ON USING SIMULATION AND STOCHASTIC LEARNING FOR PATTERN RECOGNITION WHEN TRAINING DATA IS UNAVAILABLE

The Case of Disease Outbreak

Dragos Calitoiu[1] and B. John Oommen[1, 2]

[1]*School of Computer Science, Carleton University, 1125 Colonel By Drive, Ottawa, Canada*

[2]*Department of Information and Communication Technology*
University of Agder, 36 Grooseveien, Grimstad, Norway

Keywords: Disease outbreak, Stochastic point location, Learning automata.

Abstract: Pattern Recognition (PR) involves two phases, a Training phase and a Testing Phase. The problems associated with training a classifier when the number of training samples is small are well recorded. Typically, the matrices involved are ill-conditioned and the estimates of the probability distributions are very inaccurate, leading to a very poor classification system. In this paper, we report what we believe are the pioneering results for designing a PR system when there are absolutely *no* training samples. In such a scenario, we show how we can use a model of the underlying phenomenon and combine it with the principle of stochastic learning to design a very good classifier. By way of example, we consider the case of disease outbreak: Learning the Contagion Parameter in a black box model involving healthy, sick and contagious individuals. The parameter of interest involves η which is the probability with which an infected person will transmit the disease to a healthy person. Using the theory of Stochastic Point Location (SPL), the problem is reduced to a PR or classification problem in which the SPL is first subjected to a *training* phase, the outcome of which is used for the *testing* phase.

1 INTRODUCTION

The Training phase of every PR problem is, in one sense, the most difficult. It first of all involves understanding the type of classifier that is to be used. Once this is determined, the various class-conditional distributions have to be learned, and this incorporates all the facets of learning. Indeed, it is so all-pervasive that almost all of PR considers how training can be achieved for parametric/non-parametric data, for different distributions and when one encounters the “curse of dimensionality”. But in every case, the basic premise is that the designer is given a reasonable number of training samples. Of course, every practitioner would like an “infinite” number of training samples, because then, everything converges beautifully. The problems associated with training a classifier when the number of training samples is small, are well recorded. In such cases, the matrices involved are ill-conditioned and the estimates of the probability distributions are very inaccurate, implying that the classification system is very poor.

In this paper, we consider how we can design a PR system when we have *no* training samples. Although such cases are rare, they are extremely important. Indeed, policy-related decisions concerning the spread of infectious diseases always involve phenomena that have almost never been encountered before. In a certain disease outbreak, we show how we can use a model of the underlying phenomenon and combine it with the principles of stochastic learning to design a very good classifier. As far as we know, the entire process of designing PR systems and training them in such situations is open, and thus, we believe that the results we present are, in one sense, pioneering. To clarify issues, rather than dealing with a PR problem in an abstract domain, we shall consider how PR can be achieved in the so-called “*Learning the Contagion Parameter*” (LCP) problem.

As a preface, we first present the environment against which the LCP problem must be studied and the proposed solution utilized. We assume that we are dealing with a geographical area of fixed dimensions which could represent a certain district of a city. In the

Calitoiu D. and John Oommen B. (2010).
ON USING SIMULATION AND STOCHASTIC LEARNING FOR PATTERN RECOGNITION WHEN TRAINING DATA IS UNAVAILABLE - The Case of Disease Outbreak.
In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence - Artificial Intelligence*, pages 45-52
DOI: 10.5220/0002716800450052

interest of simplicity the population of this area is assumed constant. When an infection starts within this geographical area it is desirable, first of all, that the contagion is contained *within* the area. The second issue which is primarily of concern to our present study involves understanding how the disease can spread to healthy people *within* this geographical area. In other words, we would like to determine when the outbreak of the disease is under control, and also to be able to detect an uncontrollable explosion - which are the respective classes in our PR problem. Once we are able to detect these scenarios, it will be the task of the policy makers to propose strategies by which quarantine decisions are made. This present study and the proposed results, hopefully, submit a small step in this direction. To present the problem in the right perspective, we submit a brief explanation of the disease outbreak problem and the epidemiological model used.

1.1 The Disease Outbreak Problem

A disease outbreak is commonly defined as the occurrence of an illness in a community or region, where the number of cases of the illness occur with a frequency clearly in excess of normal expectancy. The number of cases indicating the presence of an outbreak will vary according to the infectious agent, size and type of the population exposed, previous experience or lack of exposure to the disease, and the time and place of the occurrence(s). Thus, the status of an outbreak is relative to the usual frequency of the disease in the same area, among the same population, and at the same season of the year (Chin, 2000).

In outbreak situations, one often must introduce preventive interventions to control pathogen transmission and adverse outcomes. Control measures that have proven effective in similar outbreaks in the past can thus be immediately implemented.

Confirming an outbreak begins with the calculation of the background rate of infection (or adverse event) and then comparing the outbreak period rate with the background rate. Such a comparison can be performed using the rate ratio (R_r):

$$R_r = \frac{Attack\ rate\ during\ epidemic\ period}{Attack\ rate\ during\ background\ period} \quad (1)$$

An outbreak become *uncontrollable* when the proposed control measures are not able to keep the rate ratio constant or, to decrease it.

The models are used by policymakers, public health workers, and other researchers who want to analyze and compare the outcomes to better understand how an outbreak occurs, and how a disease spreads. They are also useful for understanding how to respond to emerging infectious diseases. If a disease outbreak occurs, simulations can be done by tuning specific models to aid public officials in their decision-making processes.

Our method is an alternative approach to the following two classical methods, namely, a Sensitivity Analysis and Monte Carlo Investigations, respectively.

Sensitivity analysis estimates what the true effect measure (e.g., the rate ratio) would be in the light of the observed data and a hypothetical level of bias. It produces one or more hypothetically adjusted point estimates for the specific measure of interest (Greenland, 1998).

Monte Carlo investigations, (i.e., at least in the present context), involve the simulation of real phenomena, or their idealized models, involving a random or probabilistic element in their structure, by the deliberate use of random numbers. These methods have already played an important role in many applications of stochastic models and processes, both by way of background material (in understanding qualitatively some of the properties of such models), and more quantitatively, in the study of particular problems that are not amenable to complete mathematical solution (Robert and Casella, 2005).

We propose a new methodology to investigate how an outbreak occurs, which possesses an advantage of less computational effort compared with the above-mentioned approaches.

1.2 Principles of Contagion

Epidemiology is the study of the spread of any disease, with regard to space and time. Its objective is to trace the factors and parameters that are responsible for, or contribute to, their occurrence (Diekmann and Heesterbeek, 2000).

In the process of studying the spread of a disease, the spatial structure of a population, its density and the geographical area involved can have a major contribution. As opposed to this, the non-spatial mathematical models can provide only a simple image about the dynamics of the transmission. Consequently, this requires, in many situations, more realistic models that include the description of the space and the spatial contact between individuals. The basic idea of studying the dynamics of spreading a disease is to distinguish individuals from one another according to their features, their interaction with the environment, and involves discovering the variation in time of these features. In the case of an infectious disease, the principal characteristic is the computation of the "force" of infection of a given agent that modifies the state of an individual (Diekmann and Heesterbeek, 2000).

The state of the individual is specified with the minimal "degree of freedom" so that we can adequately predict the future of the individual from an epidemiologic point of view. We attempt to describe the state of the population and nature, symbolically given as $S(t+1)$ at time '$t+1$', be only a function of $S(t)$. The evolution uses the assumption that the state $S(t)$ completely captives the history of what happened prior to the time instant, 't', as per the so-called Markovian assumption. Also, the simplest epidemic models assume that any pair of individuals is equally likely to interact with each other during a given time interval. This is referred to as the *well-mixed* assumption.

In our research, we investigate the problem of learning the Contagion Parameter (CP) in the process of transmission and perpetuation of a virus in a (human) population, by using the so-called *regular lattice* model.

A classical example of such a model for contagion simulates the transmission of a virus in a human population. The model is initialized with *N* people, of which *I* are infected at time 't'. People move randomly about the lattice, being either (i) healthy but susceptible to infection (*S*), (ii) sick and infectious (*I*), (iii) recovered and immune (*R*), or (iv) dead (*D*). We do not allow for individual to reproduce or enter in the geographical area, implying that *N*=*S*+*I*+*R*+*D*.

The density of the population also affects how often infected, immune and susceptible individuals come into contact with each other. To render the model tractable, we assume that infection leads to either death due to the illness or immunity. Thus, no individual can be infected twice. Additionally, we model the system with the understanding that contamination can take place when two individuals, either of whom is sick, make "contact" with each other. This is characterized by the probability that a contact between a contagious and a susceptible person actually leads to transmission. This is symbolized by the Contagion Parameter (CP), η, **assumed to be constant** but unknown. Finally, the period of the infection is the same for all individuals. Thus, the infectivity quantifies the probability that the transmission of the illness will occur when a contagious and a susceptible person occupy the same physical grid location.

1.3 Formal Model of Contagion

Solving the problem in its absolute generality - even from a modeling perspective - is probably intractable. To render this problem tractable, obviously, we have to resort to some simplifying assumptions listed below.

Assumption A1: We assume that we are working within a square grid of width W units.

Assumption A2: We assume that there are N, constant, individuals moving in this grid in a locally randomized manner. The number of sick people at time 't' is given by $n_s(t)$.

Assumption A3: We assume that we are working in a discretized time space, $t = 1 \cdots t_{max}$ where t_{max} is the total period of observation.

Assumption A4: We consider a scale-independent model.

Assumption A4.1: First of all, the number of individual is specified in term of a density coefficient ρ, where $\rho = \frac{N}{W^2}$.

Assumption A4.2: The number of the sick people at any time 't', namely $n_s(t)$, will be determined by the ratio σ, where $\sigma(t) = \frac{n_s(t)}{N}$.

Assumption A5: We assume that every individual C_i is characterized by a 4-tuple $\langle x_i(t), y_i(t), s_i, d_i \rangle$ where, at any time instant t,

Assumption A5.1: $x_i, y_i \in \{1 \cdots W\}$ (i.e. the conditions of the location of the individual).

Assumption A5.2: s_i is an indicator signifying the state of the individual C_i. By convention, a healthy person is assigned the index $s_i = 0$. He is sick if $s_i = 1$, immune if $s_i = 2$, and dead if $s_i = 3$.

Assumption A5.3: d_i is the duration of time that has elapsed since the instant when the value s_i was set to unity. It represents the time that has elapsed since the sick individual was infected.

Assumption A6: At every time instant the individual C_i is permitted to move to one of his neighboring grid locations or to stay where it is. We assume that the individual stays at its current location with the probability θ (assumed to be fixed and known, and moves to one of its neighboring locations with the probability $\frac{1-\theta}{K}$, where K is the number of cells which are neighboring to $\langle x_i, y_i \rangle$.

Assumption A7: Every individual can infect another with a probability η which is the unknown parameter within the simplified model of contagion.

Assumption A8: Every sick individual can either die or become immune to the illness after a period, τ, also referred to as the *Period of Infectivity*. Only during this period he is capable of infecting a healthy person.

With the above assumptions, we first formalize the model. We assume that N individuals are moving within the grid of width W, and that at every time instant each individual is allowed to stay or move to a neighboring cell as per **Assumption A6**. Whenever two individuals are on the same grid point, the contagion possibilities are three-fold: (i) If both the individuals are healthy they remain healthy. (ii) If both the individuals are infected, they continue to be infected. (iii) If exactly one of them is sick, he contaminates the other with the probability η. Based on these assumptions, since ρ is the density of the number of individuals and σ_t is the proportion[1] of sick people (among the entire population) at time t, σ_{t+1} has the following form: $\sigma_{t+1} = f(\sigma_t, \rho, \eta)$, where the functional form of $f(\cdot)$ is unknown. Our aim is, quite simply, to achieve the PR on the process, which in turn implies estimating η, and we achieve this by using a learning methodology applied to solve the Stochastic Point Location (SPL) problem.

2 STOCHASTIC POINT LOCATION

The SPL problem (Oommen, 1997; Oommen and Raghunath, 1998; Oommen et al., 2006) considers a general learning problem in which the learning mechanism (which could be a Learning Automaton (LA), or in general, an algorithm) attempts to learn a "parameter", say η^*, within a closed interval. Consider the problem of a robot moving around on a real line attempting to locate a particular point. To assist the mechanism, it communicates with an Environment which provides it with information regarding the direction in which it should go. If the Environment is deterministic, the problem is the "Deterministic Point Location Problem".

This problem is akin to the field of LA (Lakshmivarahan, 1981; Narendra and Thathachar, 1989; Poznyak and Najim, 1997; Thathachar and Sastry, 2003), in which the learning mechanism attempts to learn from a *stochastic Teacher*. More specifically, unlike the traditional LA model in which the LA attempts to learn the optimal action offered by the Environment, we consider the following learning problem: the learning mechanism is trying to locate an unknown *point* on a real interval by interacting with the stochastic Environment through a series of informed guesses.

Unlike the deterministic problem alluded to above, in the SPL, rather than receive deterministic responses as to where it should go, the learning mechanism is given, at every time step, a stochastic (i.e., possibly erroneous) response. Thus, when it should really be moving to the "right" it may be advised to move to the "left" and vice versa, as formalized below.

2.1 Formulation of the SPL Problem

We assume that there is a learning mechanism whose task is to determine the optimal value of some variable (or parameter), η. We assume that there is an optimal choice for η - an unknown value, say $\eta^* \in [0,1]$. The question which we study here is the one of learning η^*. Although the mechanism does not know the value of η^*, we assume that it has responses from an intelligent "Environment" E which is capable of informing it whether the current value of η is too small or too big. E may tell us to increase η when it should be decreased, and vice versa. However, to render the problem tangible we assume that the probability of receiving an intelligent response is $p > 0.5$.

Observe that the quantity "p" reflects on the "effectiveness" of the Environment, E. Thus, whenever the current $\eta < \eta^*$, the Environment correctly suggests that we increase η with probability p. It simultaneously could have incorrectly recommended that we decrease η with probability $(1-p)$. Similarly, whenever $\eta > \eta^*$, the Environment tells us to decrease η with probability p, and to increase it with probability $(1-p)$.

We shall assume that η is any number in the interval $[0,1]$. The question of generalizing thus will be considered later. The crucial issue that we have to address is that of determining how to change our guess of η^* in $[0,1]$. We shall attempt to do this in a discretized manner by subdividing the time unit interval into R steps $\{0, \frac{1}{R}, \frac{2}{R}, ..., \frac{R-1}{R}, 1\}$, where R is the resolution of the learning scheme. A larger value of R will ultimately imply a more accurate convergence to the unknown η^*.

The scheme which attempts to learn η^* is as below. Let $\eta(t)$ be the value at time step "t". Then,

$$\eta(t+1) := \eta(t) + \frac{1}{R}, \tag{2}$$

if E suggests increasing η and $0 \leq \eta(t) < 1$;

$$\eta(t+1) := \eta(t) - \frac{1}{R}, \tag{3}$$

if E suggests decreasing η and $0 < \eta(t) \leq 1$.

At the end states, the scheme obeys:

$$\eta(t+1) := \eta(t), \tag{4}$$

if $\eta = 1$ and E suggests increasing η;

$$\eta(t+1) := \eta(t), \tag{5}$$

if $\eta = 0$ and E suggests decreasing η.

The Markov Chain representing these transitions is given in Figure 1 below.

Notice that although the above rules are deterministic, because the "environment" is assumed faulty,

[1] In the interest of readability, for all time instants, σ_t will be used to represent $\sigma(t)$.

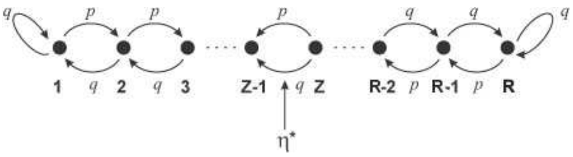

Figure 1: The Markov Chain used for solving the SPL. The states are represented here by integers in {0,1,2,..., R}, where state *i* represents a value of $\eta = \frac{i}{R}$.

the state transitions are stochastic. The main result concerning the above scheme is the following:
Theorem. The parameter learning algorithm specified by Equations (2)-(5) is asymptotical optimal.

The proof of the theorem can be found in (Oommen et al., 2008).

3 USING SPL FOR LEARNING THE CP AND TO ACHIEVE PR

This section describes the strategy by which the solution to the SPL will be used to learn the CP of the contagion model. In essence, we shall reduce this to a Pattern Recognition (PR) problem, and thus, this will consist of two phases: a *Training* phase and a *Testing* phase. The Training phase can be considered to be a calibration module, which essentially learns how an arbitrary disease will spread under the constraints of our model. This can be obtained in a completely non-real-life setting because we need to understand how the number of sick individuals at any time is related to the number of sick individuals at the prior time instant, the density, ρ, and the contagion parameter, η. The Testing phase would then involve dealing with the sick individuals in the real life or simulated setting, and attempting to estimate η. We now discuss both of these phases.

3.1 The Training Phase

We shall first explain the principle used in Training, and then illustrate the results of the training that we have achieved.

As mentioned in the modeling phase, since ρ is the density of the number of individuals and σ_t is the proportion of sick people (among the entire population) at time t, σ_{t+1} has the following form: $\sigma_{t+1} = f(\sigma_t, \rho, \eta)$. Clearly, the functional form of f is unknown, and it is our task to first of all learn it, and to secondly, use the *learned* form in the inference problem associated with learning what the true CP, η^* is.

The Training is achieved by explicitly invoking the functional interdependence of σ_{t+1} on σ_t. This will, in turn, be achieved by allowing the individuals, to move and contaminate/recover from the illness as per the model specified earlier. Thus, the training phase can be given by the following algorithm, formally described below in **Algorithm Training Phase**.

Assumptions: The algorithm is able to move the location of an individual as per **Assumption A6**. This is achieved by invoking the function *Move*.
Input:
W: The size of the grid, which is assumed to be a square $W \times W$.
N: The number of individuals in the grid.
θ: The probability of an individual not moving in the grid.
n_s : The number of initial sick individuals.
η: The probability of infection.
τ: The Period of Infectivity.
p_r: The Probability of Recovery
Output:
The updated n_s as a function of ρ, the density of the population and η.

```
BEGIN Algorithm Training Phase
Initialize every $C_i = \langle x_i, y_j, s_i, d_i \rangle$ to a random position $x_i$, $y_i$.
Set the value of $s_i$ to be 1 with the proportion determined by $N$, $\rho$, and $n_s(0)$.
$d_i$ is initialized to zero for every individual.
repeat forever
/*Compute the position for each individual*/
for i=1 to N do
    $\langle x_i(t+1), y_i(t+1) \rangle := Move[\langle x_i(t), y_i(t) \rangle, \theta]$;
end for
/*Compute the sick state for each individual*/
for i=1 to N do
    if ($s_i(t) = 1$ and $d_i < \tau$) then
        $s_i(t+1) = 1$; $d_i = d_i + 1$;
    end if
    if ($s_i(t) = 1$ and $d_i = \tau$) then
        $s_i = 2$ with probability $p_r$, and $s_i = 3$ with with probability $1 - p_r$.
    end if
    /*Check if $C_i$ should get infected depending on where he is and on his neighbors*/
    if $\exists$ a $k$ s.t $\langle x_k(t+1), y_k(t+1) \rangle = \langle x_i(t+1), y_i(t+1) \rangle \bigwedge s_k(t+1) = 1 \bigwedge s_i(t) = 0$ then
        $s_i(t+1) = 1$ with probability $\eta$; /* $C_i$ becomes infected */
        $n_s = n_s + 1$; /* Update the number of sick people */
    end if
end for
end repeat
END Algorithm Training Phase
```

To now, obtain the training curves, we will have to merely run **Algorithm Training Phase** for different initialized parameters. Observe that we have also considered the question of individuals recovering and dying, and are simultaneously able to keep track of the ratio $\sigma(t+1) = \frac{n_s(t+1)}{N}$, as a function of $\sigma(t) = \frac{n_s(t)}{N}$, since we have available the quantities $n_s(t)$ and $n_s(t+1)$.

Since the phenomenon of contagion also depends on the density, ρ, we have to do the training for two different types of settings listed below:(i) In the first setting, we assume that ρ is constant, and that η is the free variable. In other words, if the density of individuals within the grid is constant, the question now is that of understanding how $\sigma(t+1)$ is a function of $\sigma(t)$ as η changes. (ii) In the second setting, we assume that η is constant, and that ρ is the free variable. In other words, in this setting, if the contagion parameter is fixed, we would like to observe how the disease spreads as the density of the population increases. Thus, the question now is one of understanding how $\sigma(t+1)$ is a function of $\sigma(t)$ as ρ changes. The functional form of $n_s(t+1)$ in terms of $n_s(t)$ and η^* is is merely a "scaled" version of the functional form of $\sigma(t+1)$ in terms of $\sigma(t)$ and η^*, and so, to avoid confusion, we shall refer to both of them by the same function $f(\cdot)$.

3.1.1 Results of the Training Phase

The training phase was conducted for numerous settings. However, since the graphs display random variables, any meaningful representation will have to involve an ensemble of experiments. In our case, we report the results for only one set of parameters and for an ensemble of 10 experiments each, as explained below: $N = 500$ individuals were placed in the grid of dimensions $W \times W$, where $W = 30$ units. This corresponds to a value of $\rho = 0.55$. For this scale, Algorithm Training_Phase was allowed to run, and the quantities $\sigma_{t+1} = f(\sigma_t, \rho, \eta)$ were recorded for several values of $n_s(0)$.

The parametric 3-dimension plots of $\sigma_{t+1} = f(\sigma_t, \rho, \eta)$ are given in Figure (2). This figure depicts the case when ρ is constant, and η is the free variable. Similarly, the 3-dimension plots of $\sigma_{t+1} = f(\sigma_t, \rho, \eta)$ when η is constant, and when ρ is the free variable are given in Figure (3). The general observations that can be made from these two graphs are: (i) The graphs are always monotonically increasing; (ii) As the density increases, even if the CP, η, is maintained constant, the disease will spread; (iii) If the density is kept constant, the disease will spread (as we see it, more rapidly) as the CP, η increases; (iv) Since we are computing Present_state/Next_State maps, the saturation conditions are never encountered.

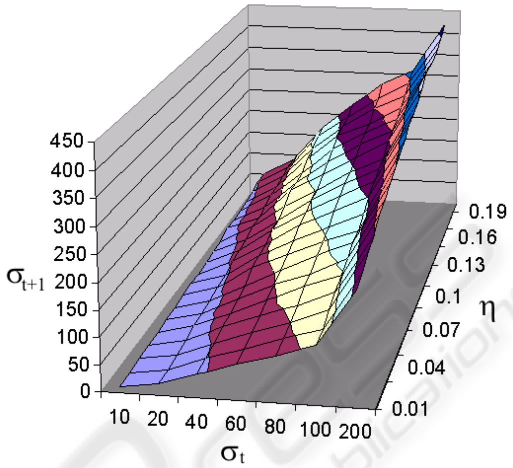

Figure 2: The evolution of the ensemble average over 10 experiments of $\sigma_{t+1} = f(\sigma_t, \rho, \eta)$ as a function of η when $\rho = 0.55$ (i.e., when 500 individuals are placed in the 30×30 grid). The scale for the X and Y axes should be divided by $N = 500$.

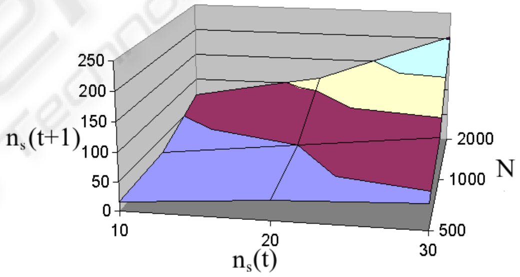

Figure 3: The evolution of the ensemble average over 10 experiments of $n_s(t+1) = f(n_s(t), \rho, \eta)$ as a function of ρ when $\eta = 0.05$. Observe that the plot here is not of σ, but rather of $n_s(\cdot)$. The value of σ at any point can be obtained by dividing the value of $n_s(t+1)$ by the corresponding value of N.

3.2 The Testing Phase

The reader will observe that we have reduced our LCP problem to a Pattern Recognition (PR) or *classification* problem. Indeed, we have transformed the problem to one of *classifying* a certain disease outbreak phenomenon in terms of its contagion parameter, η. Thus, as in every PR problem, we are first required to do a training phase which trains the classifier, and then achieve the testing. The question now is one of devising an efficient testing module which uses the training graphs and figures obtained above. We plan to accomplish the Testing by using the SPL as an integral part of the PR process.

Assume that we currently have $\eta(t)$, that it is our current estimate of the true but unknown CP, η^*. Based on the value of the density of people and the proportion of sick people within this area, and our current understanding of $\eta(t)$, we can now use our training plots to locate the current quiescent point on the corresponding curve. Now, using this as our estimate, we can determine the proportion of sick people that would result on the next day, if η^* was indeed $\eta(t)$. If the number of sick people *estimated* by $\eta(t)$ for this current quiescent point exceeds the *actual* number of sick people that do occur, we have an indication that our current estimate $\eta(t)$ is too high. In that case, we chose to decrease $\eta(t)$ by one step based on the resolution parameter, to obtain $\eta(t+1)$. Similarly, if $\eta(t)$ underestimates the number of sick people at the next time instant, $\eta(t+1)$ is increased to the next corresponding curve in the testing graphs. As long as these step-sizes are relatively small and the parameter η^* is unchanged, the results of Section 3 will guarantee that $\eta(t)$ converges to η^*. Of course, to assist us in this updating, it is more beneficial to have functional plots of $\sigma_{t+1} = f(\sigma_t, \rho, \eta)$ as 2-dimensional plots. One of these plots is given in Figure 4. Similar graphs can be obtained for the other settings but are not included here.

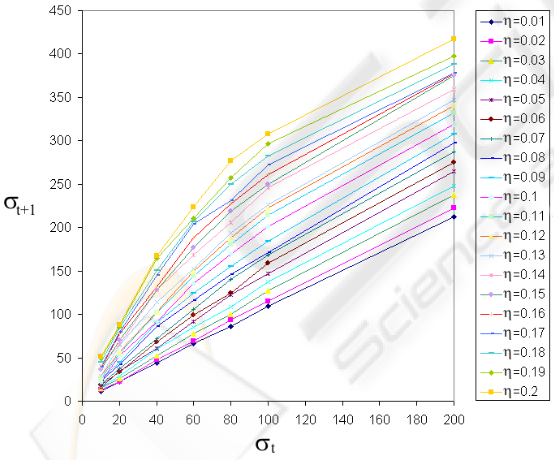

Figure 4: The 2-dimensional evolution of the ensemble average over 10 experiments of $n_s(t+1) = f(n_s(t), \rho, \eta)$ as a function of η when $\rho = 0.55$.

A simple example will help clarify matters. Suppose that we are dealing with a value of η^*, unknown to the user, but known to be in the closed interval $[0.01, 0.2]$. Also, let us assume that we are dealing with the scenario when $N = 500$ individuals are within the $W \times W$ area. We assume that the resolution parameter R divides $[0.0, 0.2]$ in steps of 0.01. Let us assume that the current value of $\eta(t)$ for the testing phase is 0.03. Suppose that the number of sick individuals at time 't' is $n_s(t) = 20$. If, indeed, η^* was 0.03, the number of sick people at time '$t+1$' would have been (on the average) $n_s(t+1) = 26$. If, however, it turns out that in the testing scenario $n_s(t+1) = 25$, we have an indicator that $\eta(t)$ *underestimates* η^*, and so the solution to the SPL would dictate that we increase $\eta(t)$ by the resolution (i.e., 0.01) to set $\eta(t+1)$ to 0.04. The alternate scenario describes the *overestimating* case and is omitted. By successively updating $\eta(t)$, we move along different curves on the corresponding 3-dimensional plot described by $\sigma_{t+1} = f(\sigma_t, \rho, \eta)$, to converge to η^*.

Table 1: Results obtained by using the SPL algorithm to learn the CP when $\eta^* = 0.09$. The starting value is $\eta(0) = 0.05$; the resolution for updating η was set to be 0.01. We denote by n_{s_Tr} the number of sick people from the training phase and by n_{s_Ts} the number of sick people from the testing phase.

step	η	n_{s_Tr}	n_{s_Ts}	$n_{s\_Tr} < n_{s\_Ts}$	$\eta \pm \Delta\eta$
0	0.05	10	10		
1	0.05	16	20	16<20 yes	0.05+0.01
2	0.06	33	52	33<52 yes	0.06+0.01
3	0.07	90	92	90<92 yes	0.07+0.01
4	0.08	158	169	158<169 yes	0.08+0.01
5	0.09	268	267	268<267 **stop**	$\eta^* = 0.09$

The parameter learning mechanism described here was experimentally evaluated to verify the validity of our analytic results and to examine its rate of convergence. Based on the assumption that the learning algorithm was ignorant of η^*, at any time instant, the number of sick people was used to update η as described earlier. The results of the algorithm are presented for two cases listed below: (i) In the first case, the value of the unknown η^* was set to be 0.09. The starting value of η was initialized to 0.05. The results are presented in Table 1, from which we observe that $\eta(t)$ converges to η^*. (ii) In the second case, the value of the unknown η^* was set to be 0.02. The starting value of η was again initialized to 0.05, and $\eta(t)$ again converges to η^* (Table 2).

4 CONCLUSIONS

In this paper, we have considered how we can design a PR system when we have *no* training samples. Although such cases are rare, they are extremely important, for example, when we consider policy-related decisions concerning the spread of in-

Table 2: Results obtained by using the SPL algorithm to learn the CP when $\eta^* = 0.02$. The starting value is $\eta(0) = 0.05$; the resolution for updating η was set to be 0.01.

step	η	n_{s_Tr}	n_{s_Ts}	$n_{s\_Tr} < n_{s\_Ts}$	$\eta \pm \Delta\eta$
0	0.05	10	10		
1	0.05	16	12	16<12 no	0.05-0.01
2	0.04	18	13	18<13 no	0.04-0.01
3	0.03	19	18	19<18 no	0.03-0.01
4	0.02	27	27	27<27 **stop**	$\eta^* = 0.02$

fectious diseases. Thus, the PR solution, which involves stochastic learning, considers the problem of recognizing the seriousness of a contagion by learning the Contagion Parameter (CP) in a black box model involving healthy, sick and contagious individuals. In our study, PR involves the parameter of interest, η, which is the probability with which an infected person will transmit the disease to a healthy person. η is learnt using the theory of Stochastic Point Location (SPL) which reduces the issue to a PR problem with Training and Testing phases.

The following are *some* of the possible avenues for future work:

1. As mentioned in (Oommen, 1997; Oommen and Raghunath, 1998; Oommen et al., 2006), apart from the problem being of importance in its own right, the SPL also has potential applications in solving optimization problems. Indeed, a SPL can be used as a scheme by which the parameters of an optimization algorithm can be determined, so as to prevent it from converging all-too sluggishly on the one hand, or from converging erroneously or oscillating, on the other. The use of the solution to the SPL to *optimally* converge to η^* within our model of contagion is open.
2. An SPL can also be used to assist in the design of neural networks. Thus, if we consider the backpropagation neural network, it is well known that given a particular input, the network uses its "current" set of weights to compute the corresponding output. The obtained output is compared to the expected output and the network weights are consequently modified so as to minimize the expected resultant error. Thus, we could use the solution to the SPL to devise neural techniques to learn η^*.
3. The problem of learning η^* when this quantity changes with time is an extremely interesting problem. We believe that solutions to this problem could have profound implications in a real-life pandemic.
4. It would be very interesting to see if a SPL solution can be applied to a contagion model which is more general than the one we have considered here. We believe that such a strategy is both feasible and expedient.

REFERENCES

Chin, J. (2000). *Control of Communicable Disease Manual*. American Public Health Association, Washington.

Diekmann, O. and Heesterbeek, J. A. P. (2000). *Mathematical Epidemiology of Infectious Diseases: Model Building, Analysis and Interpretation*. John Wiley.

Greenland, S. (343–357, 1998). Basic methods for sensitivity analysis and external adjustment. In *K.J. Rothman KJ and S. Greenland - Modern epidemiology, 2nd ed. Philadelphia, PA: Lippincott-Raven Publishers*. Lippinicot Williams & Wilkins.

Lakshmivarahan, S. (1981). *Learning Algorithms Theory and Applications*. Springer-Verlag.

Narendra, K. S. and Thathachar, M. A. L. (1989). *Learning Automata*. Prentice-Hall.

Oommen, B. J. (SMC-27B:733–739, 1997). Stochastic searching on the line and its applications to parameter learning in nonlinear optimization. In *IEEE Transactions on Systems, Man and Cybernetics*.

Oommen, B. J., Kim, S. W., Samuel, M., and Granmo, O. C. (SMC-38B:466-476, 2008.). A solution to the stochastic point location problem in meta-level nonstationary environments. In *IEEE Transactions on Systems, Man and Cybernetics*.

Oommen, B. J. and Raghunath, G. (SMC-28B:947–954, 1998). Automata learning and intelligent tertiary searching for stochastic point location. In *IEEE Transactions on Systems, Man and Cybernetics*.

Oommen, B. J., Raghunath, G., and Kuipers, B. (SMC-36B:820–836, 2006.). Parameter learning from stochastic teachers and stochastic compulsive liars. In *IEEE Transactions on Systems, Man and Cybernetics*.

Poznyak, A. S. and Najim, K. (1997). *Learning Automata and Stochastic Optimization*. Springer-Verlag, Berlin.

Robert, C. P. and Casella, G. (2005). *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer.

Thathachar, M. A. L. T. and Sastry, P. S. (2003). *Networks of Learning Automata : Techniques for Online Stochastic Optimization*. Kluwer Academic, Boston.