# Efficient Modeling of Massive Longitudinal Data Using Transition Arrays

Tamraparni Dasu
ML & IR Research
AT&T Labs - Research
Florham Park, NJ 07932

Theodore Johnson
Database Research
AT&T Labs - Research
Florham Park, NJ 07932

## Abstract

We propose a fast, inexpensive technique for summarizing and modeling massive, multidimensional, longitudinal data sets using specialized summaries called transition arrays. We do this in three steps.

1. First, characterize the data points at time $t$ by their relative position in the data space, called a **state**, using a **DataSphere** partitioning proposed in our previous work. It is based on the distance from a center and the direction of maximum variance. (representation step)

2. Next, summarize the aggregate movement in the data space over time using **transition arrays** (summarization step)

3. Finally, **rank** the data points in a given state $i$ at a given time $t$ by the likelihood of transition. (regression step).

Each class in the DataSphere partition of the data space defines a **state**. We wish to jointly estimate the functions $p_{ij}(t)$, the time dependent transition probabilities of a data point moving directly from state $i$ to state $j$ during the interval $[0, t]$. Equivalently, we can consider the **hazard rate** $\alpha_{ij}(t)$, the instantaneous rate of transition from $i$ to $j$ at $t$, given the history of the transitions until $t-$. Furthermore, for the purpose of **ranking** by the likelihood of transition, we would like to customize $\alpha_{ij}(t)$ to an individual $k$ by relating it to the remaining data attributes (known as covariates) of the individual $X_{1k}(t), X_{2k}(t), \ldots, X_{pk}(t)$. We consider additive hazard models that can be fitted using just the summaries.

No distributional assumptions are made since nonparametric methods of estimation are used. Moreover, the estimates used have maximum likelihood properties and other desirable (asymptotic) behavior useful in the computation of error bounds for the estimates. We illustrate the technique with real data.

The technique proposed in this paper uses transition arrays to summarize the aggregate behavior over time and customize the transition rates to individual sample points using additive hazard models. The DataSphere method on the other hand creates summaries called data maps that are representations of cross-sectional or snapshot views of the data. The two methods in conjunction form the basis of a powerful, fast, economical technique that can solve a broad range of analytical and modeling problems.

## 1 Introduction

The recent focus on developing techniques for large data sets has emphasized the issues of dimensionality and scale. The temporal aspect or the time varying nature of massive data sets has not been addressed in any systematic or detailed fashion. In this paper, we propose a fast, efficient method for

- characterizing the movement of points in the data space over time
- summarizing the aggregate movement over time using **transition arrays** and
- customizing transition probabilities to individual sample points by expressing them as functions of the covariates that are not used in constructing the DataSphere partition.

We use the framework developed in our past work ([4], [7]), where we proposed a fast technique for summarizing and analyzing large multidimensional data sets. It entails the construction of a DataSphere around a center, expanding in distance based layers that are further segmented into pyramids based on the direction of maximum variance. Each layer-pyramid combination in the DataSphere partition of the data space corresponds to a **state**. See Figure 1. An observation occupies one and

only one state at any time $t$. As the data attributes associated with an observation change, it moves to different positions in the data space, occupying different states in the DataSphere representation. The sequence of states define the movement of the observation through data space over time. For example, in Figure 1, the point that starts in the $Y^-$ pyramid can be represented by the sequence $(s4, t0), (s4, t1), (s8, t2), (s8, t3)$ and so on.

As an example, consider the problem of characterizing evolving customer behavior. A state could represent the extremity of multidimensional usage (far from average, atypical) and the direction (high usage). A customer moves through different states before stabilizing (or not stabilizing) in a particular state. The sequences of states are very informative by themselves. If a small set of sequences accounts for a large percentage of the customer base, we have an effective method for segmenting customer behavior. Such segmentation is highly desirable for promotions, pricing and other target marketing efforts. In addition to segmenting, we would be interested in the probability of transition to a high usage state, given the current state of a customer and the time spent in it. Furthermore, we would like to quantify the effect of the individual characteristics of the customer on the baseline transition probabilities. For instance, how does the baseline transition probability change for a customer who is female, highly educated and subscribing to a service at $X per month?

Modeling the effects of other variables on the transition probabilities of moving from one state to another has two major implications.

- First, the transition probabilities can be customized using an individual data point's attributes. This is very important for very large data sets where there could be thousands of observations in a given state at a given time. The ability to express the transition probabilities in terms of an individual's specific covariates enables us differentiate among them instead of assigning the same transition probability to all of them.

- Second, if the model fits well, just the summaries of the significant covariates need to be stored, resulting in a considerable data reduction.

In Section 2, we briefly review the DataSphere method. In Section 3 , we introduce the terminology of counting processes and functions that are relevant to the problem at hand. Section 4 illustrates the technique using a real data example. Section 5 contains the conclusions and evaluation of the method. Finally, in Section 6, we outline the directions of future research.

# 2 DataSphere Framework

The DataSphere representation is based on two components: (a) distance from a center (such as the mean or the componentwise trimmed mean if robustness is an issue) and (b) the directional pyramid. The pyramid determines the direction of maximum variation. See [2] for more on pyramids. Each layer-pyramid combination defines a "state". The number of states can be controlled at will by collapsing the layers and pyramids. This poses no problems since all summaries are aggregable. The continuation of a point in the same state corresponds to a period of "similar" behavior. Transition or movement from one state to another represents a non-trivial change in attributes, indicating altered behavior. We have thus summarized a multidimensional time series of attributes by a two dimensional sequence of discrete states and the corresponding times of entry. Next, *transition arrays* containing summaries such as counts, sums, sums of squares and sums of cross products of variables are constructed for

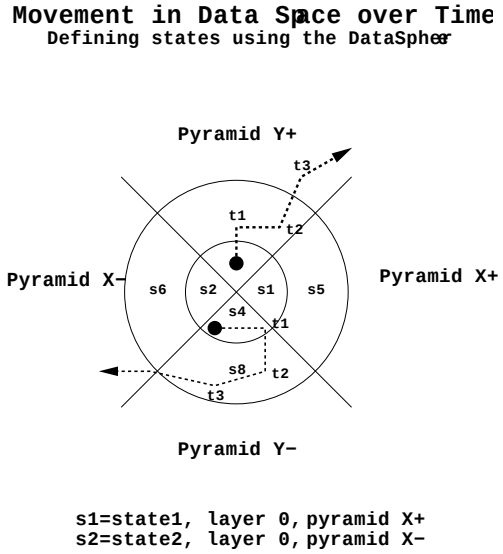


Figure 1: Movement in Data Space using DataSphere Representation.

1. the whole data set, grouped by the state at time $t^-$ (risk set grouping).

2. the subset that has made a transition at $t$ grouped by the transition type e.g. from $i$ to $j$ (transition grouping).

Transition arrays achieve considerable data reduction while retaining sufficiently granular summaries to enable the computation of statistical tests and models. Moreover, the additional storage and computation is not very

expensive. One needs to store (a) the previous state for every customer and (b) the two tables of profiles corresponding to the risk set and transition type groupings of the data. Extra computation is in the form of matching the current and previous states and computing the transition arrays, which can be accomplished during the second pass of the DataSphere construction.

# 3 Background

We introduce the terminology used in developing the longitudinal models.

## 3.1 Counting Processes

Let $N_{ij}(t)$ be the number of direct transitions from $i$ to $j$ observed in the time interval $[0,t]$. $N_{ij}(t)$ is a *counting process* that counts the number of transitions from $i$ to $j$ in $[0,t]$. Let $Y_i(t)$ be the number of points in state $i$ at time $t$. This is called the *risk set* at time $t$. If there is a transition at time $t$, the risk set corresponding to state $i$ consists of all the points that are in that state at time $t-$. Associated with the counting process is an *intensity function* $\lambda_{ij}(t)$ that measures the intensity of transitions from state $i$ to $j$ at time $t$. It is given by

$$\lambda_{ij}(t) = \alpha_{ij}(t) Y_i(t) \tag{1}$$

where $\alpha_{ij}(t)$ is the *hazard function*, the instantaneous rate of transition from $i$ to $j$ at $t$. Let

$$A_{ij}(t) = \int_0^t a_{ij}(u)du \tag{2}$$

be the cumulative hazard function. If the time intervals between the transitions are continuously distributed, then the density function, the cumulative distribution function, the hazard function and the integrated hazard function are all equivalent forms of specifying the inter event interval distribution. See [3] for details. Most of the concepts and estimators discussed in this paper carry over to a discrete time scale as well.

Under certain conditions, the following estimate called the Nelson-Aalen estimator, is the nonparametric maximum likelihood estimator of $A_{ij}(t)$.

$$\hat{A}_{ij}(t) = \sum_{t_k} \frac{\Delta N_{ij}(t)}{Y_i(t)} \tag{3}$$

where $t_k$ are the transition times in the interval $[0,t]$.

## 3.2 Estimation of the Transition Matrix

Knowing the integrated hazard function $A_{ij}(t)$, under certain conditions, the matrix of transition probabilities can be constructed using the relationship

$$P(0,t) = \prod (I + \Delta A) \tag{4}$$

where the product is taken over all transitions in the interval $[0,t]$ in left to right order, $\mathbf{I}$ is the identity matrix and $\mathbf{A}$ is the matrix that contains the increments in the cumulative hazard function of transitions. Note that the diagonal elements of A are defined by:

$$A_{ii}(t) = -\sum_{i \neq j} A_{ij}(t) \tag{5}$$

See [1] for details and further references. Error bounds, other estimates and optimality properties will be discussed in a later paper due to space constraints. Here, we use the simplest possible estimates to illustrate the longitudinal technique proposed.

## 3.3 Additive Hazards Model

We now wish to rank the data points that are in a given state at a given time by the probability of transition. In order to do this, we express the the time dependent transition probabilities as a function of the time series of data attributes of a point. Since the probabilities can be derived from the hazard function, we consider the problem of modeling the hazard function as a function of the data attributes. However, we would like to constrain ourselves to models that can be computed from aggregates. This is in keeping with the spirit of *data maps* which are aggregable summaries stored in the DataSphere representation of massive data sets. The constraint is not unduly restrictive, as we will see later. Additive hazard models are ideal candidates for aggregate based modeling. See [5] for related discussion.

To simplify notation, let $\alpha(t)$ represent $\alpha_{ij}(t)$. Let $\alpha_l(t)$ be the hazard of changing from state $i$ to state $j$ for the $l^{th}$ individual at time $t$. The additive model states that the hazard function is the sum of the contributions of the individual attributes. That is,

$$\alpha_l(t) = \beta_0(t) + \beta_1(t) Z_{l1}(t) + \ldots + \beta_d(t) Z_{ld}(t) \tag{6}$$

where $d$ is the number of attributes (or dimensions) and $Z_{lm}(t)$ represents the value of the $m^{th}$ attribute of the $l^{th}$ individual at time $t$. Also, the attributes used in the hazard model should not overlap with the attributes used to create the DataSphere, in order to eliminate an implicit relationship between the state and the attributes

used to predict the movements among the states. We wish to estimate the $\beta(t)$s based on the data. Note that the $\beta(t)$s are the same for all points and that they vary with time. It is the covariates $Z(t)$s that are different for every data point, resulting in a different hazard value. We will start by estimating the integrals of the $\beta_k(t)$s

$$B_k(t) = \int_0^t \beta_k(u)du \tag{7}$$

and subsequently derive the $\beta(t)$s from them. Let $\mathbf{B(t)}$ be the vector of $B_k(t), k = 1, \ldots, d$. Further, let the $i^{th}$ row of the attribute matrix $X$ at time $t$ be given by

$$X_i(t) = I_i(t)(1, Z_{i1}(t), \ldots, Z_{id}(t)) \tag{8}$$

where $I_i(t)$ is the indicator function that the point $i$ is in the appropriate risk set at time $t$. In other words, for computing the hazard function of transition from $h$ to $k$ at $t$, only those points that are in state $h$ at time $t$ contribute to the estimation of the hazard at $t$. The index $i$ corresponding to the rows of $X(t)$ ranges from 1 to $n$, the number of data points at time $t$. Then, an estimate of $B(t)$ is given by

$$\hat{\mathbf{B}}(\mathbf{t}) = \sum_{t_k} \left(X'(t)X(t)\right)^{-1} X'(t)\Delta N(t) \tag{9}$$

where $\Delta N(t)$ is the vector of changes in the counting process for each of the $n$ individual data points. Optimizing these estimates using kernel functions is deferred to a later paper. The $\beta(t)$s can now be computed. See [1] for an excellent discussion. Finally the mean squared error process associated with $\hat{B(t)}$ is estimated by

$$\hat{\Sigma(t)} = \sum_{t_k} \left(X'(t)X(t)\right)^{-1} \left(\Delta H(u)\right) \left(X'(t)X(t)\right)^{-1} \tag{10}$$

where $\Delta H(u)$ is the matrix of cross products of attributes for the subset of points that had a transition at $t$. See [1] for asymptotic theory and other details.

# 4 Data Example

We illustrate the technique using data from an AT&T data warehouse. The data consist of over 31,000 distinct customers of an AT&T service, each observed at nine different points in time (approximately 280,000 data records in all). At each point in time, six variables are measured, each related to a different type of usage. The first four are used to create the DataSphere representation. The last two are used as covariates in the additive hazard model for customizing the transition probabilities. In other words, the last two variables are used to predict the movement in the space of the first two variables. The partition chosen is very coarse, to make the illustration of the technique simple. There are only two layers, inner(0) and outer(1). Since four variables are used to create the DataSphere, there are 8 possible pyramids, two corresponding to each variable. We collapse all the positive pyramids into a positive "quadrant" and all the negative ones into a negative "quadrant". Hence there are four possible states that a datapoint can be in ; **0+ (Inner layer, positive quadrant), 0- (inner layer, negative quadrant), 1+ (outer layer, positive quadrant) and 1- (outer layer, negative quadrant).**

## 4.1 Distribution of the States

Initially, states **0+** and **1+** account for around 36% and 32% each of the total base of 31,000. **0-** accounts for 20% and **1-** for the remaining 12%. That is, there is a predominance of high usage points. Over the nine month period, **1-**, the low usage class, gains 5% of the total base of 31,000 data points, mainly from the **0+** class. The other classes remain more or less unchanged. We will analyze this phenomenon further using transition arrays.

## 4.2 Transition Probabilities

The time dependent transition matrix of probabilities was estimated using formulas 3 and 4. We plotted the estimated probabilities of transition from state **0-** in Figure 2. X-axis is time elapsed, Y-axis is the probability estimate. The solid lines correspond to the above average or positive quadrant states **0+** and **1+**. The dashed lines correspond to the below average or negative quadrant states **0-** and **1-**. For example, the probability of making a direct transition from **0-** to **1-** in 6 time units is around 0.2. From the plots of transitions from each of the four initial states (remaining 3 plots not shown here) we conclude that:

1. The process becomes stationary after 3 time units have elapsed.

2. The transition probabilities depend significantly on the initial state.

3. Transition to neighboring states are much more likely than dramatic leaps. In other words, change in behavior as defined by the attributes is gradual in most cases.

4. There is a surprisingly high likelihood of changing states.

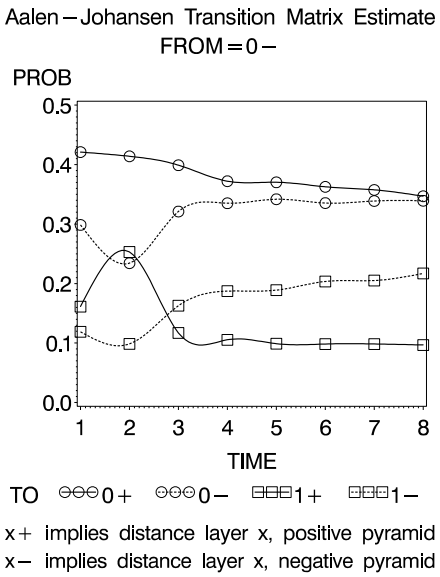


Figure 2: Transition Probabilities - State **0-**.

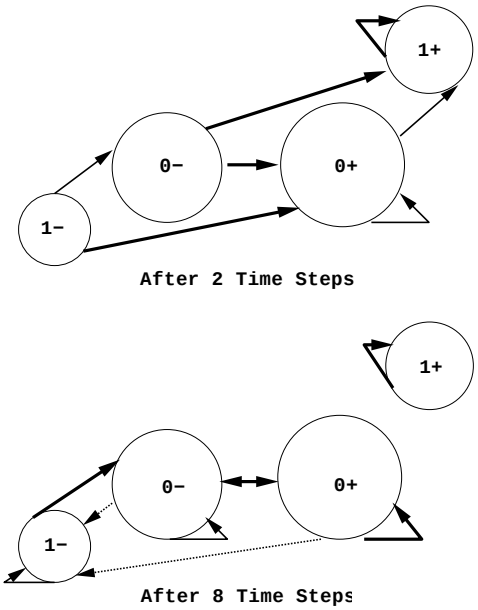


Figure 3: Transition Probabilities - Initial and Final.

## 4.3 Overall Process Behavior

The overall behavior of the process is summarized in (Figure 3). Each circle represents a state, the size being an indicator of the cardinality of the state. The arrows indicate the dominant transitions. The thickness of the arrows is proportional to the probability. After the passage of two time units, there is a tendency to move to higher usage states. But after 8 time steps, the direction of flows seems to be reversed. Arrows now point downwards from **0+** to **0-** and **1-**. This in fact contributes to the inflation of **1-** by 5% of the total base over nine units of time, as we had noticed by comparing the initial and final frequency distributions of the states. In addition, note that the very high usage class **1+** is quite isolated. It means that high users tend to declare themselves early on and remain high users for the rest of their existence.

## 4.4 Ranking - Additive Hazard Model

We fit an additive hazards model based on Equation 6, to rank all the points in a given state at a given time, using the methodology in Section 3. We stratify by the transition type. Two covariates were used, each related to two distinct types of usage. We estimated the corresponding coefficient functions $\beta_1(t)$ and $\beta_2(t)$. The result for the transition from **0-** to **0+** are shown in Figure 4. The X-axis represents the time elapsed. The two step functions named NA and RE are the cumulative hazard functions using a Nelson-Aalen empirical estimator and the regression model evaluated at the mean values of the two covariates, respectively. They represent the cumulative hazard of making the transition from **0-** to **0+** over the interval $[0, t]$. The two plots at the bottom are plots of $\beta_1(t)$ and $\beta_2(t)$, (not their integrals $B(t)$s). From Figure 4 we infer the following:

1. The empirical estimate NA and the regression estimate RE are close, therefore the additive model is a good choice for modeling the **0-** to **0+** type of transitions.
2. Larger values of type 1 covariate increase the hazard of making a transition to **0+**
3. Larger values of type 2 covariate decrease the hazard of making a transition to **0+**.

Using these functions, we can now associate distinct probabilities of transition with each data point in state $i$ at time $t$. Furthermore, we can rank them by the likelihood of transition.

## 4.5 Summary of Findings

In this example of the AT&T warehouse data, after an initial period of flux, the dataset exhibits a stable pattern of movement in the dataspace that can be summarized using the transition matrix of probabilities. After an initial trend of increased usage, there are signs of declining usage among the typical points. The data points that are above average at the outset continue to generate high usage. The process becomes stationary after 3 time steps. Observe that:

- We can use the models developed to predict the behavior of new data points based on the values of their attributes used in the hazard model.

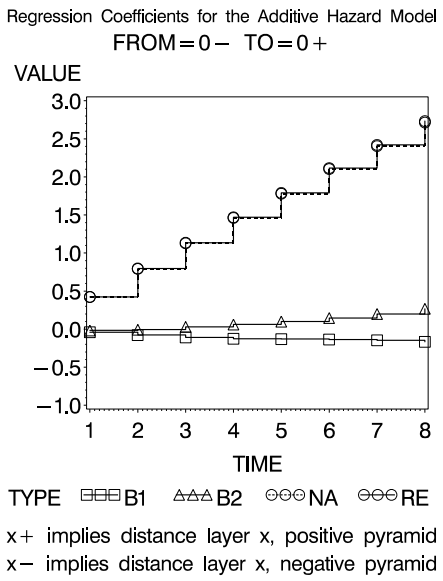


Figure 4: Additive Hazard Model for Transition **0-** to **0+**.

- If the two covariates can be controlled, then the data points could be moved in a desired direction. An example would be to encourage covariate 1 usage through incentives and discourage covariate 2 usage through penalties, with a view to moving data points from **0-** to **0+**, the higher usage state.
- Beyond three time steps, all we need to know about a point are its current state, the time spent in it and the values of the covariates, in order to predict its future behavior.

# 5 Conclusion

We have found that the longitudinal dataset can be summarized effectively using the DataSphere representation and transition arrays. Note:

- We have reduced a very high dimensional problem to that in two dimensions, for representing and summarizing aggregate movements.
- The summaries are enough to estimate transition probabilities and customize them using other. Rigorous predictive models can be built using simple order moments.
- Furthermore, due to the aggregable nature of summaries, the number of states can be controlled by collapsing layers and pyramids.

# 6 Future Research

Further research has three main thrusts. First, to integrate the technique proposed in this paper into a database environment to make it a standard analytical tool for data mining. Second, to extend the statistical analysis proposed in this paper, by

- including error bounds and significance tests for the $\beta$s.
- adapting existing multiplicative hazards models such as Cox's regression model for summary based analysis
- developing other models within the DataSphere framework for prediction and forecasting.

Finally, to develop more efficient partitioning schemes. See [6].

# References

[1] Andersen, P. K., Borgan, O., Gill, R. D. and Keiding, N. (1992), *Statistical Models Based on Counting Processes*, Springer.

[2] Berchtold, S., Bohm, C. and Kriegel, H. The Pyramid_Tree: Breaking the Curse of Dimensionality. In ACM SIGMOD 1998.

[3] Cox, D. R. and Oakes, D. (1984), *Analysis of Survival Data*, Chapman and Hall.

[4] Dasu, T. and Johnson, T. An Efficient Method for Representing, Analyzing and Visualizing Massive, High Dimensional Data. In Interface 1997.

[5] Dasu, T. and Johnson, T. Piecewise Linear Regression for Massive Data through DataSpheres. In Joint Statistical Meetings 1998.

[6] Dasu, T., Johnson, T. and Jagdish, H. V. Scalable Partitioning of High Dimensional Space through HyperDataSpheres. In preparation.

[7] Johnson, T. and Dasu, T. Comparing Massive High Dimensional Data Sets. In KDD 1998.