, selected at random, depends on the comparison between $|QI|$ and $k / 2$. If $|QI| \geq k / 2$, then the expected length of the intersection is $k / 2$; if, $|QI| < k / 2$, then the expected length is $|QI|$. Now, if $t_s - k + 1 < 0$, then the number of surrogate instances that arrived before t_s and are still active at time t_s is $(t_s + 1)\ \lambda^s_{r_i}$. The expected length of the intersecting lifespan of such a surrogate instance with QI, if $|QI| \geq k - (t_S + 1) / 2$, is $k - (t_S + 1) / 2$; if $|QI| < k - (t_S + 1) / 2$, then the expected length of the intersection is $|QI|$.

For the second type of surrogate instance arrivals, i.e., those that arrive within QI, the count of unique instances is $|QI|\ \lambda^s_{r_i}$. The expected length of the intersection depends on a comparison between k and $|QI|$. If $k \leq |QI| / 2$, then the expected length is k; otherwise it is $|QI| / 2$. Therefore, the measure we need, $MTUP$ is

$$MTUP = \left[q\ \lambda^s_{r_i} \min\left\{ k - \frac{q}{2}, |QI| \right\} + |QI|\ \lambda^s_{r_i} \min\left\{ k, \frac{|QI|}{2} \right\} \right] \lambda^h_{r_i}, \qquad (4.2)$$

where $q = \min\{k, t_s + 1\}$.

It follows that the average number of tuples of a randomly selected surrogate instance that intersects with QI is

$$MHIS = MTUP\ /\ MSUR. \qquad (4.3)$$

Number of Tuples for a Given A Value

The general procedure for finding the number of occurrences of an arbitrary value of A, call it a_j, within the query interval is the following: (1) For each surrogate instance, find the number of occurrences of a_j within that interval; (2) Summing it up over all surrogate instances gives us the desired result. Given the assumption of independent and uniform probability distribution of A values over the relation, we need not explicitly go through these steps. Instead, multiply the number of tuples within the query interval by the selectivity of A.

$$MATT = MTUP \frac{1}{|r_i(A)|}. \qquad (4.4)$$

4.2. Non-deterministic Surrogate Instance Lifespans

We now derive unary estimates for the case where the lifespan of surrogate instances follow a common general distribution G. The problem can be modeled along the lines of the *Infinite Server Queue* problem [Ross 83]. Each instance arrives in accordance with a Poisson process. Upon arrival a surrogate instance is immediately taken in for service by one of an infinite number of possible servers, and the service times are assumed to be independent with a common distribution G. In our case, the service time is the life of the surrogate instance. The first measure of interest -- the number of active surrogate instances during interval QI, can be derived by the following analysis. Let us say that an arrival is

type 1: if it arrives before time t_s and completes service between t_s and t_e,

type 2: if it arrives before t_s and completes service after t_e,

type 3: if it arrives between t_s and t_e and completes service after t_e,

type 4: if it arrives between t_s and t_e and completes service before t_e,

type 5: otherwise.

Hence an arrival at time y will be type j with probability $P_j(y)$ given by

$$P_1(y) = \begin{cases} G(t_e - y) - G(t_s - y) & \text{if } y < t_s \\ 0 & \text{otherwise} \end{cases}$$

$$P_2(y) = \begin{cases} 1 - G(t_e - y) & \text{if } y < t_s \\ 0 & \text{otherwise} \end{cases}$$

$$P_3(y) = \begin{cases} 1 - G(t_e - y) & \text{if } t_s < y < t_e \\ 0 & \text{otherwise} \end{cases}$$

$$P_4(y) = \begin{cases} G(t_e - y) & \text{if } t_s < y < t_e \\ 0 & \text{otherwise} \end{cases}$$

$$P_5(y) = 1 - \sum_{j=1}^{4} P_i(y)$$

Figure 3 provides an illustration of the breakdown of the types of arrivals.

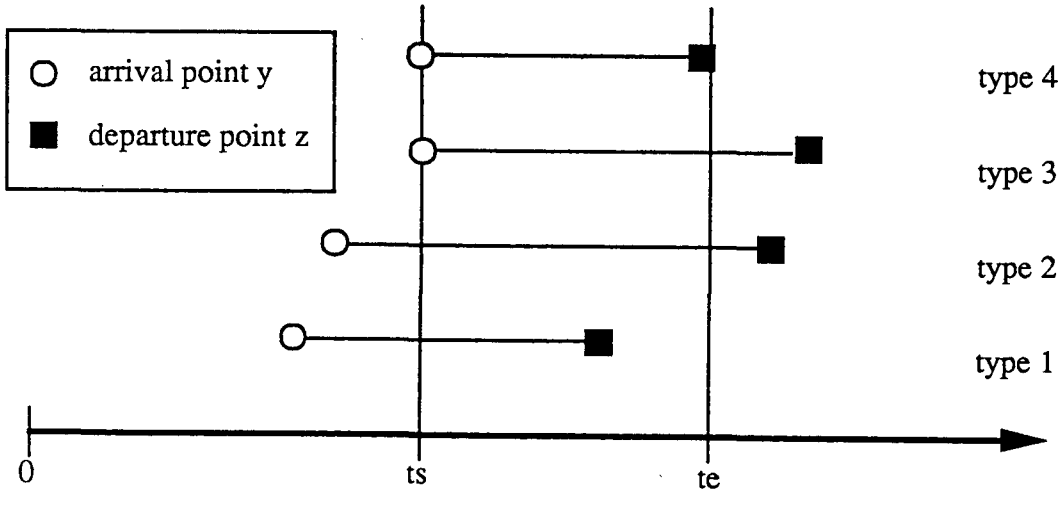


Figure 3: Illustration of Different Types of Arrivals

Let $MSUR_j$ $j = 1, ..., 4$, denote the expected number of type j events that occur. Then, applying the following Poisson property [Ross 83]:

"If $N_j(t), j = 1, ..., k$, represent the number of type j events occurring by time t then $N_j(t)$, $j = 1, ..., k$, are independent Poisson random variables having means

$$E[N_j(t)] = \lambda \int_0^t P_j(s)ds,"$$

it follows that the expected number of surrogate instances within QI of type j is

$$MSUR_j = \lambda_{r_i}^s \int_0^{t_e} P_j(y)dy, \quad j = 1, ..., 4.$$

The mean number of surrogate instances that are active within the interval QI is therefore the sum of the means for type 1 to type 4 arrivals.

$$MSUR = \sum_{j=1}^{4} MSUR_j. \tag{4.5}$$

Now, in order to derive the expected number of tuples in the relation which intersect with QI, we have to first derive the expected *length* of a surrogate instance's lifespan that intersects with QI, for event types 1 to 4; let E_j, $j = 1, ..., 4$ represent this measure. Once the E_j's are known, we find their weighted average. Finally, the expected number of tuples that arrive within this weighted average period can be estimated, thus giving us the desired estimate. For notational convenience, let the point of departure of the surrogate instance, $LS_{r_i}^s.END$, be represented by z.

$$E_1 = \int_{t_s}^{t_e} \int_0^{t_s} (z - t_s)\, dG(z - y)\, \frac{1}{t_s}\, dy,$$

$$E_2 = |QI|,$$

$$E_3 = \int_{t_s}^{t_e} (t_e - y)\, [1 - G(t_e - y)]\, \frac{1}{t_e - t_s}\, dy,$$

$$E_4 = \int_{t_s}^{t_e} \int_{y}^{t_e} (z - y)\, dG(z - y)\, \frac{1}{t_e - t_s}\, dy.$$

We can explain the derivations in the following manner. In order to compute the expected length for each type of event, we find the intersecting portion for each type of lifespan, which in turn are conditional upon the arrival and departure times, y and z respectively. The density function of the

random variable y is uniform, given the Poisson nature of tuple arrivals. The limits of each uniform distribution is in turn dependent on the event type. Integrating over the range of y and z where necessary, we derive the four equations. The measure for the number of tuples in the relation is

$$MTUP = p\ \lambda_{r_i}^{h}, \text{ where } p = \sum_{j=1}^{4} E_j\ MSUR_j. \tag{4.6}$$

It follows that the number of tuples per history, $MHIS = MTUP\ /\ MSUR$. The measures for the total number of tuples within the interval, and the expected number of tuples with a given temporal attribute value are identical to those derived under the deterministic lifespan scenario.

5. BINARY ESTIMATES

In studying binary estimates, we need not consider any restrictions on the lifespans of the result relation, since these can be carried out as a unary operation. When the two relations have unequal starting or ending lifespans, an innerjoin must be based on the intersecting lifespans defined by

$$LS_{r_{12}} = [max\ \{LS_{r_1}.START, LS_{r_2}.START\}, \min\{LS_{r_1}.END, LS_{r_2}.END\}]$$

Thus we can assume throughout that the two relations are always joined over identical lifespans with the exception of the event-join, where outerjoins are involved. We define additional notations as follows. $M(H_i, H_j)$ is the expected number of intersection tuples resulting from the cartesian product of histories H_i and H_j. $M(r_{TE}, Y)$ is the expected size (in tuples) of the result of a *TE−join* between r_i and r_j over attribute Y. $M(r_{EJ})$ is the expected size (in tuples) of the result of an *event−join* between r_i and r_j, and $M(r_{OJ}, LS_{r_i} - LS_{r_j})$ is the result of one directional outerjoins from r_i to r_j over $LS_{r_i} - LS_{r_{ij}}$, i.e., *subintervals* within the lifespan of r_i that precede and/or succeed that of r_j.

Size of Intersection Between Two Histories

Given the two histories H_1 and H_2, we assume that they are independent of one another. Knowing that within each history, arrival times are uniformly distributed, the number of intersections between the two histories follow a uniform distribution. By induction, the following bounds are derived on the number of resulting intersections.

$\min = \max\{|H_1|, |H_2|\},$

$\max = |H_1| + |H_2| - 1.$

Where $|H_i|$ is the number of tuples in that history. As an example, if H_1 has 5 tuples and H_2 has 3, then the minimum number of intersections between the two is 5, while the maximum is 7. The mean number of intersections is therefore

$$M(H_1, H_2) = \frac{1}{2} (\max\{|H_1|, |H_2|\} + |H_1| + |H_2| - 1) \qquad (5.1)$$

Size of Temporal Equijoin

In a temporal equijoin, the joining attribute is either the surrogate S or temporal attribute A. In order to find the size of an equijoin on S, we multiply the result of Eq (5.1) by $\min\{|r_1(S)|, |r_2(S)|\}$, i.e. the minimum of the surrogate instance counts for r_1 and r_2.

$$M(r_{TE}, S) = \min\{|r_1(S)|, |r_2(S)|\} \; M(H_1, H_2). \qquad (5.2)$$

When the equijoin is over A, multiply the selectivities of $r_1.A$ by $r_2.A$ (due to the independence assumption), and multiply the result by the total number of intersecting intervals in the two relations, which is calculated as the cartesian product of r_1 and r_2 followed by a restriction based on the intersection of concatenated pairs of intervals, which is equal to the product of the number of surrogate instances in each relation multiplied by the expected number of intersecting tuples for any pair of instance histories; this is how Eq. (5.3) is derived.

$$M(r_{TE}, A) = \frac{1}{|r_1(A)|} \frac{1}{|r_2(A)|} |r_1(S)|\ |r_2(S)|\ M(H_1, H_2). \quad (5.3)$$

Size of Event-Joins

In the case of an event-join, unequal lifespans produce outerjoin tuples in the result. Thus, we have to explicitly consider the original lifespans of the operand relations. One part of the result is the size of a TE-join over S, which was derived in Eq 5.2. There are two outerjoin components-- the starting and ending portions of the lifespans which do not intersect with one another, and the disjoint portions resulting from the existence of nulls in a corresponding time-interval belonging to one of the joining relations. To compute the first component, for each relation, we find the following outerjoin estimates.

$$M(r_{OJ}, LS_{r_i} - LS_{r_j}) = |r_i(S)|\ (LS_{r_i} - LS_{r_{ij}})\ \lambda^h_{r_i}, \text{ for } i = 1,\ j = 2 \text{ and } i = 2,\ j = 1.$$

Procedurally, this is the equivalent of taking the total number of surrogate instances and multiplying it by the expected number of tuples arriving within the time period(s) not covered by an equijoin. Note that these outerjoins take into consideration only the ends of the lifespans, and not the disjoint parts within it. To account for the gaps within the intersecting lifespans, we simply ignore the measures of existence density while deriving the size of the *TE−join* component. The following equation then gives the desired estimate.

$$M(r_{EJ}) = M(r_{OJ}, LS_{r_1} - LS_{r_2}) + M(r_{OJ}, LS_{r_2} - LS_{r_1}) + M(r_{TE}, S). \quad (5.4)$$

6. EXPERIMENTAL RESULTS

In this section, we present the result of a test on the accuracy of one of the measures presented. We limit the test to the estimation of the number of unique surrogate instances found within two randomly generated query intervals, and compare them to the actual count and also conventional

estimates.

The parameter values for the selectivity estimates are derived from statistics compiled by the DBMS. We assume that the following statistics are maintained: starting and ending time (or NOW if it is still active) of the relation's lifespan, the count of surrogate instances, total tuples in the relation and size of the temporal attribute domain. With the exception of the lifespan information, the other statistics are available in conventional DBMS's. The rate of arrival of surrogate instances is estimated by dividing the total number of tuples by the total size of the surrogate domain.

A relation with schema (S, A, T_S, T_E) was generated in the following manner. Instances of each surrogate was created by assuming a uniform distribution of arrivals over the time interval [0, 3,000). A total of 5,000 surrogate instances were produced in this way. The lifespan of each instance was fixed to 400 time units. Tuples were generated within this lifespan, by assuming that each has a valid interval that follows a uniform distribution within [0, 20). In this way, a 181, 373 tuple relation was generated. We randomly selected 20 samples from the relation for query intervals of lengths 25 and 100. For each sample selected, we derived our estimates and measured their error (sample estimate – actual value). We also derived an estimate using conventional methods of estimating restrictions, which assumes equal likelihood that a surrogate instance is present at any given time.

Fig. 4 shows the results, where the third column displays actual values, the fourth the mean error/standard deviation of our estimate, and the fifth the mean error/standard deviation for a conventional estimate. Within the parentheses of the last two columns are the error measures as percentages of the actual. It is very clear from the figures that our estimates are very accurate and significantly better than those obtained by conventional means.

$\|QI\|$	Measure	Actual	Derived Estimates	Conventional Estimates
25	$\bar{x}$	672.2	1.75 (<1%)	4327.9 (>100%)
	$\sigma_{\bar{x}}$	146.0	1.37 (9%)	146.0 (100%)
100	$\bar{x}$	797.0	1.95 (<1%)	4289.4 (>100%)
	$\sigma_{\bar{x}}$	119.9	12.7 (11%)	0.0 (0%)

Figure 4. Results of Test for Number of Surrogate Instances

7. CONCLUSIONS AND FUTURE RESEARCH

We have provided a framework that describes the fundamental issues involved in the statistical modeling of temporal databases and derivation of selectivity estimates. Unlike snapshot relations, temporal relations are much more complex to estimate, yet they also possess more information that may enable more accurate approximations. We introduced our model, in which a relation is characterized by the following: (1) Poisson arrival rate of surrogates instances; (2) Poisson arrival rate of tuples per surrogate instance, and (3) Independent distribution of temporal attribute values. We derived estimates on the size of key unary and binary operations using the model. Two additional factors were taken into consideration in the derivations: (1) The possibility that a surrogate instance's tenure in the relation follows a probability distribution of its own, and (2) The existence of gaps in the history of a surrogate instance, represented by nulls. We also provided results of the accuracy of a selected number of estimates and compared them to estimates derived from conventional methods.

We have not been able to cover all the ramifications and natural extensions of the model to various aspects of modeling and estimation. Nonetheless the model is general enough to be applicable under different scenarios. In the case of time predicates other than intersection, our estimates can be easily modified, since most time predicates are subsets of the intersection relationship; thus only additional restrictions need to be added. With respect to the modeling of asynchronous multi-temporal attribute relations, the primary difference in the modeling approach would be the characterization of the tuple arrival process. We could look at it as a Poisson process with as many types of arrival as there are attributes, where each type is assumed to be independent from the others. Such an approach is called time-sampling from a Poisson process, and it is a known property that each type of event has an independent Poisson process of its own. In this way, we can use decomposition (projection) to determine the independent behavior of each temporal attribute and make the appropriate selectivity estimations. The following are our plans for future research.

- Carry out extensive simulations to test the robustness of the model, and carry out a comparison with extensions of simple estimation techniques.
- Extend the model to include time-dependent temporal attribute distributions. This could prove very

useful in practical applications, since we can probably classify real life temporal behavior into a relatively small set and subsequently develop a distribution-dependent estimation procedure.

• Investigate a more sophisticated surrogate model, which explicitly accounts for the probability of null value generation and also permanent exit from the relation. One technique we are looking at employs Semi-Markov processes for the surrogate event arrivals. We are also looking at non-Poisson models for surrogate arrivals.

REFERENCES

[Ahad et al 89] Ahad, R., Rao, K.V.B., McLeod, D., On Estimating the Cardinality of the Projection of a Database Relation, *ACM Transactions on Database Systems*, 14, 1, pp. 28-40, March 1989.

[Christodoulakis 83] Christodoulakis, S., Estimating Record Selectivities, *Information Systems*, 8, 2, pp. 105-115, 1983.

[Clifford & Croker 87] Clifford, J., Croker, A., The Historical Relational Data Model (HRDM) and Algebra Based on Lifespans, *Proceedings of the International Conference on Data Engineering*, pp. 528-537, February 1987.

[Graefe 87] Graefe, G., Selectivity Estimation Using Moments and Density Functions, Computer Science Technical Report 87-012, Oregon Graduate Center, November 1987.

[Gunadhi & Segev 90] Gunadhi, H., Segev, A., A Framework for Query Optimization in Temporal Databases, *Lecture Notes in Computer Science*, Springer-Verlag, 1990, forthcoming.

[Leung & Mutz 89] Leung, T.Y.C., Muntz, R.R., Query Processing for Temporal Databases, *Proceedings of the International Conference on Data Engineering*, 1990, forthcoming.

[Lynch 88] Lynch, C.A., Selectivity Estimation and Query Optimization in Large Databases with Highly Skewed Distribution of Column Values, *Proceedings of the International Conference on Very Large Data Bases*, pp. 240-251, August 1988.

[Mulakrishna & DeWitt 88] Mulakrishna, M., DeWitt, D.J., Equi-Depth Histograms for Estimating Selectivity Factors for Multi-Dimensional Queries, *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 28-36, 1988.

[Navathe & Ahmed 86] A Temporal Relational Model and a Query Language, UF-CIS Technical Report TR-85-16, University of Florida, April 1986.

[Piatetsky-Shapiro & Connell 84] Piatetsky-Shapiro, G., Connell, C., Accurate Estimation of the Number of Tuples Satisfying a Condition, *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 256-276, June 1984.

[Ross 83] Ross, S.M., *Stochastic Processes*, Wiley, 1983.

[Segev & Shoshani 87] Segev, A., Shoshani, A., Logical Modeling of Temporal Databases, *Proc. of the ACM SIGMOD International Conference on Management of Data*, pp. 454-466, May 1987.

[Segev & Shoshani 88] Segev, A., and Shoshani, A., The Representation of a Temporal Data Model in the Relational Environment, *Lecture Notes in Computer Science*, Vol 339, M. Rafanelli, J.C. Klensin, and P. Svensson (eds.), Springer-Verlag, pp. 39-61, 1988.

[Segev & Gunadhi 89a] Segev, A., Gunadhi, H., Query Processing in Temporal Databases, *Proceedings of the Workshop on Query Optimization*, Portland, pp. 159-164, May 1989.

[Segev & Gunadhi 89b] Segev, A., Gunadhi, H., Event-Join Optimization in Temporal Relational Databases, *Proc. of the International Conference on Very Large Data Bases*, pp. 205-215, August 1989.

[Selinger et al 79] Selinger, P.G., Astrahan, M.M., Chamberlain, D.D., Lorie, R.A., Price, T.G., Access Path Selection in a Relational Database System, *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 23-34, May 1979.

[Snodgrass 87] Snodgrass, R., The Temporal Query Language TQuel, *ACM Transactions on Database Systems*, pp. 247-298, June 1987.

[Snodgrass & Ahn 87] Snodgrass, R., Ahn, I., Performance Analysis of Temporal Queries, TempIS Document No. 17, Department of Computer Science, University of North Carolina, August 1987.

[Yao 77] Yao, S.B., Approximating Block Accesses in Database Organizations, *Communications of the ACM*, 20, 4, pp. 260-261, April 1977.

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
INFORMATION RESOURCES DEPARTMENT
BERKELEY, CALIFORNIA 94720