Statistical Disclosure Control using PRAM

Eric Cator, André Hensbergen[†], Yves Rozenholc[‡]

[†] TU Delft, EWI (DIAM), Mekelweg 4, 2628 CD Delft, e.a.cator@ewi.tudelft.nl, a.t.hensbergen@ewi.tudelft.nl

[‡] Université Pierre et Marie Curie - Boîte courrier 188, P.O.Box 75252 Cedex 05 Paris France, yves.rozenholc@math.jussieu.fr

Abstract. We will look at Statistical Disclosure Control where our goal is to protect microarray data against spontaneous recognition by a user, by applying a known transition matrix P to each row of our microarray. This method was invented by Statistics Netherlands and is called PRAM. We want to choose the transition matrix P in such a way that our loss of information is minimal, while at the same time guaranteeing a certain level of security. In this paper we will define what is meant by this level of security and we will consider possible loss of information measures, also focussing on the applicability of the method.

1. Introduction

Statistics Netherlands (CBS) gathers each year a huge amount of data concerning the Dutch people. Some of these data are sensitive, and CBS is only able to obtain these data if it can guarantee that this sensitive data is not directly available to outsiders. Also Dutch privacy legislation is quite strict and enforces CBS to take appropriate measures. We will focus on one of these measures, namely to ensure that if CBS lets some researcher or company (we will call this the user) work with certain microarray data, then CBS has to modify this microarray in such a way that each individual in the data (this corresponds to one row in the microarray) is protected against spontaneous recognition by the user. This definition is still rather vague, so let us describe the situation in more detail.

Our microarray consists of a matrix, where each row corresponds to some individual (this maybe for example a person or a company) and each column corresponds to some property of the individual, for example age, salary of place of residence, which we will call the variables. CBS distinguishes two kinds of variables: the identifying variables and the sensitive variables. The identifying variables are properties of the individual that others may know or could easily find out, such as sex or place of residence. The sensitive variables are properties of the individual that are normally not known to outsiders, such as salary or health. We will call the identifying variables $\xi_1, \xi_2, \ldots, \xi_m$ and the sensitive variables $y_1, \ldots, y_M$. We define $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_m$ the space of possible values for $\xi = (\xi_1, \ldots, \xi_m)$. Likewise we define $\mathcal{Y}$ as the space of possible values for $y = (y_1, \ldots, y_M)$. Furthermore, we will call the number of individuals in the data n. We use the notation $\xi_j(i)$ for the value of the j^{th} variable corresponding to individual i. Similarly we define $\xi(i)$ and $y(i)$.

CBS states that when a user spontaneously recognizes an individual, she can only do this by overviewing 3 identifying variables at the same time. If she were to use more than 3 variables, she would be intentionally trying to find someone, and CBS has taken legal measures to prevent this. This choice of three identifying variables is defined by choosing a map

$$\pi : \{1, 2, 3\} \to \{1, 2, \ldots, m\}.$$

The set of all possible choices of three variables that we want to protect against is denoted by Π. We define $\xi_\pi = (\xi_{\pi(1)}, \xi_{\pi(2)}, \xi_{\pi(3)})$, and likewise $\mathcal{X}_\pi$. In Section 2 we will define precisely what we mean by protecting against spontaneous recognition.

The way we will secure our data is by using a transition matrix P_{kl}, where $k, l \in \mathcal{X}$. This means that the actual data set CBS will give to its user consists of entries $(X(i); y(i))$ $(i = 1, \ldots, n)$, where each $X(i)$ is a random variable such that

$$\mathrm{P}\left(X(i) = l \mid \xi(i) = k \right) = P_{kl}.$$

Each row will be transformed like this, independently of each other, so all $X(i)$ are independent, but not identically distributed! This method of securing the data is called PRAM. The idea is that the user not only receives the modified data, but also the transition matrix P, so that she can still make proper statistical inference, but she cannot spontaneously recognize an individual and thus find out sensitive information about him. In Section 3 we will describe how we can choose P such that the data is secure, but the loss of information is minimal.

2. Secure against spontaneous recognition

Suppose our user received the microarray $(x(i); y(i))$ $(i = 1, \ldots, n)$ from CBS, together with a transition matrix P. She would normally try in some way to retrieve the original data (ξ, y), then look at three of the identifying variables ξ_π and think that she recognizes a certain individual. She knows that this individual, let's call him John, has the value $k_0 \in \mathcal{X}_\pi$ for the three identifying variables $\pi(1), \pi(2)$ and $\pi(3)$. She is therefore interested in the possibility that $\xi_\pi(i) = k_0$, for a certain i that she believes to be John. However, this is not all. If she concludes

that indeed $\xi_\pi(i) = k_0$, then she still might not be sure that i is John. It might be that a lot of individuals have the value k_0 for ξ_π.
Let us suppose our user takes the following Bayesian approach in deciding whether i is John or not: since there are n individuals in the microarray, the prior probability (so without any information) of i being John equals $1/n$. Now she knows that $x_\pi(i) = k_0$. Therefore we get

$$\begin{aligned}
\mathrm{P}\left(i = \text{John} \mid x_\pi(i) = k_0\right) &= \frac{\mathrm{P}\left(x_\pi(i) = k_0 \mid i = \text{John}\right) \cdot \mathrm{P}(i = \text{John})}{\sum_{j=1}^{n} \mathrm{P}\left(x_\pi(j) = k_0 \mid \xi(j)\right) \cdot \frac{1}{n}} \\
&= \frac{\sum_{k: k_\pi = k_0} P_{\xi(\text{John})\text{k}}}{\sum_{j=1}^{n} \sum_{k: k_\pi = k_0} P_{\xi(j)k}} \\
&= \frac{\sum_{k: k_\pi = k_0} P_{\xi(\text{John})\text{k}}}{\sum_{l \in \mathcal{X}} U_0(l) \sum_{k: k_\pi = k_0} P_{lk}}.
\end{aligned}$$

Here we define

$$U_0(l) = \#\{j : \xi(j) = l\}.$$

She would decide that indeed i =John if

$$\mathrm{P}\left(i = \text{John} \mid x_\pi(i) = k_0\right) > \alpha,$$

for some significance level α (which may be specified by CBS). Therefore, if we want to protect our data, we need the following condition on the transition matrix P:

C1: For each $m \in \mathcal{X}$ such that there exists i with $\xi(i) = m$, for each $\pi \in \Pi$ and $k_0 \in \mathcal{X}_\pi$, we must have

$$\frac{\sum_{k: k_\pi = k_0} P_{mk}}{\sum_{l \in \mathcal{X}} U_0(l) \sum_{k: k_\pi = k_0} P_{lk}} \leq \alpha. \tag{1}$$

For example, if P is the identity matrix (so the original data is given to the user), then for $m \in \mathcal{X}$ with $m_\pi = k_0$, we get

$$\frac{\sum_{k: k_\pi = k_0} P_{mk}}{\sum_{l \in \mathcal{X}} U_0(l) \sum_{k: k_\pi = k_0} P_{lk}} = \frac{1}{U_\pi(k_0)}.$$

Here,

$$U_\pi(k_0) = \#\{j : \xi_\pi(j) = k_0\}.$$

So condition (C1) can only hold if for all π and k_0 we have $U_\pi(k_0) \geq 1/\alpha$ (or $U_\pi(k_0) = 0$). In other words, there should be no rare combinations in the original data.

The other extreme is when P has constant entries, so all information about the data is lost. Then

$$\frac{\sum_{k:k_\pi=k_0} P_{mk}}{\sum_{l\in\mathcal{X}} U_0(l)\sum_{k:k_\pi=k_0} P_{lk}} = \frac{1}{n},$$

so condition (C1) is always satisfied (at least if $n \geq 1/\alpha$). This shows that there always exist transition matrices that satisfy (C1).

One final remark is that

$$\frac{\sum_{k:k_\pi=k_0} P_{mk}}{\sum_{l\in\mathcal{X}} U_0(l)\sum_{k:k_\pi=k_0} P_{lk}} \leq \frac{1}{U_0(m)}. \tag{2}$$

This shows that if $U_0(m) \geq 1/\alpha$, so if $m \in \mathcal{X}$ occurs frequently enough in the original data, then Condition (C1) always holds for this m, for any choice of π, k_0 and P.

Condition (C1) can be rewritten as a linear constraint for P. However, (1) has to hold for all $\pi \in \Pi$ and all $k_0 \in \mathcal{X}_\pi$. The total number of conditions might grow exponentially (or even faster) if one doesn't control the set Π properly. One would have to think about which combinations of variables is reasonable for a spontaneous recognition.

3. Minimal loss of information

We now know which condition the transition matrix P has to satisfy, but we still have to choose an optimal P, in some sense. We want to choose P such that the information loss is minimal, but we need to quantify this loss.

Our user is interested in estimating some property of the original data $\{(\xi(i); y(i)) : 1 \leq i \leq n\}$. In fact, she will be interested in a property of the empirical measure μ of all the rows $(\xi(i); y(i))$. A logical way to estimate this property is using the data available to her (i.e. $\{(x(i); y(i)) : 1 \leq i \leq n\}$) to obtain an estimate $\hat{\mu}$ of μ, and use this $\hat{\mu}$ to estimate this property. Since we do not know which property the user will be interested in, we could try and make sure that $\hat{\mu}$ is a good estimate for μ. We have to choose an estimator for μ, and we choose the Maximum Likelihood estimator. Our information loss will now be measured in terms of how far in expectation $\hat{\mu}$ lies from μ.

The Maximum Likelihood Estimator. To define the MLE, the user will assume that our original data is generated by some measure μ^* on $\mathcal{X} \times \mathcal{Y}$. So she assumes that

$$(\Xi; Y) \sim \mu^*.$$

Given the transition matrix P, she then knows that the log-likelihood of her data $\{(x(i); y(i)) : 1 \leq i \leq n\}$ is given by:

$$l(\mu^*) = \sum_{i=1}^{n} \log(\sum_{k \in \mathcal{X}} \mu^*(k; y(i)) \cdot P_{kx(i)}).$$

Now define $\mu^*_{\mathcal{Y}}$ as the marginal measure on $\mathcal{Y}$, i.e. the distribution of Y. Then

$$\mu^*(k; y) = \mathrm{P}^*(\Xi = k | Y = y) \cdot \mu^*_{\mathcal{Y}}(y).$$

Define

$$\mu^*(k|y) = \mathrm{P}^*(\Xi = k | Y = y).$$

We then have

$$l(\mu^*) = \sum_{i=1}^{n} \log(\mu^*_{\mathcal{Y}}(y(i))) + \sum_{i=1}^{n} \log(\sum_{k \in \mathcal{X}} \mu^*(k|y(i)) \cdot P_{kx(i)}).$$

This shows that we can maximize over $\mu^*_{\mathcal{Y}}$ and $\mu^*(k|y)$ separately. The first term is just the log-likelihood when we have a sample $y(1), \ldots, y(n)$ from $\mu^*_{\mathcal{Y}}$, so the MLE for $\mu^*_{\mathcal{Y}}$ is equal to the empirical measure of the $y(i)$'s, which makes sense, since the $y(i)$'s are not changed by P. So $\hat{\mu}_{\mathcal{Y}} = \mu_{\mathcal{Y}}$.

In order to find the MLE for $\mu^*(k|y)$, we introduce some vector notation: P is a matrix in $\mathbb{R}^{\mathcal{X} \times \mathcal{X}}$ and each μ^*_i is a vector in $\mathbb{R}^{\mathcal{X}}$ such that

$$\mu^*_i(k) = \mu^*(k|y(i)).$$

Define the following vectors in $\mathbb{R}^{\mathcal{X}}$:

$$b_i = P^t \mu^*_i.$$

For $a, b \in \mathbb{R}^{\mathcal{X}}$ we define $\langle a, b \rangle = \sum_{k \in \mathcal{X}} a(k) b(k)$. Then

$$\langle b_i, 1 \rangle = \langle \mu^*_i, P1 \rangle = \langle \mu^*_i, 1 \rangle = 1.$$

This means that each b_i represents a probability measure on $\mathcal{X}$. Furthermore, if P is invertible, which we will assume from now on, we get that

$$\mu^*_i = (P^{-1})^t b_i.$$

So if we can maximize

$$\tilde{l}(b) = \sum_{i=1}^{n} \log(b_i(x(i))), \tag{3}$$

we would have that

$$\hat{\mu}(k|y(i)) = \sum_{l \in \mathcal{X}} \hat{b}_i(l) P^{-1}_{lk}.$$

However, in Equation (3) we recognize the loglikelihood for the data $\{x(i) : 1 \leq i \leq n\}$ with

$$\mathrm{P}(X(i) = k) = b_i(k).$$

So if we define

$$U(k; y) = \#\{i : (x(i); y(i)) = (k; y)\} \quad \text{and} \quad U(\cdot; y) = \#\{i : y(i) = y\},$$

we get that

$$\hat{b}_i(k) = \frac{U(k; y(i))}{U(\cdot; y(i))}.$$

Our conclusion is (note that $\mu_{\mathcal{Y}}(y) = U(\cdot; y)/n$):

$$\hat{\mu}(k; y) = \frac{1}{n} \sum_{l \in \mathcal{X}} P^{-1}_{lk} U(l; y). \tag{4}$$

Note that we can interpret $U(k; y)$ as a random variable by replacing $x(i)$ by $X(i)$ in its definition; this also makes $\hat{\mu}$ into a random measure, so we can calculate its expected distance (which we need to choose) to μ.

The L^2 loss. We define the loss as some expected deviation of $\hat{\mu}$ from μ, the original empirical measure. We will take this expectation only with respect to the change of the variables using P, so we will no longer view $(\xi(i); y(i))$ as a realization of a random variable. In information theory, a usual measure for the deviation of $\hat{\mu}$ from μ is the Kullback-Leibler divergence:

$$l_{\mathrm{KL}}(P) = \mathrm{E}_P \left[\sum_{(k;y) \in \mathcal{X} \times \mathcal{Y}} - \log \left(\frac{\hat{\mu}(k; y)}{\mu(k; y)} \right) \mu(k; y) \right].$$

Here $\mathrm{E}_P[\cdot]$ denotes expectation with respect to the probability measure induced by the transition matrix P; remember that $\hat{\mu}(k; y)$ is a stochastic variable whose distribution depends on P (and on the original data, of course). This is why we denote our loss function l_{KL} as a function of P.

Another possible choice for our loss function is the quadratic loss, or L^2-loss:

$$l_2(P) = \mathrm{E}_P \left[\sum_{(k;y) \in \mathcal{X} \times \mathcal{Y}} \left(\hat{\mu}(k; y) - \mu(k; y) \right)^2 \right].$$

This leads to a more feasible loss function than l_{KL}, i.e. one that leads to an easier optimization problem, which is important if we want the method to work for large datasets and, more importantly, for large Π.

We will spend some time evaluating $l_2(P)$. Define

$$U_0(l;y) = \#\{i : (\xi(i);y(i)) = (l;y)\}.$$

We will assume that P is invertible. This means that we can use Equation (4) to see that

$$\begin{aligned}
\mathrm{E}_P[\hat{\mu}(k;y)] &= \frac{1}{n}\sum_{l\in\mathcal{X}} P^{-1}_{lk}\mathrm{E}_P[U(l;y)] \\
&= \frac{1}{n}\sum_{l\in\mathcal{X}} P^{-1}_{lk}\sum_{l'\in\mathcal{X}} P_{l'l}U_0(l';y) \\
&= \frac{1}{n}U_0(k;y) \\
&= \mu(k;y).
\end{aligned}$$

This shows that $\hat{\mu}$ is an unbiased estimator of μ. So we get

$$l_2(P) = \sum_{(k;y)} \mathrm{E}_P\left[\hat{\mu}(k;y)^2\right] - \mu(k;y)^2.$$

Now we concentrate on the random variables $U(l;y)$, for fixed y:

$$\begin{aligned}
\mathrm{E}_P[U(l;y)U(l';y)] &= \mathrm{E}_P\left[\sum_{\{i,j:y(i)=y(j)=y\}} 1_{\{X(i)=l\}}1_{\{X(j)=l'\}}\right] \\
&= \sum_{\{i\neq j:y(i)=y(j)=y\}} P_{\xi(i)l}P_{\xi(j)l'} + \delta_{ll'}\sum_{\{i:y(i)=y\}} P_{\xi(i)l}.
\end{aligned}$$

This means that

$$\begin{aligned}
n^2\mathrm{E}_P\left[\hat{\mu}(k;y)^2\right] &= \sum_{l,l'\in\mathcal{X}} P^{-1}_{lk}P^{-1}_{l'k}\mathrm{E}_P[U(l;y)U(l';y)] \\
&= \sum_{l,l'\in\mathcal{X}}\sum_{\{i\neq j:y(i)=y(j)=y\}} P^{-1}_{lk}P^{-1}_{l'k}P_{\xi(i)l}P_{\xi(j)l'} + \sum_{l\in\mathcal{X}}\sum_{\{i:y(i)=y\}}\left(P^{-1}_{lk}\right)^2 P_{\xi(i)l} \\
&= \sum_{\{i\neq j:y(i)=y(j)=y\}} \delta_{\xi(i)k}\delta_{\xi(j)k} + \sum_{l\in\mathcal{X}}\sum_{\{i:y(i)=y\}}\left(P^{-1}_{lk}\right)^2 P_{\xi(i)l} \\
&= U_0(k;y)^2 - U_0(k;y) + \sum_{l\in\mathcal{X}}\sum_{\{i:y(i)=y\}}\left(P^{-1}_{lk}\right)^2 P_{\xi(i)l}.
\end{aligned}$$

Therefore,

$$\begin{aligned} l_2(P) &= \sum_{(k;y)\in\mathcal{X}\times\mathcal{Y}} \left[\frac{U_0(k;y)^2}{n^2} - \frac{U_0(k;y)}{n^2} + \frac{1}{n^2} \sum_{l\in\mathcal{X}} \sum_{\{i:y(i)=y\}} \left(P^{-1}_{lk}\right)^2 P_{\xi(i)l} - \frac{U_0(k;y)^2}{n^2} \right] \\ &= \frac{1}{n^2} \sum_{(k;y)\in\mathcal{X}\times\mathcal{Y}} \sum_{\{i:y(i)=y\}} \sum_{l\in\mathcal{X}} \left(P^{-1}_{lk}\right)^2 P_{\xi(i)l} - \frac{1}{n} \\ &= \frac{1}{n^2} \sum_{i=1}^{n} \sum_{k,l\in\mathcal{X}} \left(P^{-1}_{lk}\right)^2 P_{\xi(i)l} - \frac{1}{n}. \end{aligned}$$

This function of P is feasible as a loss function, as we will show in the next section.

4. Implementation

We will focus on a relatively small example, where $\mathcal{X} = \{0,1\}^4 \times \{0,1,2\}$, with 48 elements. This means that our PRAM matrix P will be a 48×48 matrix, where 48 is the total number of distinct elements in $\mathcal{X}$. It is easy to check that the number of different combinations (π, k_0) is 104. Handling bigger data sets requires a more sophisticated approach to the optimization problem, but we do wish to point out that according to inequality (2), we only need to check Condition (C1) for those $m \in \mathcal{X}$ whose frequency in the micro array is smaller than $1/\alpha$, where α is the required security level. This means that if we consider a lot of identifying variables (so $\mathcal{X}$ is big), but our micro array contains a lot of individuals, we still might end up with a reasonably small number of side conditions for our optimization. We will give some comments on bigger $\mathcal{X}$ later on.

In our example we used a micro array of 2500 individuals and a security level of $\alpha = 0.1$. Furthermore we incorporated some probability structure on the variables to get a reasonable number of rare individuals. It turned out that seventeen individuals m from $\mathcal{X}$ satisfied $0 < U_0(m) < 10$ $(= 1/\alpha)$. Only for those seventeen we needed to check Condition (C1). A good starting value for P when solving the optimization problem is the identity matrix I. In general, for large micro arrays it turns out that the optimal P will be very close to I. The explanation is: suppose we give elements in $\mathcal{X}$ that are quite frequent in the micro array small probabilities to be changed into rare elements, and leave the other elements unaltered. Then it will already be likely that the occurrence of a rare combination of π and k_0 is actually due to the PRAM transformation, and therefore will not lead to the identification of an individual. Of course, if there are many rare elements in $\mathcal{X}$ and few frequent ones, the optimal P might be quite different from I.

To check that the optimal P in our example is indeed close to I, we noted that the lowest diagonal element was 0.9971. Furthermore, we found that, applying the PRAM transformation to our array X, using this P, on average leads to only 4 changes (out of 2500 possible changes). This implies only a small loss in information, and indeed the L^2-loss is a mere $6.4 \cdot 10^{-7}$. To give some more insight into the structure of P, we consider Figure 1, which shows the probability of an element being altered, set out against its frequency in the micro array X.

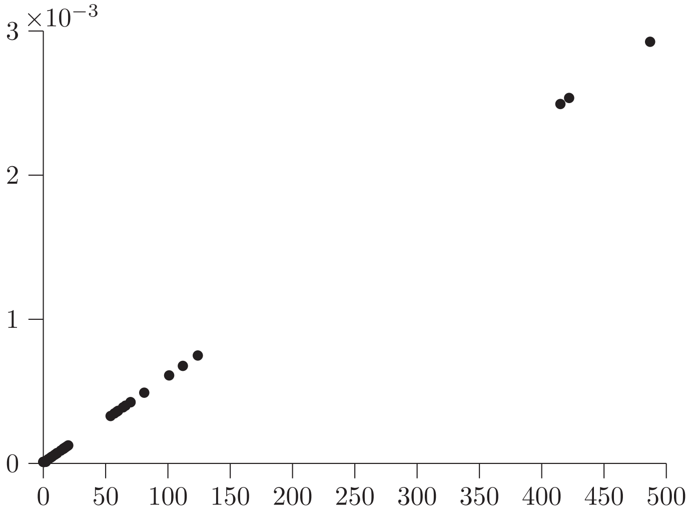

Figure 1. Probability of change against frequency.

As one can see, there is almost a linear relationship. When zooming in it appears that only for elements with very small frequencies there are deviations from this linearity.

We also noted that if an element is changed during the PRAM procedure, there is a 0.99999 chance that it gets changed into one of the following three elements: $(0,0,0,0,0)$, $(0,0,0,0,1)$ or $(0,0,0,0,2)$. The reason is that there were only five combinations of (π, k_0) that satisfied $U_\pi(k_0) < 1/\alpha$, and for all these five combinations there exists a k out of the three elements above, such that $k_\pi = k_0$. In other words, these three elements are enough to take care of the problem cases. Remember that only if there are (π, k_0) with $0 < U_\pi(k_0) < 1/\alpha$ something has to be done, that is, only then the micro array needs to be changed (otherwise Condition (C1) would be satisfied for $P = I$, the identity matrix.) Finally, it turned out that when the PRAM transformation is applied to the original micro array, in 88 % of the cases an element that undergoes a change is one of the three most frequently occurring elements.

These observations can help when dealing with large micro arrays: to find a good PRAM matrix, it might be enough to optimize over all P that only allow changes from frequent elements to elements that together cover the problem combinations of π and k_0. This way one may not find the optimal P, but perhaps one whose information loss is still acceptable.

We conclude with two inequalities containing the left hand side of (C1) that are useful for big micro arrays. For both inequalities we assume that the PRAM matrix P has all its diagonal elements $P_{kk} > \gamma$, for some γ close to 1.

First, for any $m \in \mathcal{X}$:

$$\begin{aligned}\frac{\sum_{k:k_\pi=k_0} P_{mk}}{\sum_{l\in\mathcal{X}} U_0(l)\sum_{k:k_\pi=k_0} P_{lk}} &\leq \frac{1}{\sum_{k:k_\pi=k_0} U_0(k)P_{kk}}\\ &\leq \frac{1}{\gamma U_\pi(k_0)}.\end{aligned}$$

This means that we only have to check combinations of π and k_0 that satisfy

$$U_\pi(k_0) \leq \frac{1}{\gamma\alpha}.$$

In our example we could have picked $\gamma = 0.99$, since our optimal P has all diagonal elements bigger than 0.99, but we do not know this before we do the optimization. However, choosing $\gamma = 0.9$ could already drastically reduce the number of combinations that needs to be checked. If in this way you find diagonal elements lower than γ, you could start the optimization again with a smaller γ.

Secondly, for elements $m \in \mathcal{X}$ for which $m_\pi \neq k_0$, we can show that

$$\frac{\sum_{k:k_\pi=k_0} P_{mk}}{\sum_{l\in\mathcal{X}} U_0(l)\sum_{k:k_\pi=k_0} P_{lk}} \leq \frac{1-\gamma}{\gamma\cdot U_\pi(k_0)}. \tag{5}$$

Namely, if $m_\pi \neq k_0$, for the numerator we have

$$\sum_{k:k_\pi=k_0} P_{mk} \leq 1-\gamma,$$

since the sum does not contain the diagonal element, and the denominator is taken care of as above. So if the right hand side of (5) is smaller than α, then for the given m condition (C1) is certainly fulfilled. Now from $U_\pi(k_0) \geq 1$ it follows that

$$\frac{1-\gamma}{\gamma\cdot U_\pi(k_0)} \leq \frac{1-\gamma}{\gamma},$$

so if the last expression is smaller than α, we have can verify Condition (C1) by checking all $m \in \mathcal{X}$, $\pi \in \Pi$ and $k_0 \in \mathcal{X}_\pi$ such that

- $U_0(m) < \frac{1}{\alpha}$

- $U_\pi(k_0) < \frac{1}{\gamma\alpha}$
- $m_\pi = k_0$.