Bayesian Classification vs. k-Nearest Neighbor Classification for the Non-Invasive Hepatic Cancer Detection

Smaranda BELCIUG

Faculty of Mathematics and Computer Science,
Department of Computer Science,
University of Craiova, Romania
smaranda.gorunescu@gmail.com

Abstract. The aim of this study is to compare the classification performances of two well-known algorithms: naive Bayesian and k-nearest neighbor, using a real medical dataset concerning the hepatic cancer. Three main performance measures have been used: the classification accuracy, sensitivity and specificity. This approach can help physicians to make fast and accurate decisions in the non-invasive hepatic cancer detection.

Keywords: naive Bayesian classification, k-nearest neighbor algorithm, hepatic cancer
Math. Subject Classification 2000: 62C10; 62H30

1 Introduction

Bayesian classifiers are a class of simple probabilistic algorithms which apply the Bayes' theorem in order to learn the underlying probability distribution of the data. A difficulty arises when we have more than a few variables and classes. In this case, we have to require an enormous number of observations to estimate these probabilities. Naive Bayesian classification solves this problem by not requiring that we have a lot of observations for each possible combination of the variables. Rather, the variables are assumed to be independent one each other and, thus, we assume that the effect of a variable value on a given class is independent of the values of the other variables. This assumption is called *class conditional independence* and is made to simplify the computation (in this sense it is considered to be "naive").

Recall that studies comparing classification algorithms have found that the naive Bayesian classifier is comparable in performance with decision trees and neural network classifiers.

The k-nearest neighbors (K-NN) classification rule is a technique for non-parametric supervised pattern classification, representing a type of instance-based learning. Basically, for the K-NN algorithm, a new instance is classified based on the majority of the K-nearest neighbor categories. The purpose of this

algorithm is to classify a new object based on attributes and training samples. Given a new instance, we find K objects (or training points) closest to this instance. The classification is using majority vote among the classification of the K objects.

The K-NN algorithm has some strong consistency results. Thus, it is guaranteed to approach the Bayes error rate, for some value of K (where K increases as a function of the number of data points). In this study we try to compare the above two algorithms, by using a concrete medical database, concerning patients with and without hepatic cancer (HCC).

2 Bayesian classifier

Let X be a data sample whose class label is unknown. Let A be some hypothesis such as that the sample X belongs to a specified class Ω. For classification problems we want to determine P(A/X) -the probability that the hypothesis A holds, given the observed data sample X. P(A/X) is the *posterior probability* of A conditioned on X, and P(A) is the *prior probability* of A.

The Bayes' theorem is useful in that it provides a way of calculating the posterior probability P(A/X) from P(A), P(X) and P(X/A), which may be estimated from given data. Technically, the Bayes' theorem says that:

$$P(A/X) = \frac{P(X/A)P(A)}{P(X)}$$

Naive Bayesian classifier assumes that the effect of an attribute value of a given class is independent of the values of the other attributes. This assumption, known as 'class conditional independence', is made to simplify the computation involved. It works as follows:

1. Each sample is represented by an m-dimensional feature vector $X = (x_1, x_2, ..., x_m)$, depicting a measurement made on the sample from n attributes, respectively $A_1, A_2, ..., A_m$.
2. Suppose that there are p classes $\Omega_1, \Omega_2, ..., \Omega_p$. Given a data sample X which has no class label, the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X. Thus, it assigns the unknown sample X to the class Ω_j if and only if: $P(\Omega_j/X) > P(\Omega_i/X), 1 \leq i \leq p, i \neq j$

   By Bayes formula, we have:

   $P(\Omega_i/X) = \frac{P(X/\Omega_i)P(\Omega_i)}{P(X)}$
3. Only $P(X/\Omega_i)P(\Omega_i)$ needs to be maximized because $P(X)$ is constant. The class prior probabilities may be estimated by $P(\Omega_i) = \frac{n_i}{n}$, where n_i is the number of training samples of class Ω_i and n is the total number of training samples.

4. In order to reduce the computation effort in evaluating $P(X/\Omega_i)$, the naive assumption of class conditional independence is made. This presumes that there are no dependence relationships among the attributes. Thus:
   $P(X/\Omega_i) = \prod_{k=1}^{m} P(x_k/\Omega_i)$

The probabilities $P(x_k/\Omega_i)$ can be estimated from the training sample. Thus:

- If A_k is a categorical variable, then:
  $P(x_k/\Omega_i) = \frac{n_{ik}}{n_i}$, where n_{ik} is the number of training samples of class Ω_i having the value x_k (n_i is the number of training samples of class Ω_i).
- If A_k is a continuous variable, then the attribute is assumed to have a Gaussian distribution:
  $P(x_k/\Omega_i) = \frac{1}{\sqrt{2\pi}\sigma_{\Omega_i}} \exp - \left(\frac{(x_k - m_{\Omega_i})^2}{2\sigma^2_{\Omega_i}} \right)$
  where m_{Ω_i} and σ_{Ω_i} are the mean and standard deviation, given the values of attribute A_k for training samples of class Ω_i.

Theoretically, the Bayesian classifiers have the minimum error rate in comparison to all other classifiers, but in practice this is not always the case, owing to inaccuracies in the assumption of class conditional independence.

3 K-nearest neighbor classification

K-NN classification is one of the most fundamental and simple classification methods and should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data. The goal of this method is to separate the data, based on the assumed similarities between various classes. Thus, the classes can be differentiated from one another by searching for similarities between the data provided.

The K-NN classifier requires three things:

- The set of stored records (training data);
- A distance metric to compute distance between records;
- The value of k, the number of nearest neighbors to retrieve.

To classify an unknown record we need to:

- Compute the distance to other training records;
- Identify k nearest neighbors;
- Use class labels of nearest neighbors to determine the class label of unknown record (e.g. by taking majority vote).

The K-NN classifier is commonly based on the Euclidean distance:

$d(X,Y) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2}$

where $X = (x_1, x_2, ..., x_m)$ and $Y = (y_1, y_2, ..., y_m)$, between a test sample and the specified training data. The K-NN classification rule is to assign to a new, unknown, object the majority category label of its k nearest training data. Technically, we take the majority vote of class labels among the k-nearest neighbors and we weigh the vote according to distance by using a weight factor. The only major problem is the choice of the value of k (if k is too small, the method is sensitive to noise points, and if k is too large, neighborhood may include points from other classes).

4 Comparison methodology

In order to assess which methodology performs better, we have used both the *confusion matrix*, given by:

Count		**Predicted class**	
		+	-
Actual class	+	a = true positive (TP)	b = false negative (FN)
	-	c = false positive (FP)	d = true negative (TN)

and the *cost matrix*, given by:

Cost		**Predicted class**	
		+	-
Actual class	+	p = cost (TP)	p = cost (FN)
	-	r = cost (FP)	s = cost (TN)

Thus, we have:

Accuracy = (a+d)/(a+b+c+d)
Cost = p*a + q*b + r*c + s*d

Moreover, we consider the following two classification parameters:

Sensitivity = $\frac{TP}{TP+FN}$,
Specificity = $\frac{TN}{TN+FP}$.

5 The dataset

We have applied the two classifiers to a concrete dataset consisting of 299 patients from the Department of Gastroenterology, University Hospital of Craiova, with or without hepatic cancer (classes Ω_1 = HCC and Ω_2 = non-HCC patients).

The final diagnosis has been established by using the classical biopsy procedure. In the classification process we have used as attributes the most important serum enzymes: total bilirubin (TB), direct bilirubin (DB), indirect bilirubin (IB), alkaline phosfatase (AP), gamma glutamyl transpeptidase (GGT), leucine amino-peptidase (LAP), aspartate amino transferase (AST), alanine amino transferase (ALT), lactic dehydrogenase (LDH), prothrombin index (PI), Gamma, Albumin, Glycemia, Cholesterol and age.

6 Results

We applied the two classification algorithms, by using the Java programming language. The results of this approach are displayed in Table 1 below.

Table 1. The comparison results between the two classifiers

Classifier	Accuracy	Sensitivity	Specificity	Costs
Bayesian classifier	93%	33%	100%	603
K-NN classifier	90%	0%	100%	900

Remark 1. The two classifiers provide near similar results on this particular database.

2. The specificity value is the same for the two classifier (100%). The higher the specificity, the fewer healthy people are labelled as sick.

3. The sensitivity differs from a classifier to another one. The higher the sensitivity, the fewer real cases of diseases go undetected.

7 Conclusions

In this study we have compared two well-known classifiers by using a real medical dataset concerning the hepatic cancer, They have been used to detect the hepatic cancer (HCC) in 299 patients depending on several important serum enzymes. The classifiers worked well on this database, with comparable accuracies. This computer-aided medical diagnosis will provide physicians with a powerful tool to support them in a fast and accurate medical decision.

References

1. **Altman, D.G.**, Practical Statistics for medical Research. Chapman and Hall (1991)
2. **Bishop, C.**, Neural networks for pattern recognition (2nd Ed.). Oxford University Press (1995)
3. **Duda, R., Hart, P., Storck, D.**, Pattern Classification (2nd Ed.). Wiley Interscience, New York (2000)
4. **Gorunescu, F.**, Data Mining - Concepte, modele si tehnici. Ed Albastra-Microinformatica, Cluj-Napoca(2006)
5. **Gorunescu, M.**, Clasificare si prognoza, Seria "Computer Science" (Centrul de Cercetare in Inteligenta Artificiala). Editura Universitaria Craiova(2007)

Morphogenetic Pattern Generation Using an Ultra-Discrete Reaction-Diffusion System

Costin-Radu BOLDEA

Faculty of Mathematics and Computer Science,
Department of Computer Science,
University of Craiova, Romania,
cboldea@inf.ucv.ro

Abstract. The paper present an alternative approach to the determination of a cellular automaton simulator for the stable patterns generation and the evolution of the mammals skin models. The proposed simulator uses the ultra-discrete version of an important reaction-diffusion systems arising from the bio-mathematics domain: the Thomas-Murray reaction-diffusion system.
Keywords: Turing reaction-diffusion system, cellular automata, modeling biological patterns
Math. Subject Classification 2000: 92B05; 68W20

1 Introduction

The Reaction-Diffusion modeling and simulations, particularly in a sense of chemical computation or in the domain of biophysics, becomes a hot topic of computer science, physics and chemistry.

The Reaction Diffusion process is one in which a number of substances or morphogens can diffuse over a surface and react with each other to produce stable patterns on the surface. This mechanism has been studied by biologists as well as mathematicians as the system, which consists of a series of non-linear coupled partial differential equations, is thought to be responsible for pattern formation in nature, such as the patterns on an animal's coat. Work on reaction diffusion began when Alan Turing proposed a mechanism which could explain the development of animal embryos and the fact that they can be self-organizing.

The pattern generating reaction-diffusion systems are governed by a set of coupled partial differential equations as seen above. The problem with simulating a system such as this is that the equations are continuous. This means that the equation represents the entire solution space of the system which has an infinite number of values for time and space. The continuous nature of the system makes it very difficult to simulate on a computer which by nature cannot handle continuous systems.

We must therefore discretize these equations so that we can simulate the system on a grid or lattice which can be used in a computer. This discretization

is done on a lattice where the simulation takes place. The simulation method used is that of cellular automata.

In 2007 I proposed a very simple CA model that can simulate the zebra skin patterns formation (see [1]). This model was constructed beginning from the Yang discretization of Turing's system, following the approach of Gravan C. and Lahoz-Beltra R.([2]). The purpose of the present paper approach is to propose an alternative approach to construct a pattern generator simulation for mammals skin models (leopard, tiger or giraffe), obtained by a direct ultra-discretization procedure applied to the Thomas-Murray Reaction-Diffusion System, and to verify if the ultra-discretization method preserves the reaction-diffusion phenomena in this particular case.

2 The Thomas-Murray system (TMS)

In 1952 Turing ([8]) published a paper suggesting that, under certain conditions, some chemicals can react and diffuse from an initially nearly homogeneous state to create spatial stable patterns as a consequence of the breakdown of symmetry and homogeneity. Turing proposed that the temporal variation of the concentrations of two different chemicals, named by Turing as morphogens (the activator morphogen $\underline{u}$ and the inhibitor morphogen $\underline{v}$), both diffusible but at different rates (d_a and d_i), can create patterns on an initially homogeneous tissue by reacting in accordance with some nonlinear functions f and g:

The general form of Turing's Reaction-Diffusion systems is

$$\begin{cases} \frac{\partial u}{\partial t} = d_a \nabla u + f(u,v) \\ \frac{\partial v}{\partial t} = d_i \nabla v + g(u,v) \end{cases} \tag{1}$$

where d_a, d_i are diffusion constants, x, y are the spatial coordinates and u, v are functions of x, y and t.

In Turing's model, the activator morphogen u activates the production of itself and the production of the inhibitor v , whereas v inhibits the production of itself and decreases the activator u production.

More that twenty years later, Thomas ([6]) proposed a model of enzyme reaction, based on the Turing's Reaction Diffusion system. This particular system was largely studied by Murray ([4], [5] Chap.15) as the possible mechanism responsible for laying down most of the mammals coat spacing patterns. The model assumes the animal skin is formed by a uniform distribution of pigmented cells (black, state 1), differentiated by melanocytes, and undifferentiated cells (white, state 0). Melanocytes produce the activator morphogen u which stimulates the transition from state 0 to 1 of nearby undifferentiated cells, as well as the inhibitor morphogen v promoting the opposite transition, thus from state 1 to 0, for nearby differentiated cells. The time evolution of the concentrations of activator/inhibitor morphogenes is determined by the Thomas-Murray system (TMS):

$$\begin{cases} \frac{\partial u}{\partial t} = \nabla u + \gamma[a - u - h(u,v)] \\ \frac{\partial v}{\partial t} = d\nabla v + \gamma[\alpha(b - v) - h(u,v)] \\ h(u,v) = \frac{\rho \cdot uv}{1+u+Kv^2} \end{cases} \tag{2}$$

where *a,b,* α *,* γ and K are positive parameters, the ratio of diffusion coefficients $d = d_i/d_a$ is greater that one (normally take values greater that 12) and the scale factor γ is a measure of the domain and control only the dimension of patterns. (In the next we consider the parameters *a=b=* γ *=1.*)

In order to investigate the type of spatial pattern generated by the full nonlinear system (2) and to construct a pattern generator simulator, we must discretize this system until to a simulator cellular automaton witch simulate the nonlinear behavior of the (TMS).

A cellular automaton (Wolfram [9], Weimar [10]) provide a framework for a large class of discrete models with homogeneous interactions, characterized by the following properties:

- They consist of a regular discrete lattice of cells.
- The evolution takes place in discrete time steps.
- Each cell is characterized by a state taken from a finite set of states.
- Each cell evolves according to the same rule which depends only on the state of the cell and a finite number of neighboring cells.
- The neighborhood relation is local and uniform.

Cellular automata (CA) have been widely adopted in the sciences as simple but powerful models of the real world because the complex patterns produced by their long-time behaviors can mimic observations with tremendous accuracy.

In the particular case of mammalian coat patterns generation, a simple cellular automata model of Turing's system was succefully used by Gravan C. and Lahoz-Beltra R. ([2]) to simulate the patterns formation on the zebra's skin. In 2007 Boldea C. proposed (in [1]) a similar CA model, obtained using a genetic algorithm aproach. But these models are only empirical experiences, witch do not reflect correctly the nonlinear phenomena characterizing the Thomas Murray system. For this reason, we choused an ultra-discretization procedure that preserves the continuous nonlinear behavior of the system (2).

3 Discrete and ultra-discrete reaction diffusion system derived from the Thomas-Murray system

Cellular automata (CA) have been widely adopted in the sciences as simple but powerful models of the real world because the complex pat-terns produced by their long-time behaviors can mimic observations with tremendous accuracy ([9],[10]). However, the lack of mathematical tools makes prediction difficult in CA models. , think was by The work of Tokihiro et al. [7] developed a method to ultra-discretize continuous systems, based on a limit passing procedure and

confirmed that there are integrable, predictable, Cellular Automata obtained by this method.

First step to apply the method of Tokihiro et al. in order to obtain a discrete valuated, discrete time, discrete space variables system (ultra-discrete system) from equations (2), is to pass by a classical discrete version of this equation.

The discrete versions of the above (TM) system are obtained by replacing the time derivative

$$\frac{\partial u(x,y,t)}{\partial t} \to \frac{u(x,y,t+\Delta t)-u(x,y,t)}{\Delta t} \to u(x,y,t+1)-u(x,y,t) \tag{3}$$

and the space derivatives by

$$\frac{\partial u(x,y,t)}{\partial x} \to u(x+1,y,t)-u(x,y,t) \tag{4}$$

$$\frac{\partial u(x,y,t)}{\partial y} \to u(x,y+1,t)-u(x,y,t) \tag{5}$$

$$\begin{array}{l}\Delta u \to u(x+1,y,t)+u(x-1,y,t)+\\ +u(x,y+1,t)+u(x,y-1,t)-4u(x,t)\end{array} \tag{6}$$

By plugging these discretizations into the system (2), one obtain

$$\left\{\begin{array}{l} u(x,t+1)=u(x,y,t)+[u(x+1,y,t)+u(x-1,y,t)+\\ u(x,y+1,t)+u(x,y-1,t)-4u(x,t)]-\\ -[u(x,y,v)-1+h(u(x,y,t),v(x,y,t))]\\ \\ v(x,t+1)=v(x,y,t)+d[v(x+1,y,t)+v(x-1,y,t)+\\ +v(x,y+1,t)+v(x,y-1,t)-4v(x,t)]-\\ -[\alpha(u(x,y,t)-1)+h(u(x,y,t),v(x,y,t))]\end{array}\right. \tag{7}$$

Next, we apply the ultra-discretization procedure on the system (7): given a rational function in u and v, the ultra-discretization method requires that we introduce new variables U and V defined by $u = exp(U/å)$, $v = exp(V/å)$. After we take the limit å0+ of the equations using the identities:

$$\lim_{\varepsilon\to 0+} \varepsilon \log\left[e^{A/\varepsilon}+e^{B/\varepsilon}\right] = \max(A,B)$$

$$\lim_{\varepsilon\to 0+} \varepsilon \log\left[e^{A/\varepsilon}-e^{B/\varepsilon}\right] = Alt\max(A,B)$$

where

$$Alt\max(A,B) = \left\{\begin{array}{c} A,\ if\ A>B\\ 0,\ if\ A=B\\ -B,\ if\ A<B\end{array}\right.$$

is the alternate maximum function. Ultra-discrete equations are naturally posed on the max-plus semi-ring (defined in [3]).

The ultra-discretisation procedure transform the diffusion part of eq. (7)

$$\begin{cases} u(x,t+1) = u(x,y,t) + [u(x+1,y,t) + u(x-1,y,t) + \\ u(x,y+1,t) + u(x,y-1,t) - 4u(x,t)] \\ \\ v(x,t+1) = v(x,y,t) + d \cdot [v(x+1,y,t) + v(x-1,y,t) + \\ +v(x,y+1,t) + v(x,y-1,t) - 4v(x,t)] \end{cases} \tag{8}$$

into

$$\begin{cases} u_{x,y}^{t+1} = \max[u_{x,y}^t, Alt\max(u_{x+1,y}^t, u_{x,y}^t), Alt\max(u_{x-1,y}^t, u_{x,y}^t), \\ Alt\max(u_{x,y+1}^t, u_{x,y}^t), Alt\max(u_{x,y-1}^t, u_{x,y}^t)\} \\ v_x^{t+1} = \max[v_{x,y}^t, Alt\max(v_{x+1,y}^t, v_{x,y}^t) + \delta, Alt\max(v_{x-1,y}^t, v_{x,y}^t) \\ +\delta, Alt\max(v_{x,y+1}^t, v_{x,y}^t) + \delta, Alt\max(v_{x,y-1}^t, v_{x,y}^t) + \delta\} \end{cases} \tag{9}$$

where δ =[ln(d)] is a diffusion ratio parameter. The reaction part become:

$$\begin{cases} h(u,v) \rightarrow u_{x,y}^t + v_{x,y}^t - \\ \quad - Alt\max(\max(u_{x,y}^t, 2v_{x,y}^t + k), 1) + R \equiv H(u_{x,y}^t, v_{x,y}^t) \\ (u-1) + H(u,v) \rightarrow \max[Alt\max(u_{x,y}^t, 1), H(u_{x,y}^t, v_{x,y}^t)] \equiv \\ \quad \equiv \Psi_1(u_{x,y}^t, v_{x,y}^t) \\ \alpha(v-1) + h(u,v) \rightarrow \max[Alt\max(v_{x,y}^t, 1) + A, H(u_{x,y}^t, v_{x,y}^t)] \equiv \\ \quad \equiv \Psi_2(u_{x,y}^t, v_{x,y}^t) \end{cases} \tag{10}$$

where *R,A,k* are constants. (We used the operations $uv \rightarrow U+V$, $cv \rightarrow V+k$and $u+v \rightarrow max(U,V)$ in the ultra-discrete limit, with *k=⌊ln(c)⌋* for any positive constant *c*.)

The ultra-discrete version of Thomas-Murray system (2) will be:

$$\begin{cases} u_{x,y}^{t+1} = \max(u_{x,y}^t, u_{x+1,y}^t, u_{x-1,y}^t, u_{x,y+1}^t, u_{x,y-1}^t) \\ \quad - \Psi_1(u_{x,y}^t, v_{x,y}^t) \\ v_{x,y}^{t+1} = \max(u_{x,y}^t, u_{x+1,y}^t + \delta, u_{x-1,y}^t + \delta, u_{x,y+1}^t \\ \quad + \delta, u_{x,y-1}^t + \delta) - \Psi_2(u_{x,y}^t, v_{x,y}^t) \end{cases} \tag{11}$$

Note that a supplementary reset condition must be introduced in order to simulate the real physical phenomena: ux,yt+1 $u_{x,y}^t \leftarrow 0$, $v_{x,y}^t \leftarrow 0$ if the calculated values from (11) are negatives, corresponding to a negative concentration of a morphogene.

4 Simulation results and conclusions

The system (11) can be used to define a non-standard cellular bi-valuated automaton, according to each cell a pair of values $C_{x,y}(t) = (u_{x,y}^t, v_{x,y}^t)$, where

$x, y, t \in IN$, and $u^t_{x,y}, v^t_{x,y}$ are real functions. The pattern generator simulator is simply defined by

$$P^t_{i,j} = \begin{cases} 1, & if \ u^t_{i,j} > v^t_{i,j} \\ 0, & if \ u^t_{i,j} \leq v^t_{i,j} \end{cases} \tag{12}$$

for $i, j, t \in \Omega \subset IN$. Conventionally, we consider that *" 1"*is associated with colored cells, and *"0"* correspond to the "white" cells.

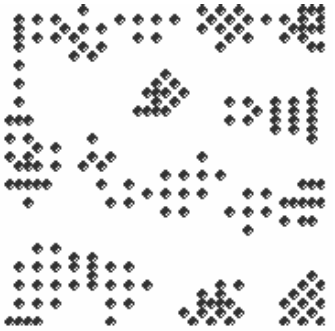

Fig. 1. Initial configuration of the cell automaton. Only the presence of activator is represented.

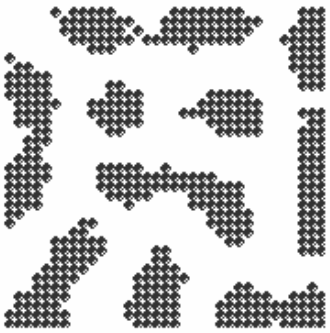

Fig. 2. Example of a generated pattern for δ =R=A=0

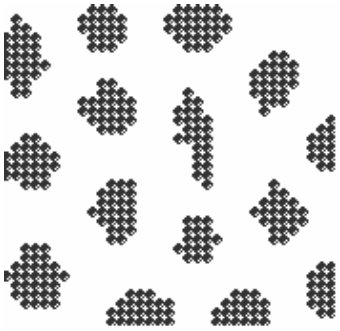

Fig. 3. Example of a generated pattern for δ =R=1

For the purpose of testing the reaction diffusion simulation we ran some simulations with different parameters to see what kinds of patterns would be produced. In the simulations we used two lattices for the initial values of $(u^t_{x,y}, v^t_{x,y})$ with equal concentrations of each morphogene. Each cell take the initial value $u^0_{x,y} \leftarrow 1$ with a certain probability p_a, and the initial value $v^0_{x,y} \leftarrow 1$, with a certain probability p_i, in rest any initial state become null (see Fig.1).

The more that 50 numerical experiments realized by the authors permit to remark that the CA model described by eqn. (11) is always superior limited by a maximum value and the generated patterns become stable after a certain time.

The Figures 2 and 3 shows the results of a reaction diffusion simulation which we carried out using a fluctuation of the parameters of the model.

In conclusion, the simulation results are consistent with the general picture of pattern modeling and simulation based on Turing's reaction-diffusion scheme. The cell automaton P from (12) present a similar nonlinear behavior like the full continuous Thomas-Murray system (2).

References

[1] **Boldea C.R.**, Evolving Morphogenetic Patterns with a Genetic Algorithm,Research Notes in Artificial Intelligence and Digital Communications, Vol. 107, 2007, 127-132.

[2] **Gravan C. and Lahoz-Beltra R.**, Evolving Morphogenetic Fields In The Zebra Skin Pattern Based On Turing'S Morphogen Hypothesis - Int. J. Appl. Math. Comput. Sci., Vol. 14, No. 3, 2004, pp. 351–361.

[3] **Joshi N. and Ormerod C**, The General Theory Of Linear Difference Equations Over The Max-Plus Semi-Ring, Studies in Applied Mathematics, Volume 118, Number 1, January 2007, pp. 85-97.

[4] **Murray J.D**, On Pattern Formation Mechanisms for Lepidopteron Wing Patterns and Mammalian coat markings, Philosophical Translations of the Royal Society B, Vol. 295, 1981, pp. 473-496.
[5] **Murray J.D**, Mathematical Biology, New York, Springer-Verlag, 1989.
[6] **Thomas D.**, Artificial enzyme membranes, transport, memory and oscillatory phenomena, in D. Thomas and J.-P. Kerneves (eds.) Analysis and Control of Immobilized Enzyme Systems, Berlin, Springer, 1975, pp. 223-254.
[7] **Tokihiro T., Takahashi D., Matsukidaira J. and Satsuma J.**, From soliton equations to integrable cellular automata through a limiting procedure Phys. Rev. Lett. 76, 1996 , pp. 3247–3250.
[8] **Turing A.**, The Chemical Basis of Morpho-genesis, Philosophical Translations of the Royal Society B, Vol. 237, 1952, pp. 37-72.
[9] **Wolfram S.**, New constructions in cellular automata Proceedings of the Conference Santa Fe NM 1998, New York, Oxford University Press, 2003.
[10] **Weimar J. R.**, Simulation with Cellular Automata, Braunschweig, 1996.