Sparse Matrices and RDBMS for Statistical Testing of Hypotheses in Astrology

Prithwis Mukerjee[1], PhD

Abstract

This paper introduces traditional statistical rigour into the testing of hypotheses in Indian astrology. It describes a framework where horoscope charts are defined in terms of sparse matrices and shows how the same can be stored in a relational database and retrieved with SQL. The paper concludes by introducing a specific software implementation of this framework that is available in the public domain.

Introduction

Astrology has always been an integral part of the grand intellectual traditions of India but historical and cultural factors have kept it at an arm's length from the science and technology community today. However, widespread public interest in this subject has opened the floodgates to a large number of frauds, fakes, charlatans plus a few honest astrologers who have made this into a thriving but sometimes dishonest service business. This situation can be rectified if computer technology and modern mathematical methods can be used but any attempts to introduce this subject into the formal curriculum is met with indifference and hostility that of late has even taken on a political colour.

A little knowledge is a dangerous thing – it traps a person in a small zone of comfort. This is particularly true for many of our "scientific fundamentalists" – students of science and engineering who are so uncomfortable with anything outside this zone that they will criticise any attempt to go beyond as being *beneath their dignity* whereas in reality it is *beyond their ability* !

Astrology is a subject that lies outside this traditional zone of comfort and a typical scientist finds that it is *beyond his ability* to tackle the subject in a manner that a scientist is accustomed or expected to. Even if we leave aside the leap of imagination that is necessary to explain *why* astronomical bodies can influence personal events, they are still stuck with three major roadblocks : (i) The inability to frame hypotheses that can decide whether there is at all any influence. (ii) The inability to devise an experiment to test the hypothesis and (iii) The non-availability of data in a format that can be used in such an experiment.

This paper addresses these three roadblocks. It shows how hypotheses can be set up. It introduces a software architecture that can be used to store, retrieve data that can be used to design an experiment and describes an actual software built on spreadsheet, PHP and RDBMS technology that does the same. It describes a scheme that can be used gather data in a collaborative ("parallel") manner to support the experiment that in turn can be used to accept or reject the hypotheses.

We conclude that this approach is adequate if astrology is viewed as an applied science and there is *no need for* – though it will certainly be good to have – theories that explain the mechanics of *why* astronomical bodies influence human lives.

This paper demonstrates that it is well within the ability of any scientist to carry out research into astrology with a level of rigour that is comparable to that found in any other area of social sciences.

Survey of Literature

Astrology has two aspects : computational and interpretative. Computational astrology is a very structured and algorithmic process that is used to

1 Dr Prithwis Mukerjee, formerly with the Praxis Business School, Calcutta, is currently Professor at the Vinod Gupta School of Management, IIT Kharagpur. EMAIL : *prithwis@yantrajaal.com* PORTAL : www.yantrajaal.com

calculate positions, aspects, conjuncts, *dashas* and there are many books that explain the process. Majumder [1] in his book on applied astrology gives a fairly detailed account of the process but Braha's book, [2] written by a native English speaker for a western audience may be easier for scientists and engineers to read – simply because it is written with the rigour of a semi-scientific text. Similar rigour is visible in Rao's introductory text [3] that not only explains the computational principles but lays down a very structured approach, based on Positions, Aspects and Conjuncts (PAC) to address the next – and more difficult – aspect of astrology : predictions.

The basic tenets of astrology can be defined in terms of the level of correlation between patterns found in the horoscope chart and corresponding patterns of events in the life of the person. Given the volume and nature of data, standard statistical techniques as is given text books like Levin and Rubin [4] may not prove to be adequate. Instead we look at and adopt data mining techniques like association rules mining that are described by Gupta [5].

But before these data mining techniques can be used, the data itself – that is the charts and the catalogue of events – must be transformed into a form that can be used in a statistics based approach. Specifically we should be able to store the data in a manner that is amenable to speedy and selective retrieval. Storing charts as images is possible but retrieval on the basis of specific characteristics is very difficult with images. Sparse matrices [6,7] offer a way to represent horoscope charts and Beckmann [8] shows one way of how such matrices can be stored in an RDBMs.

Finally any statistical approach needs access to a significantly large volume of data and the Web 2.0 approach – as first described by Knorr [9] and then popularised by O'Reilly [10] – can be used. Web 2.0 is a post "dotcom" phenomenon that describes a class of software platforms that have four main characteristics :

- User generated content
- Network of trust
- Rich Media
- Being in perpetual beta

ORKUT, FACEBOOK, LINKEDIN, WIKIPEDIA, GOOGLE SEARCH, FLICKR, SECONDLIFE are some of the more visible and prominent members of this class.

The importance of Web 2.0 in the current social and economic context is explored further by Mukerjee [11] and Pauniker [12].

Finally we look at the ways to implement these ideas using a software package. A quick search on Google reveals about 17.6 million hits for "astrology software" and about 513,000 hits if we narrow the search to "indian astrology software" -- so there is no dearth of software on astrology. However a closer look reveals that most these software packages are either meant for casting horoscopes or for making predictions based on logic that the software developer *believes in*. None of these software packages address the challenges associated with statistical validation of astrological hypotheses that is the focus of the current paper.

Approach to the problem

Astrology deals with charts and events in the life of the person. So the chart of an individual k can be defined as $C_K = (P_K, E_K)$ where P_K is a *set* of astrological patterns and E_K is a *set* of life events.

We will next focus on a class of hypotheses that can be defined as :

H_0 : <Astrological Pattern X_I> is associated with <Life Event Y_J>

This hypothesis can be accepted if we can identify a significant number of charts that satisfy the three condition : $C_K = (P_K, E_K)$ *and* X_I belongs to P_K *and* Y_J belongs to E_K

To test this hypothesis, we need to have a "database" of *patterns,* DB_P, and *events,* DB_E, built with data from actual individuals and then search within this "database" for cases where pattern X_I and event Y_J occur together. This is analogous to the data mining technique of *association rules mining or "market basket analysis"* where the aim is to establish, for example, that customers who buy bread (item X) will

also buy milk (item Y).

Market basket analysis is based on the concept of *support* and c*onfidence*. Support is the *joint probability* of X and Y and calculated as the ratio (or fraction) of the number of transactions where the two occur together to the total number of transactions. Confidence is the *conditional probability* of Y given X and is calculated as the ratio (or fraction) of transactions where X and Y occur together to the number of transactions where only X occurs. To establish a rule that X_I implies Y_J, or that the X_I and Y_J occur together, we need to prove that the values of support and confidence for the pair (X_I,Y_J) are higher than a specified threshold. The a-Priori algorithm [5, pg 53] is one of the many such algorithms available to identify and accept rules that meet the minimum acceptable values of support and confidence. The same approach can be used to accept the null hypothesis H_0 as defined above.

But before we can apply these mathematical techniques we need to define the nature of the data and the structure of the database so that it is possible calculate the the ratios – for support and confidence – by identifying and retrieving horoscope charts that contain these patterns and events.

There are many kinds of "databases" but we choose an RDBMS because (a) it is the natural way to store and retrieve sets (b) it is so widely used that there is no ambiguity in its definition and (c) the software to implement it is widely available.

The next challenge is to create a relational data model DB_P and DB_E to store the two kinds of sets P_K and E_K that represent patterns and events.

Relational Data Model for Patterns (P_K)

A pattern in a horoscope chart consists 10 "planets"[2] that can reside in any of the 12 zodiac signs. This can lead to 12^{10} (or 61,917,364,224) possible patterns. Dealing with so many patterns is both difficult and pointless because with only about 6 billion people on the planet, the support probability for any of these patterns will be far lower than any acceptable value. There will not be enough people who have the same pattern for us to draw any meaningful conclusion. Hence we need to look for sub-patterns that have a likelihood of being repeated across multiple individuals. What are these sub-patterns and how can we define a relational data model to store sub-patterns ?

A horoscope chart or *pattern* consists of 10 numbers that specify the longitudes of the 10 planets plus the binary information on whether 5 of these planets are retrograde or not. If we use K N Rao's technique, then this information can be used to define these sub-patterns in terms of positions, aspects and conjuncts (PAC).

Positions of planets can be defined in terms of the absolute position – or house, that is one of the 12 zodiac sign like Aries, Taurus, ... Pisces – or in terms of a relative position from the ascendant (also known as *lagna*) – the first *bhava*, the second *bhava*, ... the twelfth *bhava*.

Planets in some positions are known to be exalted, debilitated, could be located in their own house, in a friendly house or in a hostile house and in each case their "influence" is enhanced or reduced. Positions also specify whether a planet is benefic or malefic.

Aspects are defined when planets are positioned 180^0 from another or, as in the case of planets like Mars, when they are within some specified angles of each other.

Conjuncts are defined when two planets reside in the same house or within some specified angle from each other.

For the purpose of the data model, we shall ignore the subjective or qualitative aspects of terms like benefic, friendly etc., and reduce this information to a set of ***pure binary values***. For example :

- Is planet P_A exalted or not ? Is it debilitated or not ? Is it benefic or not ? ... and so on ...

2 Our objects of interest are not all planets as defined in astronomy. Astrology deals with five planets, one star, one satellite, two points in space defined by the intersection of the ecliptic and the elliptic and the horizon – all of which are referred to as "planets" for convenience. We continue with this convention.

- Is planet P_A aspected by planet P_B or not ?
- Is planet P_A conjunct with planet P_B or not ?
- Is planet P_A resident in a specific house ? In a specific *bhava* ?

To complicate matters further, some of the planets are defined as lords of certain houses and so all these questions can be repeated ...

- Is lord L_A exalted or not ? Debilitated or not ? Is benefic or not ? ... and so on ...
- Is lord L_A aspected by planet P_B or not ?
- Is lord L_A conjuncted by lord L_B or not ?
- Is lord L_A resident in a specific house ? In a specific *bhava* ?

To complicate matters even further, some of these questions can be asked for the *bhavas* as well.

*Each binary value, or a combination, constitutes a specific **sub-pattern** that is a potential candidate for a hypothesis*. The set of patterns P_K in the horoscope chart $C_K = (P_K, E_K)$ can now be defined in terms of this set of sub-patterns.

This set of sub-patterns can in turn be defined in terms of a matrix – or rather a sparse matrix. Illustration 1 shows a *part* of this matrix, the rest of which can be seen in the Parashar21 software that will be described later.

We know that the entire set of sub-patterns (or secondary patterns) can be algorithmically derived from the primary or principal pattern consisting of the longitudes of the 10 planets. However it is important and necessary, even at the cost of redundancy, to store these secondary patterns, separately and independently, because subsequent queries, all of which are SQL queries, will be made against C_K the set defined as the collection of these sub-patterns.

This approach is analogous to that taken in the design of a data warehouse where denormalised, redundant, aggregate data is stored in addition to primary atomic data – in violation of the principles of third normal form, so as to ensure efficiency in the retrieval process.

Net-net we have a sparse matrix that represents binary information like :

- Lagna is aspected by Mars ? Y
- Fifth Lord is debilitated ? N
- and so on and on and on ...

The data model – defined in terms of rows and columns – to store a sparse matrix can now be defined as follows :

- We define the rows and columns of the sparse matrix in terms of the requirements of the sub-patterns. For example
  - COLUMNS go like : Lagna, Sun, Moon ... , First Lord, Second Lord, .. First Bhava, Second Bhava and so on
  - ROWS go like : Aspected by Sun, Aspected by Moon .., Conjunct with Sun, Conjunct with Jupiter ... and so on
  - The complete set of ROWS and COLUMNS that we believe are necessary to store Positions / Aspects / Conjuncts is shown in the Parashar21 software that is available in the public domain.
- We specify a unique code for each row and column
  - Columns go like X00, X11, X12 ...while Rows go like Y00, Y01,..
- Now in a particular chart, if a specific sub-pattern occurs (that is, a boolean Yes) then all we need to store is to store the row and column that correspond to this sub-pattern.

If we have a relational table that is created as :

SQL >> create table pac2 (chartid char(12) not null, kause char(6) not null);

then the sub-pattern “Seventh Lord is aspected by Fifth Lord” will be stored by :

SQL >> insert into pac2 values ('pmuk13991000','X27Y35');

In this implementation the relational **table** *pac2* represents the database DB_P against which we can execute a query that, for example, identifies charts that have “Lagna aspected by Mars” AND “Venus conjunct with Mars” through this SQL command :

SQL >> select chartid from pac2 where kause = 'X14Y55' and chartid in (select chartid from pac2 where kause = 'X00Y25');

While this data model can be supported by any RDBMS, the complex nature of the SQL statements – where human readable English text needs to be converted into code – would require the presence of a front end tool[3].

This data model addresses the first of the two requirements, namely storage and retrieval of natal charts. The second part, namely, storage and retrieval of life events is shown next.

Relational Data Model for Events (E_K)

Events are easier to address because the universe of possible events is not very large. We have identified about a hundred odd events but even if another group of astrologers were to do so the total number cannot be more than twice that number. We classify events into four categories :

- Type A : which is a binary event. For example, Person is an Engineer or NOT. There are no values or parameters associated with such an event
- Type B : these are events that need one parameter. For example, Number of siblings, or age at which person entered first job
- Type C : there is a small class of events that need two parameters, for example, X job changes in Y years.
- Type D : these are events that are associated with a particular date, for example a bereavement or marriage. Associated with a type D event is the corresponding Dasha, Antardasha and Pratyantar Dasha which of course can be easily calculated.

The event list E_K of the chart $C_K = (P_K, E_K)$ can now be defined as a set of these specific events.

In this analysis we have used the term “event” rather generously to include not just hard events but also characteristics of the life of a person whose chart has been stored. This is necessary because we need to define and test hypotheses that correlate sub-patterns in charts to “events” in a person's life.

Illustration 2 shows how one method of defining the set E_K in terms of predefined events. It shows the four kinds of events – including sub-events and even sub-sub-events – and associates a unique event code at the granularity of each sub-sub-event. When an event needs to be included in E_K, a field is set from N to Y and this generates additional data – like *dasha* – and also the SQL needed to store this event.

The Parashar21 software from which this screen shot has been taken is smart enough to determine the type of event from the structure of the event code and highlight the fields that need to be filled in. For example a type D event needs a date whereas a type A event needs nothing more.

Once defined in these terms, the storage and retrieval of event data using only SQL is very straight forward and is very similar to the approach used for charts.

We need two tables that are created as follows:

SQL >> create table eventcodes (eCode char(4) not null, Event varchar(20), SubEvent varchar(20), SubSubEvent varchar(20));

SQL >> create table events (chartid char(12) not null, eCode char(4) not null, p1 int, p2 int, Dasha char(3), Antar char(3),Pratyan char(3));

3 The SmartChart Excel workbook of Parashar21 has macros that generate these SQL statements automatically. However any other application can be used as well.

Next an event like “having X siblings” is defined with a code like “910B” in the eventcodes table :

```
SQL >> insert into eventcodes
values("910B","Siblings","X siblings","");
```

similarly a “Bereavement of Father” is defined as :

```
insert into eventcodes
values("A10D","Bereavement","Father","");
```

Finally events corresponding to a persons having 3 siblings will be be stored as :

```
SQL >> insert into events values
("pmuk15130000","910B",3,0,"-","-","-");
```

or that the person suffered a bereavement at age 47 when *dashas* were of jupiter-venus-rahu will be stored with a statement like :

```
SQL >> insert into events values
("pmuk15130000","A10D",0,47,"jup","ven","rah");
```

In this implementation the two relational **tables** *events* and *eventcodes* represents the database DB_E from can be retrieved, for example, through this fragment of SQL code[4] embedded in a PHP program that joins the two tables and formats it for human readability

```
$sql40 = "SELECT chartid, events.ecode, event,
subevent, subsubevent,";
$sql41 = "if((mid( events.ecode, 4, 1 )='B')||
(mid( events.ecode, 4, 1 )='C'),'X=',' ') , if(p1
=0,'',p1),";
$sql42 = "if(mid( events.ecode, 4, 1 )
='C','Y=',if(mid( events.ecode, 4, 1 )='D','Age=',' ')) ,
if(p2=0,' ',p2),";
$sql43 = "dasha, antar, pratyan FROM events,
eventcodes where events.ecode =
eventcodes.ecode";
$sql44 = $sql40.$sql41.$sql42.$sql43;
$sql45 = " order by chartid, events.ecode";
$sqlquery4 = $sql44." and chartid in (" . $members .
")".$sql45.";";
```

4 These code fragments taken from the Parashar21 software are used only as examples. There could be other – and better – ways of implementing this model.

Where the variable *$members* is either a list of charts entered manually or generated automatically through another (nested) SQL like

```
SQL >> select distinct(chartid) from events where
ecode ='141A' and chartid in ( select distinct(chartid)
from events where ecode ='530A' and chartid in
( select distinct(chartid) from events where ecode
='Z40D' ))
```

.

This data model addresses the second of the two requirements, namely storage and retrieval of events.

Building the Database : A Web 2.0 approach

Our model consists of two databases DB_P and DB_E defined in terms of RDBMS tables that can store horoscope charts $C_K = (P_K, E_K)$. This data can now be used to test hypotheses like :

H_0 : <Astrological Pattern X_I> is associated with <Life Event Y_J>

by observing the number of occurrences of chart K that satisfy the three conditions : $C_K = (P_K, E_K)$ *and* X_I belongs to P_K *and* Y_J belongs to E_K and applying the standard techniques of association rules mining.

But this needs a significant volume of data if the results are to be significant. So to build and test hypotheses using this model, an astrologer would need access to thousands of charts and their corresponding events and this is where the concept of Web 2.0 can be of great use.

Web 2.0 platforms get users to contribute content and facilitate the formation of communities whose members trust each other. For example, every article in WIKIPEDIA is contributed by one or more users and then vetted by members of the community for authenticity – and the power of this model is so immense that it has put both the venerable Encyclopaedia Britannica as well as the tech-savvy Microsoft Encarta out of business! Moreover a Web 2.0 platform handles rich media – which in most cases means images and videos – and the data that it contains is forever being added to as users contribute new content.

We strongly believe that the same Web 2.0 approach can be used successfully in astrology research because of the following reasons :

- User Generate Content : As more and more astrologers contribute charts the pool of charts available to each individual astrologer increases. This will allow statistical tests to be conducted with increasing confidence.

- Network of Trust : Astrologers who contribute charts that are deemed good and useful will be recognised by other members of the community and relationships built upon reputation will emerge as in the case of "traditional" social networks.

- Rich Media : The charts $C_K = (P_K, E_K)$ as implemented in this model constitute a new kind of "rich" media that contains a significant quantity of information.

- Perpetual Beta : As charts keep getting added to the database, the value of the platform will keep on increasing and this in turn will – hopefully – feed a virtuous cycle.

Parashar21 : An experimental implementation

Now that we have a collaborative model to gather vast amounts of data and a data model to store and retrieve the same, we would need to build a software to actually implement this research framework.

This has been done on an experimental basis by building the Parashar21 community network that is available for public access at the website located at *http://parashar21.yantrajaal.com* .

This is an Orkut style network built on the free Ning platform (*http://www.ning.com*) that allows members to register – at no cost – and then access the Parashar21 software that has been developed and distributed under the GNU Public License of the Open Software Foundation.

The Parashar21 software consists of an MS-Excel spreadsheet – the SmartChartBeta – front end and a back end MySQL database that is accessible through PHP programs. The MS-Excel front end uses the Swiss Ephemeris software, which is also free for non-commercial use, to calculate planetary positions and then uses an ingenious set of formulae to convert this this ephemeris data into a sparse matrix that represents P_K. Similarly life events are manually recorded and transformed into the matrix format through the SmartChartBeta spreadsheet.

The SmartChartBeta spreadsheet can also be ported to the open source OpenOffice platform but MS-Excel was chosen as the primary platform because it is more likely to be available with people who are not open source enthusiasts or are not professional programmers.

Similarly, the back end database can be migrated to a commercial RDMS like Oracle or DB2 if there are any doubts about the reliability or scalability of MySQL.

Parashar21 is not essential for this model but demonstrates that the model described in this paper can be easily implemented with technology available off-the-shelf today.

Conclusions and Future Research Directions

This paper has demonstrated a scheme to (a) collect data on astrological charts and (b) to store and retrieve this data in a manner that can be used for statistical testing of astrological hypotheses.

It presents a model that will allow the tenets of astrology to be tested to the same level of rigour that is applied to any of the behavioural sciences like psychology, anthropology or sociology. However it does not attempt to address the more fundamental question of how or why an astronomical body can influence the course of events in the life of person.

But this is no real handicap. When Isaac Newton introduced the law of gravitation, he never bothered to explain ***why*** massive bodies attract each other – he restricted himself to a description of the nature of the attraction in the form of the inverse square law. The tenets of astrology that we seek to prove or disprove through this model are similar in nature. In the statistical approach described in this paper, our

interest is very similar : we seek to accept or reject hypotheses about whether or not there exists any correlation between specific planetary patterns and corresponding life events, not ***why*** such correlations should exist !

The framework presented in this paper can be used to validate these hypotheses. It is a tool that can be used by astrologers to build their own hypotheses that describe how planets influence lives.

The next phase of research could use this model to actually define specific hypotheses and then use the data to accept or reject the same. A collection of hypotheses that have been accepted based on data that is believed to be accurate and genuine would constitute a body of knowledge comparable to that compiled intuitively by astrologers in the past. But since these hypotheses have been accepted on the basis of statistical tests on validated data, the acceptability of the same should be significantly higher within the science and technology community of the twenty-first century.

Acknowledgement

A. Raghuram Hebbar was the subject matter expert on astrology and also provided the non-programmer's perspective that was invaluable in the design of the Parashar21 software. His contribution to the ideas described in this paper is acknowledged with gratitude.

References

1. Majumder, Harihar, ***Folit Jyotish*****,** (Bengali), Puthipatra, 9 Anthony Bagan Lane, Calcutta 700009.

2. Braha, James, ***Ancient Hindu Astrology for the Modern Western Astrologer***, Hermetician Press, Longboat Key, Florida 34228, US. ISBN : 0-935895-04-3

3. Rao, K.N., Ashu Rao, K, ***Learn Hindu Astrology Easily*****,** Vani Publications, F291 Saraswati Kunj, 25 IP Extension, Delhi 110092. ISBN : 81-892221-06-X

4. Levin, R.I., Rubin, D.S., **Statistics for Management,** Prentice Hall, New Delhi, ISBN : 978-81-203-1235-7

5. Gupta, G.K., **Introduction to Data Mining with Case Studies**, Prentice Hall, New Delhi. ISBN : 81-203-3053-6

6. Sparse Matrix, ***Wikipedia***, http://en.wikipedia.org/wiki/Sparse_matrix

7. Davis, Tim. ***Sparse Matrix***. From MathWorld--A Wolfram Web Resource, created by Eric W. Weisstein. http://mathworld.wolfram.com/SparseMatrix.html

8. Beckman et.al., ***Extending RDBMSs To Support Sparse Datasets Using An Interpreted Attribute Storage Format***, Proceedings of the 22nd International Conference on Data Engineering 2006 ISBN:0-7695-2570-9

9. Knorr, E., ***2004 - The Year of Web Services,*** in CIO Magazine, Special Issue on *Fast Forward 2010 - The Fate of IT* , December 2003. [Knorr actually quotes Scott Dietzen, in this article !]

10. O'Reilly, T.,***What Is Web 2.0 Design Patterns and Business Models for the Next Generation of Software***, 2003 http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

11. Mukerjee, Prithwis, ***Business Information Systems*****,** Jaico Publishers, 2009. ISBN : 978-81-7992-998-8

12. Pauniker, Aditya N., ***Web 2.0 – Concepts and Business Models***, Lulu, 2009. Available on-line at http://www.lulu.com/content/paperback-book/web-20---concept-and-business-models/7078379

Illustrations

			Grahas									Lords											Bhava											
<= FactID			lagna	sun	moon	mercury	venus	mars	jupiter	saturn	rahu	ketu	first lord	second lord	third lord	fourth lord	fifth lord	sixth lord	seventh lord	eighth lord	ninth lord	tenth lord	eleventh lord	twelveth lord	first bhava	second bhava	third bhava	fourth bhava	fifth bhava	sixth bhava	seventh bhava	eighth bhava	ninth bhava	tenth bhava	eleventh bhava	twelveth bhava
+	ObjectID =>		X00	X11	X12	X13	X14	X15	X16	X17	X18	X19	X21	X22	X23	X24	X25	X26	X27	X28	X29	X30	X31	X32	X41	X42	X43	X44	X43	X46	X47	X48	X49	X50	X51	X52
	Aspects																																		
Y21	aspected by	sun							Y					Y									Y					Y							
Y22	aspected by	moon																								Y									
Y23	aspected by	mercury									Y																Y								
Y24	aspected by	venus																										Y							
Y25	aspected by	mars																								Y			Y	Y					
Y26	aspected by	jupiter	Y	Y	Y					Y			Y					Y	Y					Y	Y								Y		Y	
Y27	aspected by	saturn				Y						Y					Y			Y						Y					Y			Y		
Y28	aspected by	rahu				Y											Y			Y														Y		
Y29	aspected by	ketu																										Y							
Y31	aspected by	first lord				Y						Y					Y			Y						Y					Y			Y		
Y32	aspected by	second lord	Y	Y	Y					Y			Y					Y	Y					Y	Y								Y		Y	
Y33	aspected by	third lord																								Y			Y	Y					
Y34	aspected by	fourth lord																											Y						
Y35	aspected by	fifth lord									Y																Y								
Y36	aspected by	sixth lord																								Y									
Y37	aspected by	seventh lord							Y					Y									Y					Y							
Y38	aspected by	eighth lord									Y																Y								
Y39	aspected by	ninth lord																											Y						
Y40	aspected by	tenth lord																								Y			Y	Y					
Y41	aspected by	eleventh lord	Y	Y	Y					Y			Y					Y	Y					Y	Y								Y		Y	
Y42	aspected by	twelveth lord				Y						Y					Y			Y						Y					Y			Y		

Illustration 1: Portion of a sparse matrix that stores PAC information

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1																			
2	PARRAHS		**106**		**Life Events**														
3																			
4			pmuk19121026		Female			19-12-1965											
5		**Code**	**Event**		**Sub Event**		**Sub-Sub**									**Dasha**	**Antar**	**Pratvantar**	**Age**
20	**14**	180A	Education		Post Graduate			N											
21	**15**	210D	Break in Education		Illness			Y					date?	22-04-2009		mercury	rahu	rahu	43
22	**16**	220A	Break in Education		Accident			N											
23	**17**	230A	Break in Education		Marriage			N											
24	**18**	240A	Break in Education		Money Matters			N											
25	**19**	250A	Break in Education		Others			N											
26	**20**	310A	Profession		Technical			N											
27	**21**	320A	Profession		Commercial			N											
28	**22**	330A	Profession		Legal			N											
29	**23**	340A	Profession		Medical			N											
30	**24**	350A	Profession		Academic			N											
31	**25**	360A	Profession		Media			N											
32	**26**	370A	Profession		Entertainment			N											
33	**27**	380A	Profession		Administration			N											
34	**28**	390A	Profession		Security			N											
35	**29**	3A0A	Profession		Diff. from Education			N											
36	**30**	3B0D	Profession		Change in Profession			N					date?						
37	**31**	3C0D	Profession		Transfer			N					date?						
38	**32**	3D0D	Profession		Transfer + promotion			N					date?						
39	**33**	3E0D	Profession		Transfer + relocation			N					date?						
40	**34**	410B	Job		First job within age X			N	X =						no data				
41	**35**	420C	Job		X jobs in Y years			N	X =		Y =				no data				
42	**36**	430A	Job		Job Change			N											
43	**37**	440D	Job		Change : disagreement			N					date?						
44	**38**	450D	Job		Change : better salary			N					date?						
45	**39**	460D	Job		Change : better position			N					date?						

Illustration 2: Partial view of the set of events for an individual