TBD

Nam H. Lee & c

March 26, 2011

1 Overview

For each (undirected) pair ij of n vertices and each $t \in [0,1]$, we let $N_{ij}(t)$ be the number of (undirected) communication events between vertex i and vertex j, where each event is marked with one of the communication content topic classes $\{1,2\}$. Let, for each $t \in [0,1]$,

$$\mathcal{D}(t) = \{(\tau_\ell, ij_\ell, k_\ell) : \ell = 1, \ldots, N(t)\},$$

where $N(t) = \sum_{i<j} N_{ij}(t)$ and for simplicity, let

$$\mathcal{D} = \mathcal{D}(1).$$

In this paper, we consider a problem of classifying the n vertices into two groups after seeing $\mathcal{D}$, where the members of each class have the similar degrees of interests in topics from two content topic classes. For this, we propose a classification algorithm derived from an EM-based paramter estimation procedure.

To formulate our EM-based algorithm, we introduce a model for $\mathcal{D}$ through a data augmentation principle. The hidden variables augmenting the data are the collection of random variables $\{\Lambda_{i,k}(t)\}$, where $\Lambda_{i,k}(t)$ is the degree of vertex i's interest in topics from content topic class k at time t. We assume that each $\Lambda_{i,k}(t)$ is a strictly positive random variable such that for some fixed constant $\lambda_i \in (0,\infty)$,

$$\lambda_i = \Lambda_{i,1}(t) + \Lambda_{i,2}(t). \tag{1}$$

Our model, yielding the observation $\mathcal{D}$, is then to be a doubly stochastic marked Poisson process whose latent intensity process is specified by the $n \times 2$ random matrix

$$\Lambda(t) = (\Lambda_{i,k}(t); i = 1, \ldots, n, k = 1, 2). \tag{2}$$

For convenience, we denote the i-th row of $\Lambda(t)$ by $\Lambda_i(t)$.

To demonstrate the peformance of our classification algorithm, we will also consider a so-called vertex nomination problem, in which the memembers of a particular class is already known and the objective is then to decide whether or not the class is to be expanded further and if to be expanded, which (single) vertex is to be included next. A solution to the vertex nomination problem is likely to be useful if some other information other than $\mathcal{D}$ is available. On the other hand, we will also denomstrate the peformance of our classification algorithm when the only available information is $\mathcal{D}$ and the vertex nomination is then interpreted as a two-pass procedure.

2 Single-period model

For each vertex i and time $t \in [0, 1]$, let

$$X_i(t) = (X_{i,1}(t), X_{i,2}(t)), \tag{3}$$

where

$$X_{i,k}(t) = \frac{\Lambda_{i,k}(t)}{\Lambda_i(t)}, \tag{4}$$

and note that each $X_i(t)$ is a probability vector. For each (undirected) pair ij, we assume that the process $N_{ij} = (N_{ij}(t) : t \in [0, 1])$ is a doubly stocastic Poisson process that whose intensity process is

$$\Lambda_{ij}(t) = \lambda_i \lambda_j (X_{i,1}(t) X_{j,1}(t) + X_{i,2}(t) X_{j,2}(t)). \tag{5}$$

A particularly useful chracterization is to describe each N_{ij} as a process obtained by thinning a homogeneous Poisson process C_{ij} whose constant intensity is $\lambda_i \lambda_j$. This affords an interpretation that each t such that $C_{ij}(t) = C_{ij}(t-) + 1$ is a potential communication opportunity between vertex i and vertex j, and at each such t, depending on values of $X_i(t)$ and $X_j(t)$, the opportunity may or may not give rise to an actual communication event, i.e. t such that $N_{ij}(t) = N_{ij}(t-) + 1$. To be more precise, it is our assumption that given the filtration

$$\mathcal{F}_{ij}^{\Lambda}(t) = \sigma(\Lambda_i(s), \Lambda_j(t) : s \leq t),$$

if $C_{ij}(t) = C_{ij}(t-) + 1$, then $N_{ij}(t) = N_{ij}(t-) + 1$ with probability $\Lambda_{ij}(t)$ and the topic mark is of content class k with probaiblity

$$\frac{X_{i,k}(t) X_{j,k}(t)}{X_{i,1}(t) X_{j,1}(t) + X_{i,2}(t) X_{j,2}(t)} = \frac{\Lambda_{i,k}(t) \Lambda_{j,k}(t)}{\Lambda_{i,1}(t) \Lambda_{j,1}(t) + \Lambda_{i,2}(t) \Lambda_{j,2}(t)}.$$

Now, we fix a stationary diffusion process $Y_i = (Y_i(t) : t \in [0,1])$ such that

$$dY_i(t) = \mu(Y_t(t))dt + \sigma_i(Y_i(t))dB_i(t), \tag{6}$$

where each μ_i and σ_i are twice continuously and boundedly differentiable functions, and $B_1, \ldots, B_n$ are independent standard Brownian motions. In fact, in this paper, for concreteness, we assume that

$$dY_i(t) = \beta_i(\mu_i - Y_i(t))dt + \sigma_i dB_i(t), \tag{7}$$

for some fixed constants $\mu_1, \ldots, \mu_n \in \mathbb{R}$, and $\sigma_1 = \ldots = \sigma_n \in (0, \infty)$ and $\beta_1 = \ldots = \beta_n \in (0, \infty)$. Then, we assume that

$$X_{i,1}(t) = \frac{\exp(-Y_i(t))}{1 + \exp(-Y_i(t))}. \tag{8}$$

While we could also take $\sigma_1, \ldots, \sigma_n$ and $\beta_1, \ldots, \beta_n$ to be parameters, but for our vertex nomination problem or more generally classification problem, these potential paramters are (perhaps) nuisance parameters (?). In our algorithm, we propose to fix the nusiance parameters before proceding to a classification/vertex notimation task. Finally, for future reference, we let

$$\psi = (\mu_1, \ldots, \mu_n, \lambda_1, \ldots, \lambda_n).$$

The term single period here emphasises the fact that the parameter Ψ (and the nusiance paramters) are assumed to be constant within time period $[0, 1]$.

3 Greedy algorithm for iterative vertex nomination

Our main result of this paper concerns an algorithm for correctly assigning each vertex into one of two classes, where witnin each group, all the members have the same parameter i.e. if vertex i and vertex j are in the same class, then $\psi_i = \psi_j$. In this section, we provide a greedy algorithm for arriving at a partition of the vertex set which is likely to be the true partition of the vertex set that have produced the data $\mathcal{D}$.

For the initial step $q = 1$, suppose that we have an arbitrary partition $(C_1(1), C_2(1))$ of $\{1, \dots, n\}$. Let ψ^1 be the output of our EM algorithm, where we assume that within each group, all vertice have the same paramter. Then, for each pair $i, j \in C_k(1)$, $\psi_i^1 = \psi_j^1$. Let

$$Q(1) := \mathbf{E}_{\psi(1)}\left[\log(f(\mathcal{D}, Y; \psi^1))\right], \tag{9}$$

where $f(;\psi)$ denote the likelikehood function of observing $\mathcal{D}$ and Y with the given parameter ψ (see the next section for more details about this). Next, for the $(r+1)$-st iteration, suppose that we have a partition $(C_1(r), C_2(r))$ from the r-th iteration. If $C_2(r) = \varnothing$, then our partitioning algorithm terminates, yielding the r-th partition $(C_1(r), C_2(r))$ as the final output. So, we assume that $C_2(r) \neq \varnothing$. Then, for some $\ell > 0$ and $m > 0$ with $\ell + m = n$, we have

$$C_1(r) = \{u_1, \dots, u_\ell\}, \tag{10}$$
$$C_2(r) = \{v_1, \dots, v_m\}. \tag{11}$$

For each $i \in C_2(r)$, let

$$C_1^i(r) = C_1(r) \cup \{i\}, \tag{12}$$
$$C_2^i(r) = C_2(r) \setminus \{i\}. \tag{13}$$

Now, for each i, let $\psi^{i,r}$ be the EM estimate of ψ with the partition $(C_1^i(r), C_2^i(r))$, and let

$$Q^i(r) = \mathbf{E}_{\psi^{i,r}}\left[\log(f(\mathcal{D}, Y; \psi^{i,r}))\right]. \tag{14}$$

If for all $i \in C_2(r)$, $Q^i(r) \le Q(r)$, then the algorithm terminates, yielding $(C_1(r), C_2(r))$ as the final partitioning. Otherwise, we set

$$i^* = \arg\max_{i \in C_2(r)} Q^i(r), \tag{15}$$

and let

$$C_1(r+1) = C_1(r) \cup \{i^*\}, \tag{16}$$
$$C_2(r+1) = C_2(r) \setminus \{i^*\}. \tag{17}$$

4 Estimation by EM procedure

4.1 Overview

Let

(i) M_{ij} is the (total) number of messages between vertex i and vertex j,

(2) $\tau_{ij}(\ell)$ is the time at which occurred the ℓ-th communication event between vertex i and vertex j.

In the expectation stage of our EM algorithm, for each ψ, we compute the expected value of the following random variable:

$$\prod_{i<j} f_{ij}(\mathcal{D}(ij); X_i^r, X_j^r, \psi) \prod_{i=1}^{n} f_i(Y_i^r; \psi), \tag{18}$$

where

(i) Y_i^r is a diffusion process whose distribution is a conditional distribution of Y_i given the observation $\mathcal{D}$ with the unconditional distribution of Y_i for the r-th estimated parameter ψ^r,

(ii) $f_i(\cdot; \psi)$ is the likelihood of the i-th vertex process' sample path Y_i with the parameter ψ,

(iii) $f_{ij}(\cdot; X_i^r, X_j^r, \psi)$ specifies the conditional likelidhood given $X_1^r, \dots, X_n^r$ of observing message at time

$$\tau_{ij}(1), \dots, \tau_{ij}(M_{ij}),$$

repsectively, about topic

$$\kappa_{ij}(1), \dots, \kappa_{ij}(M_{ij}).$$

Note that conditioning on the path of Y_i and Y_j (but not on $\mathcal{D}$), and given the value of ψ, the value of $f_{ij}(\cdot; X_i, X_j, \psi)$ can be seen to be

$$\prod_{i<j} \left(\prod_{\ell=1}^{M_{ij}} \Lambda_{i,\kappa_{ij}(\ell)}(\tau_{ij}(\ell)) \Lambda_{j,\kappa_{ij}(\ell)}(\tau_{ij}(\ell)) \right) \exp \left(- \int_0^1 \Lambda_{ij}(s) ds \right). \tag{19}$$

Next, given $\sigma_i = 1$ and $\beta_i = 1$ for all $i = 1, \ldots, n$, it can be shown that

$$\begin{aligned} & \log(f_i(Y_i^r(t); \psi)) \\ = \;& \mu_i(Y_i^r(1) - Y_i^r(0)) - \frac{1}{2}((Y_i^r(1))^2 - (Y_i^r(0))^2) - \frac{1}{2}\int_0^1 (\mu_i - Y_i^r(t))^2 - 1dt. \end{aligned}$$

Now we remark here that while unconditionally, the diffusion processes $Y_1, \ldots, Y_n$ are independent, conditionally, $Y_1, \ldots, Y_n$ need not be independent. Computing the exact conditionally likelihood of Y_i^r analytically is difficult if not practically impossible even for the relatively simple diffusion process $Y_1, \ldots, Y_n$ that we are considering in this paper. Because of this, our algorithm relies instead on an EM algorithm in where the expected value in the E step is estimated by simulation and in the M step, the maximization is done numerically. An except to this arises if we set the nusiance parameter $\sigma_1 = \ldots = \sigma_n = 0$. In this case, each $Y_i^r = \mu_i$ and it is not difficult to see that the MLE of ψ given $\mathcal{D}$ satisfies the following equations:

$$\frac{1}{\lambda_i} \left(\sum_{j \neq i} M_{ij} \right) - \sum_{j \neq i} \lambda_j (2X_i X_j + 1 - X_i - X_j) = 0, \tag{20}$$

$$\frac{1}{X_{i,1}} \left(\sum_{j \neq i} M_{ij,1} \right) + \frac{1}{1 - X_{j,1}} \sum_{j \neq i} M_{ij,2} - \lambda_i \sum_{j \neq i} \lambda_j (2X_{i,j} - 1) = 0. \tag{21}$$

4.2 Conditional sampling algorithms

Fix an (unordered) pair ij. Note that

$$\mathcal{L}^r_{ij} := \mathbf{E}_r[\log \prod_{m=1}^{M_{ij}} \log(\Lambda_{ij,k_{ij}(\ell)}(\tau_{ij}(\ell)) - \int_0^1 \Lambda_{ij}(s)ds] \tag{22}$$

$$= \sum_{m=1}^{M_{ij}} \log(\lambda_i) + \log(\lambda_j) \tag{23}$$

$$+ \sum_{m=1}^{M_{ij,1}} \mathbf{E}_r[\log(X_{i,1}(\tau_{ij,1}(m)))\,|\mathcal{D}] + \mathbf{E}_r[\log(X_{j,1}(\tau_{ij,1}(m)))\,|\mathcal{D}] \tag{24}$$

$$+ \sum_{m=1}^{M_{ij,2}} \mathbf{E}_r[\log(X_{i,2}(\tau_{ij,2}(m)))] + \mathbf{E}[\log(X_{j,2}(\tau_{ij,2}(m)))\,|\mathcal{D}] \tag{25}$$

$$-\lambda_i - \lambda_j - \int_0^1 \mathbf{E}_r[X_{i,1}(s)X_{j,1}(s) + X_{i,2}(s)X_{j,2}(s)|\mathcal{D}]ds, \tag{26}$$

where $\mathbf{E}_r$ denote the expectation under the parameter ψ^r. To compute the gradient of $\mathcal{L}^r_{ij}$ with respecto ψ, it suffices to know how to evaluate, for any bounded continuous function f, for each $s \in [0,1]$,

$$\mathbf{E}_r[f(X_{i,1}(s))\,|\mathcal{D}] \propto \mathbf{E}_r[f(X_{i,1}(s)); \mathcal{D}], \tag{27}$$

where the proportionality constant does not depend on ψ. Now, note that

$$\mathbf{E}_r[f(X_{i,1}(s)); \mathcal{D}] \tag{28}$$

$$= \mathbf{E}_r[f(X_{i,1}(s))P_r[\mathcal{D}\,\big|\mathcal{F}^X]] \tag{29}$$

$$= \mathbf{E}_r[f(X_{i,1}(s)) \prod_{j \neq i} \prod_{m=1}^{M_{ij}} \lambda^r_i \lambda^r_j X_{ij,k_{ij}(m)}(\tau_{ij}(m)) \exp(-\lambda^r_i \lambda^r_j \int_0^1 X_{i,1}(t)X_{j,1}(t) + X_{i,2}(t)X_{j,2}(t)dt)], \tag{30}$$

In the E step, one needs to know how to compute, for each ψ,

$$Q_\ell(\psi) = \mathbf{E}_{\psi_\ell,\mathcal{D}} \left[\log f_{[0,T]}(\mathcal{D}, X^{(\ell)}; \psi)\right], \tag{31}$$

where the law of $X^{(\ell)}$ is the law of X conditioning on the data $\mathcal{D}$ and the paramter ψ_ℓ. The general closed-form expression of $Q(\psi)$ is difficult to obtain, and we develop a way to obtain a close approximation $\widehat{Q}_\ell$ of Q. To do this, we use a Monte Carlo estimate based on the rejection technique. First, our simulation of each X_i (uncondtionally) is achieved through discriteziation. For example, we can fix a small $\Delta t > 0$ and

$$Y_i(t+\Delta t) = Y_i(t) + \beta_i(\mu_i - Y_i(t))\Delta t + \sigma_i Z \Delta t, \tag{32}$$

where Z is a standard normal random variable. Next, note that

$$\begin{aligned}
f(Y\,|\mathcal{D}\,;\psi) &\propto f(Y,\mathcal{D};\psi) && (33)\\
&= f(\mathcal{D}\,|X,\psi)f(Y\,|\psi) && (34)\\
&= \prod_{i<j} f_{ij}(\mathcal{D}(ij);X_i,X_j,\psi)\prod_{i=1}^{n} f_i(Y_i;\psi), && (35)\\
&\leq \prod_{i<j}(\lambda_i\lambda_j)^{M_{ij}}\prod_{i=1}^{n} f_i(Y_i;\psi). && (36)
\end{aligned}$$

Therefore, (in theory), to simulate conitionally, it is enough to simulate X unconditionally as a proposal, and accept the proposed sample path with probability:

$$\prod_{i<j}\left(\exp\left(-\int_0^1 \lambda_i\lambda_j\langle X_i(\tau_{ij}(\ell)), X_j(\tau_{ij}(\ell))\rangle ds\right)\prod_{\ell=1}^{M_{ij}}\langle X_i(\tau_{ij}(\ell)), X_j(\tau_{ij}(\ell))\rangle\right).(37)$$

On the other hand, the rejection probability (as the algorithm is presented now) is not trivial and as $\sum_{i<j} M_{ij}$ gets larger, the second factor is likely to be a very small number. To remedy this, we use a sequential rejection sampling prodedure. First, we partition the interval $[0,1]$ into many pieces, say, with grid $0, t_1, \ldots, t_L$, so that each interval $[t_\ell, t_{\ell+1})$ contains at most one messaging event (of any kind). Then, progressing from the left end point to the right end point recursively, starting from the $X(t_\ell)$ obtained from the preceding interval simulation, simulate unconditionally. Then, accept and append the proposed sample path segment with the probability:

$$\exp\left(-\int_{t_\ell}^{t_{\ell+1}} \lambda_i\lambda_j\langle X_i(s), X_j(s)\rangle ds\right). \tag{38}$$

if the new interval $[t_\ell, t_{\ell+1})$ has no messaging event but otherwise, accept and append the proposed sample path segment with the probability:

$$\langle X_i(\tau_{ij}^{\ell}), X_j(\tau_{ij}^{\ell})\rangle \exp\left(-\int_{t_\ell}^{t_{\ell+1}} \lambda_i\lambda_j\langle X_i(s), X_j(s)\rangle ds\right), \tag{39}$$

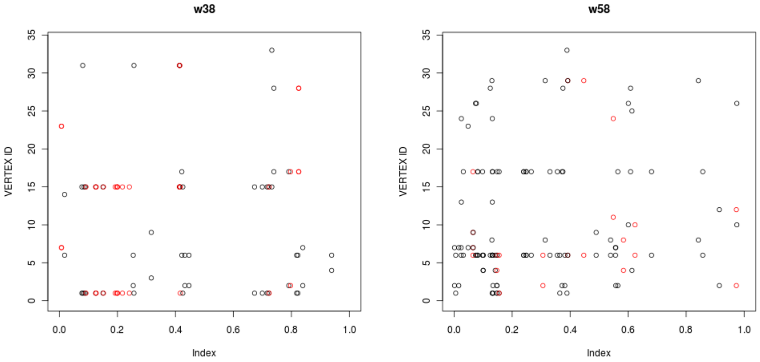

Figure 1: w38 versus w58, the $\mathcal{D}(V_{34})$ data, where $V_{34} = \{1, 2, \ldots, 34\}$

where τ_{ij}^{ℓ} here denotes the occurrence time of the messaging event.

In our experiment, this sequential simulation technique significantly improve the efficiency of algorithm but it is important to note that the size of each $t_{\ell+1} - t_{\ell}$ is chosen with care since making it too small creates another rare event type issue, namely, always accepting the proposed sample path segment when ocassionally rejected.

5 Estimation by Kalman-Bucy filtering

6 Experiment Results

In this section, we present the experiment results done on the so-called Enron E-mail data using the algorithms presented in this paper. The raw full Enron E-mail data does not necessarily conform to our $\mathcal{D}$ data format as we presented in this paper. For example, some E-mails contain more than one receiver or in some cases, the sender and the receiver are one and the same. Unfortunately, there is not a conventionally accepted method for assigning an E-Mail to one or two categories that everyone can agree on, and apriori, our model is not adequate to handle such data. While we hope to extend our algorithm that can handle the full data set in near future, in this paper, we procede to apply our algorithms to a data set that is derived from the raw full Enron Email data so that it fits the format of $\mathcal{D}$. One way to achieve this is to consider only the messages such that the receiver is exactly one

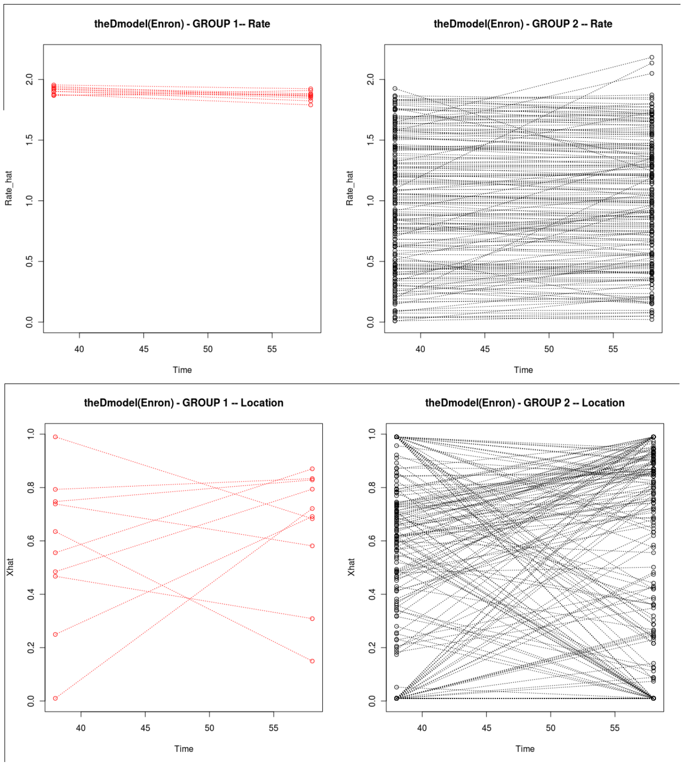

Figure 2: w38 versus w58, $\hat{\lambda}_i$ and $\hat{\mu}_i$ obtained by the initialization step of the EM alglorithm on $\mathcal{D} = \mathcal{D}(V_{184})$, where $V_{184} = \{1, \ldots, 184\}$

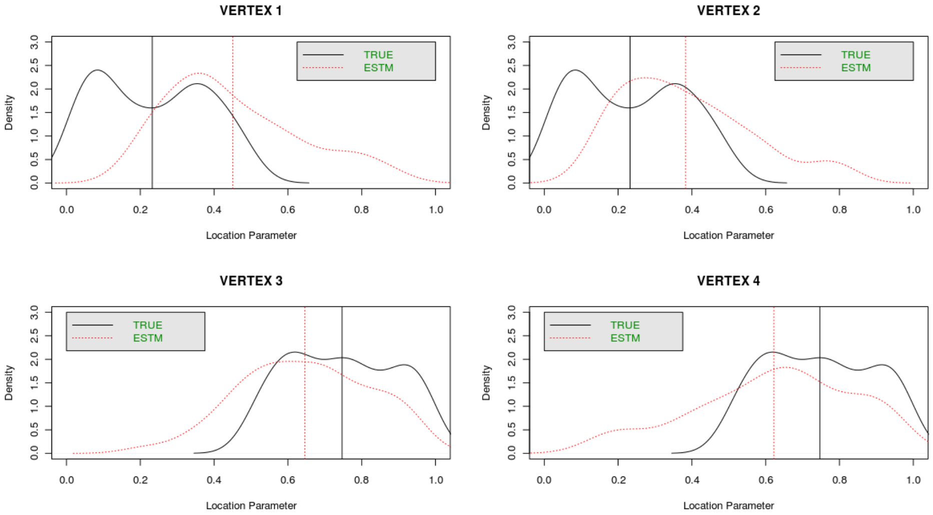

Figure 3: Performance of the algorithm on simulated data

that is also different from the sender and also if there is any message with the same time stampe even after that, we randomly perturb the time stamp minutely ad hoc so that the resulting data conform to our framework. We consider two sets of data. Each one for approximately four months periods starting at the week of 38 and the week of 58 since Jan 1, 1970. These weeks are of interest in this paper because there are other studies done on the Enron Email data that suggests that the first data set is significantly different from the second in their communication pattern (cf. the importance sampling method in [GC]). The method in [GC] relies on a graph theoretic statistic and also suggests that the ten vertices out of 184 vertices form a coherent social/communication group. We aim to support the same claim here instead using our classification algorithm.

The subset of the data $\mathcal{D}$ is presented above only for thirty four vertices in the data, but our experiment is done for 184 vertices. For clarity, for each sub-collection $\mathcal{V}$ of vertices, we let $\mathcal{D}(V)$ denote the subsubet of $\mathcal{D}$, where both the sender and the receiver are from $\mathcal{V}$. The figure in Figure 1 is for $\mathcal{D}(V) \subset \mathcal{D}$, where $V = \{1, \ldots, 34\}$. In Figures 2, the left hand side is for week 38 and the right hand side is for week 58.

The algorihm run on a simulated data shows that it does right thing.