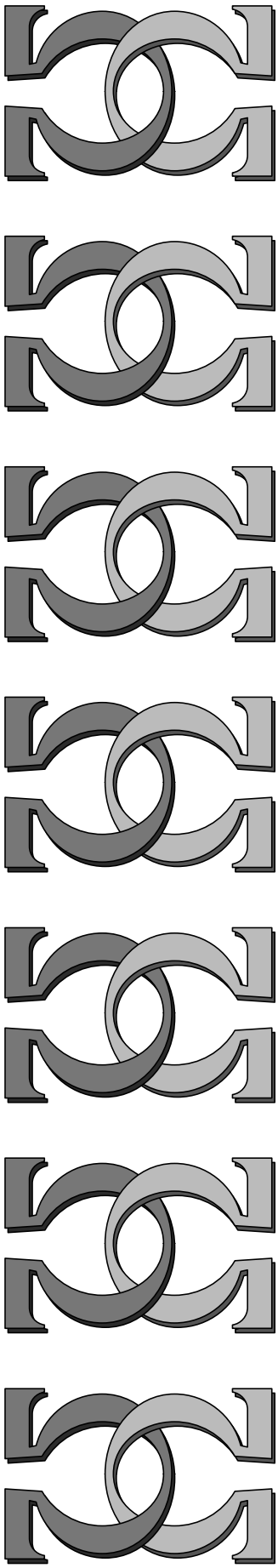


CDMTCS Research Report Series

Supplemental Papers for the 2nd International Conference on Unconventional Models of Computation, UMC'2K

I. Antoniou
Solvay Institutes, Brussels, Belgium

C. S. Calude and M. J. Dinneen
University of Auckland, New Zealand

(Editors)

CDMTCS-147
November 2000

Centre for Discrete Mathematics and
Theoretical Computer Science

Editors' Notes

These are additional papers of talks that were presented at the *Second International Conference on Unconventional Models of Computation (UMC'2K).* UMC'2K, organized by the Centre for Discrete Mathematics and Theoretical Computer Science, the International Solvay Institutes for Physics and Chemistry and the Vrije Universiteit Brussel Theoretical Physics Division was held at Solvay Institutes during December 13–16, 2000.

The conference encompasses all areas of unconventional computation, especially quantum computing, DNA-based computation, evolutionary algorithms and other proposals for computation models that go beyond the Turing model. Additional refereed and invited papers of UMC'2K appear in the following proceedings:

> I. Antoniou, C. S. Calude, M. J. Dinneen, editors. *Unconventional Models of Computation, UMC'2K*, Springer-Verlag, London, December 2000, XI + 301 pages.

Multi-Sample Stochastic Filtering of Digital Images of Skin Micro-Structure

P.G. Akishin[1], E.P. Akishina[1], P. Akritas[2,3], I. Antoniou[2,3], J. Ioannovich[4] and V.V. Ivanov[1,2]

[1] Joint Institute for Nuclear Research, Dubna, Russia
[2] International Solvay Institutes for Physics and Chemistry, Brussels, Belgium
[3] Theoretische Natuurkunde, Free University of Brussels, Brussels, Belgium
[4] University of Athens and General State Hospital of Athens, Department of Plastic Surgery and Microsurgery, Mesogion 154, Greece - 115 27 Athens

Abstract. We generalize our stochastic filter [1], in order to suppress the noise due to chaotic scattering and improve the quality of the optical tomography image (OCT). The main idea of the new filter is based on the use of several images obtained by the OCT probing head from the same location on the skin surface. The efficiency of the multi-sample filtering algorithm is demonstrated on test examples and on real OCT images.

1 Introduction

A new application of optical coherence tomography (OCT) for the analysis of skin micro-structure in real time has been recently developed [2]. At present a compact system for non-invasive imaging of human tissues in vivo has been constructed.

The first OCT images have shown distortions mainly due to the chaotic scattering of light by the skin tissue. In order to increase the reliability of diagnosis we developed a new stochastic filter and tested it on model data and on real OCT images [1]. This filter uses the information obtained from one digital image only. However, the optical system of the OCT device permits to obtain several digital images from the same location of human tissue. This provides additional possibilities for the improvement of OCT images.

The objective of this paper is to use these possibilities and to improve the images. After a brief description of the experimental device to obtain the biotissues images (section 2), we generalize the stochastic filter developed in [1] to several images (section 3). We present (section 4) the results of the multi-sample filtering algorithm application on model data and on real OCT images.

2 Scheme of Optical Coherence Tomography

The scheme of a fast-scanning device for the optical coherence tomography of biotissues is shown in Fig. 1.

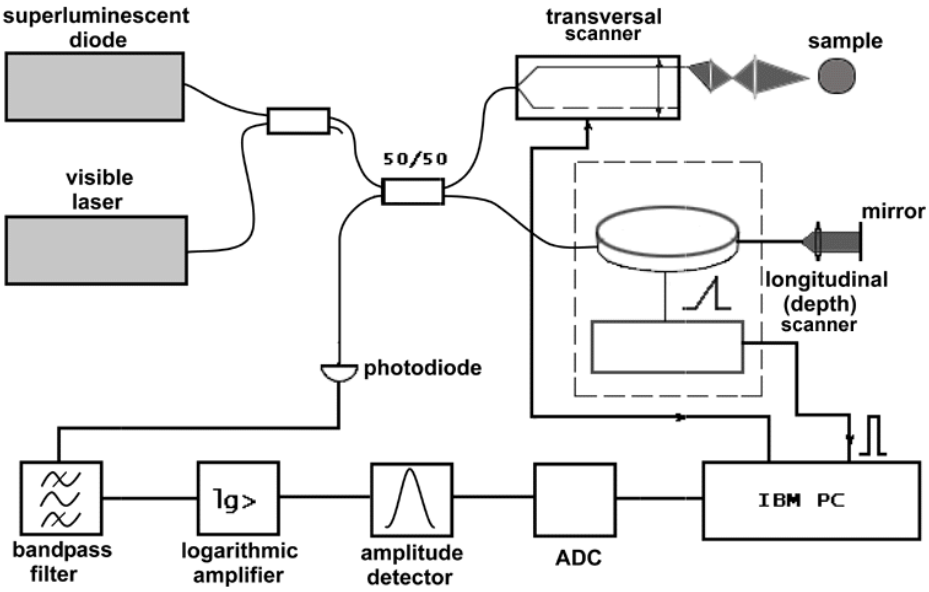


Fig. 1. Scheme of optical coherence tomography

The experimental device, for obtaining the tissue images, is based on the unimodal fiber Michelson interferometer with the semiconductor superluminescence diode: $\lambda = 0.83\,nm, \Delta\lambda = 30\,nm$, the intensity power on an object surface is 30 μW [3]. The sample to be analyzed is placed in one of interferometer arms as the reflecting object. The optical length of another (support) arm is scanned with the fixed linear speed V. The interference signal at the Doppler frequency $f = 2V/\lambda$ is proportional to the reflecting coefficient of the non-reflected component by the optical non-homogeneity inside the sample. The position of this non-homogeneity is determined by equality of optical paths passing by interference light rays. The spatial resolution in the longitudinal direction (inside the sample depth), probably, coincides with the coherence length ($\sim 10\,\mu m$). The scanning in the transfer direction is realized by a mechanical-optical system, which transfers the focused spot of the probing radiation along the surface of sample. The transfer resolution of the location is determined by the radius α of the focal spot which is usually less than 20μm. The radius α is chosen from the condition that the ratio $2n\pi a^2/\lambda$ (where $n \approx 1.35$ is the typical reflection index for biotissues) must not be less than the longitudinal size of the probing region. For more detailed resolution of regions of interest one have to decrease a together with corresponding decreasing the longitudinal scanning range. The obtaining interference signal

passes through stages of analogue and digital processing. The sequential visualization of signal permits to obtain in real time the two-dimensional images – optical tomogrammes.

The portable OCT device consists of an opto-electronic unit with an outer flexible optical cable terminated by a probe head. It is connected with a standard cable to the power supply and with an interface cable, to a PC that is provided with a data acquisition card. The image recording is automated and controlled from the PC keyboard. A specially developed software for control of the OCT device, processing and presenting images in DOS or Windows environments. The OCT image with up to 200×200 (depth $\times$ one lateral coordinate) pixel size with 100–150 μm spatial resolution is recorded for 1–5 sec and monitored in real time. The transverse two-dimensional scanning system provides the probing beam motion along an arbitrary trajectory in the surface, hence allowing for the three-dimensional image acquisition [4].

3 Filtering algorithm

Following [1], we denote $\zeta(x)$ the intensity of the original uncorrupted image at the point $x = (x_1, x_2)$. As a result of chaotic scattering of light by the skin tissue, a contaminated signal ξ is measured. The effect of chaotic scattering can be represented by a stochastic field η (see below), which we have to filter out from the measured signal ξ. The filter gives an estimate $\tilde{\xi}(x)$ for the original signal $\zeta(x)$ obtained from the measured data $\{\xi^\mu(x), \mu = 1, ..., M\}$. Here M is the number of OCT images obtained from the same location of the human tissue. In [1] we considered a simple case of one image, $M = 1$.

On the basis of our experience with the probabilistic description of chaotic systems we propose to assess the influence of chaotic scattering on the images through a simple stochastic field $\eta(x)$ satisfying the following conditions [1]: 1) η is additive at each lattice point, 2) η is identically distributed, 3) η has zero mean value, 4) η has variance σ^2, 5) no correlations exist between the different points in the lattice.

From here we get:

$$\xi^\mu(x) = \zeta(x) + \eta^\mu(x), \qquad \mu = 1, ..., M.$$

To construct the filtering algorithm we use a circular working window centered at $c = (c_1, c_2)$. The nodes within the working window are represented by the set: $\{x \,|\, x^\nu = z^\nu + c,\ \nu = 1, ..., N\}$. The index $\nu = 1, ..., N$ labels the points of the selected window and $z^\nu = (z_1^\nu, z_2^\nu)$ are the coordinates of points within the window with respect to the center $c = (c_1, c_2)$.

The estimate $\tilde{\zeta}(c)$ from the measured signals $\{\xi^\mu(x)), \mu = 1, ..., M\}$ is obtained from the following linear filter:

$$\tilde{\zeta}(c) = \sum_{\mu=1}^{M} \sum_{\nu=1}^{N} W^\mu(z^\nu) \xi^\mu(z^\nu + c), \tag{1}$$

where $W^{\mu}(z^{\nu})$ are the weights to be specified by the probabilistic assessment $\eta(x)$ of the effect of chaotic scattering of light by the skin tissue on the measured OCT image.

In order to specify the weights we shall consider the limit case, when only uncorrupted signals are fed to the filter.

We approximate the original signals within the selected window in a finite set of basis functions of two variables with respect to the center c:

$$\bar{\zeta}(x) = \sum_{k=1}^{N_0} A_k g_k(x-c) = \sum_{k=1}^{N_0} A_k g_k(z).$$

For example, in the case of polynomials basis

$$\bar{\zeta}(x_1, x_2) = \sum_{0 \le n+m \le K} A_{nm} (x_1 - c_1)^n (x_2 - c_2)^m. \tag{2}$$

If the signal is uncorrupted, the estimation $\tilde{\zeta}$ should be identical to the original one, i.e. $\zeta = \tilde{\zeta}$.

Thus, for signals identical to the basis functions $\zeta(x) = g_k(x-c)$ we must have:

$$g_k(0) = \sum_{\mu=1}^{M} \sum_{\nu=1}^{N} W^{\mu}(z^{\nu}) g_k(z^{\nu}), \quad k = \overline{1, N_0}, \tag{3}$$

or

$$g_k(0) = (\sum_{\mu=1}^{M} \mathbf{W}^{\mu}) \cdot \mathbf{g_k}, \quad k = \overline{1, N_0}. \tag{4}$$

where

$$\mathbf{W}^{\mu} = (W^{\mu}(z^{1}), ..., W^{\mu}(z^{N})), \quad \mathbf{g_k} = (g_k(z^{1}), ..., g_k(z^{N})). \tag{5}$$

Therefore, for signals ζ, that are linear combinations of functions $\{g_k(z)\}$, the filter should have weights which minimize the dispersion associated with an input:

$$D_M = E\left[\sum_{\mu=1}^{M} \sum_{\nu=1}^{N} W^{\mu}(z^{\nu})\, \eta^{\mu}(z^{\nu} + c)\right]^2 =$$

$$\sum_{\mu=1}^{M} \sum_{\mu'=1}^{M} \sum_{\nu=1}^{N} \sum_{\nu'=1}^{N} W^{\mu}(z^{\nu}) W^{\mu'}(z^{\nu'}) E\left[\eta^{\mu}(z^{\nu} + c) \eta^{\mu'}(z^{\nu'} + c)\right] =$$

$$\sum_{\mu=1}^{M} \sum_{\mu'=1}^{M} \sum_{\nu=1}^{N} \sum_{\nu'=1}^{N} W^{\mu}(z^{\nu}) W^{\mu'}(z^{\nu'}) \sigma^2 \delta(\nu - \nu') \delta(\mu - \mu') = \sum_{\mu=1}^{M} ||\mathbf{W}^{\mu}||^2 \sigma^2. \tag{6}$$

In order to find out the weights $W^{\mu}(z^{\nu})$, we minimize the quadratic form:

$$\sum_{\mu=1}^{M} ||\mathbf{W}^{\mu}||^2 = \mathbf{W_0} \cdot \mathbf{W_0} = ||\mathbf{W_0}||^2, \qquad \mathbf{W_0} = (\mathbf{W^1}, ..., \mathbf{W^M}),$$

respecting condition (4).

Using the Cauchy-Bunjakovsky inequality [5] we have:

$$\frac{||\sum_{\mu=1}^{M}\mathbf{W}^{\mu}||^2}{M} \leq \sum_{\mu=1}^{M}||\mathbf{W}^{\mu}||^2. \tag{7}$$

The equality in (7) takes place only for the case:

$$\mathbf{W}^{\mu} = \frac{\sum_{\mu'=1}^{M}\mathbf{W}^{\mu'}}{M}. \tag{8}$$

Let us decompose the vector $\sum_{\mu'=1}^{M}\mathbf{W}^{\mu'}$ in the form:

$$\sum_{\mu'=1}^{M}\mathbf{W}^{\mu'} = \mathbf{U} + \sum_{k=1}^{N_0}\beta_k\mathbf{g_k},$$

where the vector $\mathbf{U}$ is orthogonal to all vectors of the set $\{\mathbf{g_k}\}$. Then, condition (4) can be rewritten as follows:

$$\sum_{k'=1}^{N_0}\beta_{k'}(\mathbf{g_{k'}}\cdot\mathbf{g_k}) = g_k(0), \quad k = 1,...,N_0. \tag{9}$$

Taking into account (7) for the functional $||\mathbf{W_0}||^2$ we get:

$$||\mathbf{W_0}||^2 \geq \frac{||\mathbf{U}||^2 + ||\sum_{i=1}^{N_0}\beta_i\mathbf{g_i}||^2}{M}. \tag{10}$$

This means that the minimum of the functional $||\mathbf{W_0}||^2$ is reached at $\mathbf{U} = 0$. In this case:

$$\mathbf{W}^{\mu} = \frac{\sum_{i=1}^{N_0}\beta_i\mathbf{g_i}}{M} \quad for \quad \mu = 1,...,M. \tag{11}$$

Therefore, all vectors $\{\mathbf{W}^{\mu}\}$ are identical.

As a result, (4) becomes:

$$g_k(0) = M\mathbf{W_M}\cdot\mathbf{g_k}, \quad k = 1,...,N_0. \tag{12}$$

The above described algorithm is oriented to filtering of smooth images. In this case, as coefficients β_i in (9) do not depend on the central point of the working window, the weights $W^{\mu}(z_{\nu}) = W(z_{\nu})$ can be calculated only once.

When the image has sharp changes of intensity, we apply the method developed in [1]. The intensity histogram is constructed on the basis of the points falling inside the working window and using all measured data $\{\xi^{\mu}(x), \mu = 1,...,M\}$. This histogram must have at least two clusters. The summation in Eq. 1 is taken only over those points which belong to the cluster Ω

containing the central point of the working window. Equation 4 will assume the following form:

$$g_k(0) = M \sum_{z^\nu \in \Omega} W(z^\nu) g_k(z^\nu), \quad k = \overline{1, N_0}. \tag{13}$$

The set of the corresponding weights W are calculated using system (9), for each case. A similar procedure is also used for filtering the points situated near boundaries of image.

Let $\mathbf{W_1}$ be the solution of Eq. 12 for the case $M = 1$. Then, for an arbitrary M we have:

$$\mathbf{W_M} = \frac{\mathbf{W_1}}{M}.$$

Thus, the dispersion D_M for the M image case can be expressed through the dispersion D_1 as follows:

$$D_M = \frac{D_1}{M}. \tag{14}$$

This means that the mean squared error of the filtering procedure decreases with increasing of the number M as $\frac{1}{\sqrt{M}}$.

4 Numerical results

We present here the numerical results on the application of the multi-sample stochastic filtering algorithm to model images and real OCT data.

Figure 2 shows the original model image: 512×512 points, where we see sharp changes of intensity. In order to estimate the efficiency of the filtering algorithm based on the simultaneous use of several images, we prepared M noisy images by adding Gaussian noise ν with mean value $E(\nu) = 0$ and the standard deviation σ. Figure 3 shows one of such images corrupted by noise corresponding to $\sigma = 5$.

The efficiency of the filtering algorithm has been estimated using the following measures:

$$d_1 = \max_{i=1}^{N_t} |\zeta_i - \tilde{\zeta}_i|, \quad d_2 = \sqrt{\sum_{i=1}^{N_t} |\zeta_i - \tilde{\zeta}_i|^2 / N_t}.$$

Here N_t is the total number of points in the image, $\zeta_i = \zeta(c_i)$ is the original signal and $\tilde{\zeta}_i = \tilde{\zeta}(c_i)$ is the filtered signal at each point c_i. The values of ζ_i varied between -175 and 372.

Figure 4 presents the dependence of the d_1 measure against the number M ($M = 1, 4, 9, 16, 25$) and against the radius R_w ($R_w = 5, 7, 9$) of the working window. Here $\sigma = 1$ and the maximal power of the approximating functions $K = 2$. Figure 5 shows the similar results for the d_2 measure.

Figures 6 and 7 demonstrate the similar results for the case $\sigma = 5$.

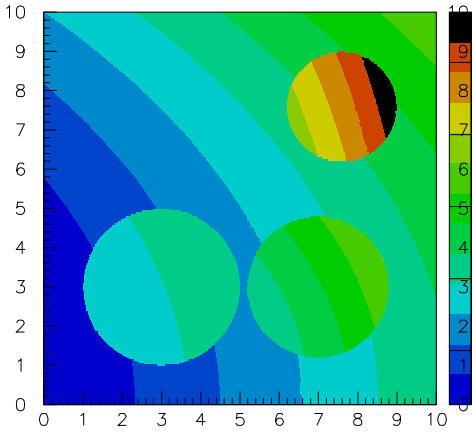


Fig. 2. Original model image: 512×512 points

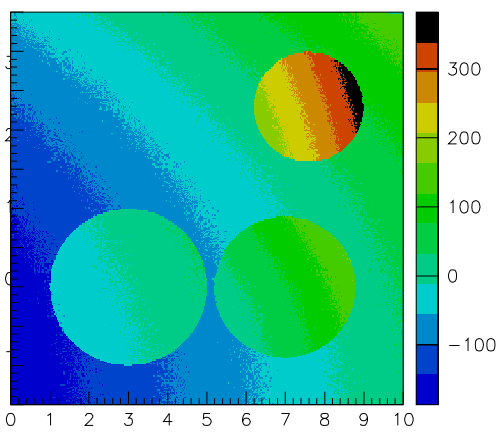


Fig. 3. Model image corrupted by Gaussian noise: $\sigma = 5$

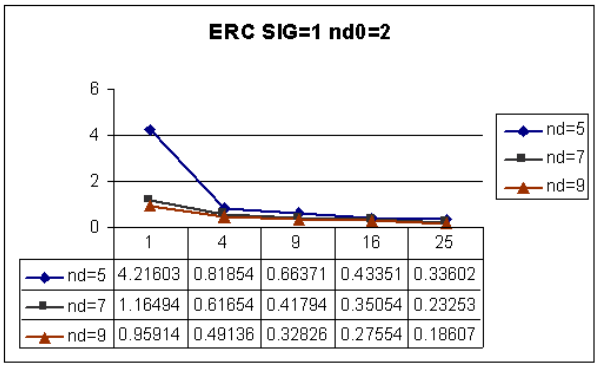
One can see from figures 4–7 that while M changes from 1 to 25, the values of d_1 and d_2 decreases more than 30 times. We also see that the minimal values of d_1 and d_2 correspond to the working window with the maximal radius $R_w = 9$.

Figures 5 and 7 show that the value d_2^2 changes with the number M of samples, involved in the filtering procedure, as it was predicted by Eq. 14. It follows from the fact that for images with large number of points the measure d_2^2 is very close to the dispersion of noise. This interesting behaviour demonstrates the d_1^2 measure, which also follows Eq. 14. The comparison of values $d_1^2 = d_1^2(M,\sigma)$ and $d_2^2 = d_2^2(M,\sigma)$ for different values of σ, shows that these measures are proportional to σ. This result is also in the agreement with formula (14).

The comparison of results obtained for different values of the maximal power K has shown that there is no any noticeable gain while $K = 2$ is changed to $K = 3$.

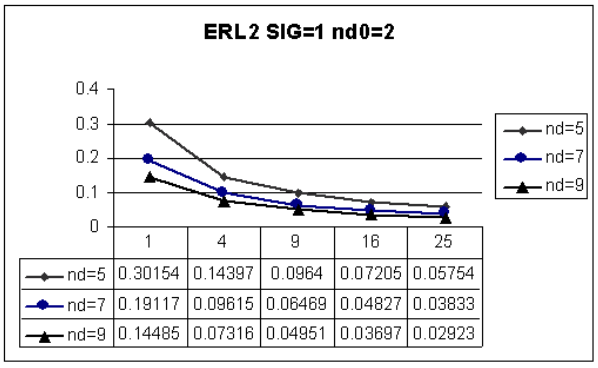
In Fig. 8 we present the result of application of the proposed filtering algorithm to model images. We consider the case: $\sigma = 5$, $M = 25$, $R_w = 9$ and $L = 2$. Figure 8 demonstrates that the filtering procedure restores the original image with a high quality.

The results of application of the multi-sample algorithm to real OCT images are given in Figures 9–10. We simultaneously used five OCT images. One of these images is presented in Fig. 9: 250×250 points.



	1	4	9	16	25
nd=5	4.21603	0.81854	0.66371	0.43351	0.33602
nd=7	1.16494	0.61654	0.41794	0.35054	0.23253
nd=9	0.95914	0.49136	0.32826	0.27554	0.18607

Fig. 4. Dependences of d_1 against the samples number M and against the radius of the working window R_w: $\sigma = 1$ and $K = 2$



	1	4	9	16	25
nd=5	0.30154	0.14397	0.0964	0.07205	0.05754
nd=7	0.19117	0.09615	0.06469	0.04827	0.03833
nd=9	0.14485	0.07316	0.04951	0.03697	0.02923

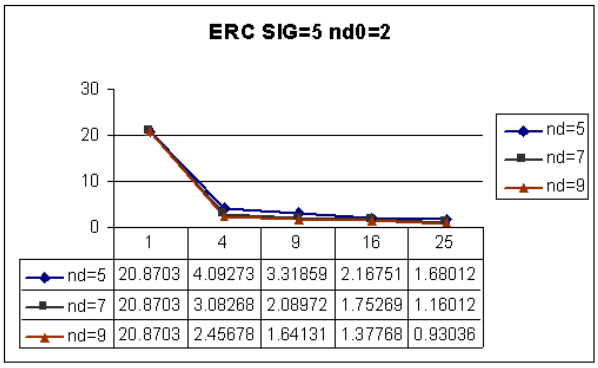
Fig. 5. Dependences of d_2 against the samples number M and against the radius of the working window R_w: $\sigma = 1$ and $K = 2$

The circle window with $R_w = 4$ and polynomials with total power $L = 2$ were used for real images filtering. Figure 10 shows the result of application of the multi-sample filtering procedure.

5 Conclusion

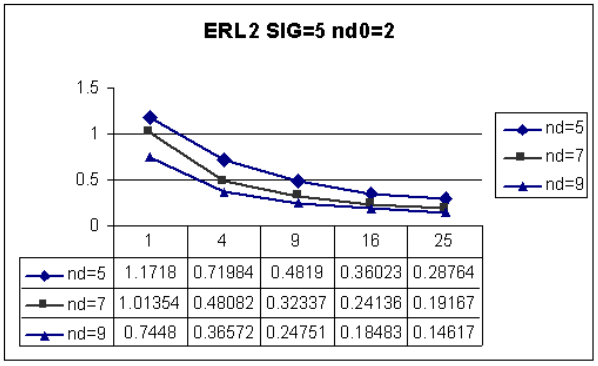
We have generalized the filtering algorithm developed in [1] in order to simultaneously use in the filtering procedure several OCT images, which can be obtained from the same location of the human tissue. The new filter was tested on model data. These tests demonstrated its efficiency.

The application of new filter to real OCT images show that the filtering procedure restores the characteristic contours and details of the original image. This provides more reliable analysis of images of skin micro-structure and medical diagnosis compared to one image. More detailed study with real



	1	4	9	16	25
nd=5	20.8703	4.09273	3.31859	2.16751	1.68012
nd=7	20.8703	3.08268	2.08972	1.75269	1.16012
nd=9	20.8703	2.45678	1.64131	1.37768	0.93036

Fig. 6. Dependences of d_1 against the samples number M and against the radius of the working window R_w: $\sigma = 5$ and $K = 2$



	1	4	9	16	25
nd=5	1.1718	0.71984	0.4819	0.36023	0.28764
nd=7	1.01354	0.48082	0.32337	0.24136	0.19167
nd=9	0.7448	0.36572	0.24751	0.18483	0.14617

Fig. 7. Dependences of d_2 against the samples number M and against the radius of the working window R_w: $\sigma = 5$ and $K = 2$

data is needed to integrate the filter to the OCT prototype for the skin micro-structure.

Acknowledgements

The authors are grateful to Dr. V. Gelikonov and G. Gelikonov for the preparation of the OCT images and for fruitful discussions and to Prof. I.Prigogine for his enthusiastic support.

This work was supported by the Commission of the European Community within the framework of the ESPRIT Project CTIAC-21042 and the Belgian Government through the Interuniversity Attraction Poles.

References

1. Akishin P.G., Akishina E.P., Akritas P., Antoniou P., Ioannovich J. and Ivanov V.V.:

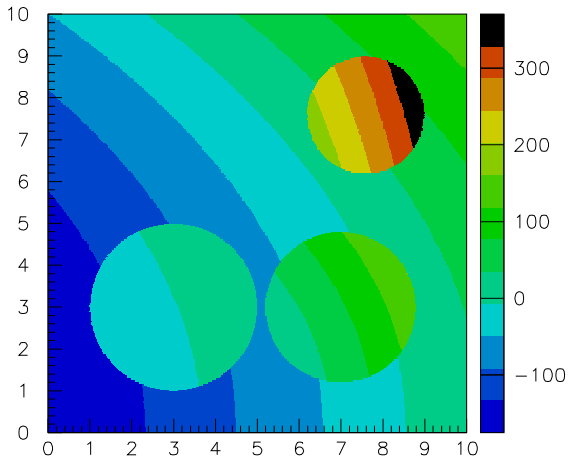


Fig. 8. Filtered model image for the case: $\sigma = 5$, $M = 25$, $R_w = 9$ and $L = 2$

"Stochastic Filtering of Digital Images of Skin Micro-Structure", Presented at the First International Conference "Modern Trends in Computational Physics", June 15-20, 1998, Dubna, Russia. Accepted by "Computer Physics Communications".

2. Gelikonov V., Sergeev A., Gelikonov G., Feldchtein F., Gladkova N., Ioannovich J., Fragia dK., Pirza T. and Antoniou I., *"Compact fast-scanning OCT device for in vivo biotissue imaging"*, in: Conf. on Lasers and Electro-optics, v.9, 1996 OSA Technical Digest Series (Optical Society of America, Washington D.C., 1996), p. 58.
3. Gelikonov V.M., Gelikonov G.V, Gladkova N.D., Kuranov R.V., Nikulin N.K., Petrova G.A., Pochinko V.V., Pravdenko K.I., Sergeev A.M., Feldchtein F.I., Khanin Ya.I., Shabanov D.V., *"Letters to ZhETP"*, vol. **61**, issue 1-2, 1995, p. 149-153 (in Russian).
4. Antoniou I., Babloyantz A., Berry A., Chetverushkin B., Ioannovich J., Ivanov V., Jordan A., Kocharovsky Vi., Melnikov Yu., Mikhov V., Nardone P., Pavlov B., Prigogine I., Sadovnichy V., Schürmann B., "*Computational Tools and Industrial Applications of Complexity*", ESPRIT Project CTIAC-21042, final report, Solvay 99-1, Brussels, 1999, p. 12-32.
5. Kolmogorov A.N., Fomin S.V., "*Introductory Real Analysis*", English translation of R. Silverman, Dover, New York, 1979.

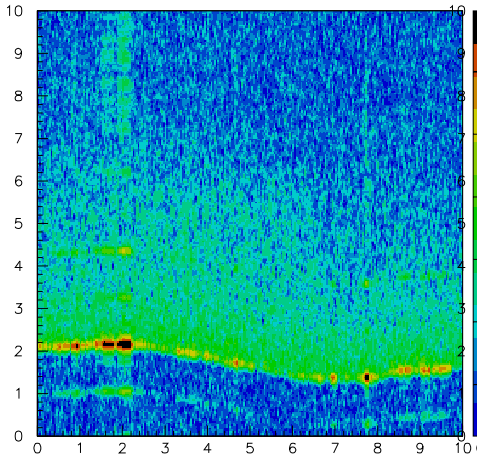


Fig. 9. Real OCT image: 250×250 points

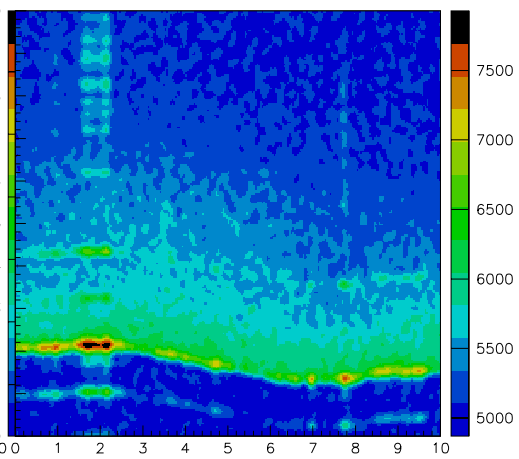


Fig. 10. Filtered OCT image based on 5 samples taken from the same location on the skin tissue: $R_w = 4$ and $L = 2$

Elastic Neural Net Algorithms for the Traveling Salesman Problem

I.E. Antoniou[1,2], S.V. Gorbunov[3], V.V. Ivanov[1,3], I.V. Kisel[3,4], and E.V. Konotopskaya[3]

[1] International Solvay Institutes for Physics and Chemistry, CP-231, ULB, Bd. du Triomphe, 1050, Brussels, Belgium
[2] Theoretische Natuurkunde, Free University of Brussels, Brussels, Belgium
[3] Joint Institute for Nuclear Research, Dubna, 141980 Russia
[4] Max-Planck-Institut für Physik München, Werner Heisenberg Institut, Föhringer Ring 6, D-80805 München, Germany

Abstract. The detailed analysis of an elastic neural net (ENN) proposed for the traveling salesman problem [1], has shown its weak convergence to an optimal tour solution. Aiming to speed up the standard algorithm and make it robust against local minima problem we present three variants of the elastic net method, namely, the Modified ENN, the Double ENN and the Smooth ENN. The efficiency of new algorithms is demonstrated on various sets of cities, on sets of cities with known optimal tours and in comparison with some well-known classical algorithms.

1 Introduction

The Traveling Salesman Sroblem (TSP)

> *Given the positions of N cities, what is the shortest closed tour in which each city can be visited once?*

is a typical problem of combinatorial optimization.

The TSP is a well known NP-complete problem. This means that all known exact methods for the optimal tour determination require a computional time that grows exponentially with the amount of cities. Thus, in practice, exact solutions can be attempted only for cases involving few hundreds cities or less. At the same time, if one finds a polynomial solution for the TSP problem, it can be applied to all other NP problems.

Many heuristic algorithms were developed for the TSP aiming to bypass the combinatorial difficulties. All of them could be subdivided into three classes: tour construction, tour improvement and composite [2].

The tour construction algorithms are aimed to approximately search an optimal tour. The tour improvement algorithms (so-called post-processors) start with a feasible tour and improve it through the local corrections. One of the best tour improving procedures is the Or-opt algorithm [2, 3]. The composite algorithms combine both first algorithms.

The most known "classical" tour construction algorithms include: the arbitrary insertion algorithm [4], the convex hull insertion algorithm [5], the greatest angle insertion algorithm [6, 7] and the ratio times insertion algorithm [3]. Last years some new approaches based on genetic algorithms [8] and neural nets [1, 9, 10] were proposed for the TSP and for other optimization problems.

Among of most successful approaches to the TSP problem is an Elastic Neural Net (ENN) proposed by Durbin and Willshaw [1]. This approach is now widely applied in different areas of optimization problems [9, 11, 12]. For instance, it is effectively applied to the problem of identifying the trajectories of charged particles in high energy physics [11, 13, 14].

We consider the main features of the ENN in Section 2. In order to speed up the original algorithm and to improve the robustness of the ENN against the local minima problem, we propose three modifications of the ENN, namely, the Modified ENN in Section 3, the Double ENN in Section 4 and the Smooth ENN in Section 5. A brief description of a local optimization algorithm is given in Section 6, and the results are discussed in Sections 7 and 8.

2 The ENN algorithm

The ENN can be thought of as a number of beads connected by elastic string to form a closed path.

Following the deformable template approach [11], we denote the cities by $\boldsymbol{x}_i$. We are going to match these cities with template coordinates $\boldsymbol{y}_a$ so that the distance $\sum_a |\boldsymbol{y}_a - \boldsymbol{y}_{a+1}|$ is minimal and that each $\boldsymbol{x}_i$ is matched by at least one $\boldsymbol{y}_a$. Define a binary neuron s_{ia} to be 1 if a is matched to i and 0, otherwise. The energy functional

$$E\left(s_{ia}, \boldsymbol{y}_a\right) = \sum_{ia} s_{ia} \cdot |\boldsymbol{x}_i - \boldsymbol{y}_a|^2 + \gamma \cdot \sum_a |\boldsymbol{y}_a - \boldsymbol{y}_{a+1}|^2 \tag{1}$$

then should be minimized for a valid tour. The multiplier γ governs the relative strength between matching and tour length. The canonical distribution for the energy $E(s_{ia}, \boldsymbol{y}_a)$ is

$$P(s_{ia}, \boldsymbol{y}_a, T) = \frac{\exp[-E(s_{ia}, \boldsymbol{y}_a)/T]}{Z} \tag{2}$$

with partition function

$$Z = \sum_{s_{ia}} \sum_{\boldsymbol{y}_a} \exp[-E(s_{ia}, \boldsymbol{y}_a)/T]. \tag{3}$$

The problem (1) is parametrized in two ways: with neurons s_{ia} and with the template coordinates $\boldsymbol{y}_a$. Choosing the former way and applying the mean field approximation [15, 16] one obtains so-called *marginal distribution* Z_M as

$$Z_M = \sum_{\boldsymbol{y}_a} \prod_i \left[\sum_a \exp(|\boldsymbol{x}_i - \boldsymbol{y}_a|^2/T) \right] \exp[\gamma \sum_a |\boldsymbol{y}_a - \boldsymbol{y}_{a+1}|^2/T]. \tag{4}$$

Only those configurations where s_{ia} is 1 for only one a for each i have been summed over. Equation (4) can be rewritten as

$$Z_M = \sum_{\boldsymbol{y}_a} \exp[-E_{eff}(\boldsymbol{y}_a)], \tag{5}$$

where the *effective energy* E_{eff} is

$$E_{eff}(\boldsymbol{y}_a) = -T \sum_i \log \left[\sum_a \exp(|\boldsymbol{x}_i - \boldsymbol{y}_a|^2/T) \right] + \gamma \sum_a |\boldsymbol{y}_a - \boldsymbol{y}_{a+1}|^2/T. \tag{6}$$

Now we can minimize E_{eff} with respect to $\boldsymbol{y}_a$ using gradient descent method [17]. The deviation of the template coordinate is:

$$\Delta \boldsymbol{y}_a = \eta \left[2 \sum_i v_{ia} \cdot (\boldsymbol{x}_i - \boldsymbol{y}_a) + \gamma \cdot (\boldsymbol{y}_{a+1} - 2\boldsymbol{y}_a + \boldsymbol{y}_{a-1}) \right], \tag{7}$$

where the "continuous" neuron v_{ia} describes the matching of a to i:

$$v_{ia} = \frac{\exp(-|\boldsymbol{x}_i - \boldsymbol{y}_a|^2/T)}{\sum_b \exp(-|\boldsymbol{x}_i - \boldsymbol{y}_b|^2/T)}. \tag{8}$$

The "temperature" T is decreasing at each update of templates $\boldsymbol{y}_a$ (see Table 1), and η is the parameter controlling the minimization speed.

The ENN algorithm for the TSP is summarized in Table 1.

1. Choose the problem — $\boldsymbol{x}_i$'s
2. Choose the number of templates $\boldsymbol{y}_a,\ a = 1, \ldots, M$; $M > N$.
3. Compute the center of gravity of the cities $\boldsymbol{x}_i$ and displace it slightly with a random seed. Place the templates $\boldsymbol{y}_a$ with equal spacing on a circle around this center.
4. For a sequence of temperatures T_n:
4.1 Set $T_n = 0.9 \times T_{n-1}$.
4.2 At each T_n update the templates $\boldsymbol{y}_a$:
$\boldsymbol{y}_a(T_n) = \boldsymbol{y}_a(T_{n-1}) + \Delta \boldsymbol{y}_a$,
where $\Delta \boldsymbol{y}_a$ is given by Eq. 7.

Table 1. The ENN algorithm for the TSP

In the beginning, one has to formulate the problem by choosing it from the set with already known solutions or by using its own distribution of cities [18]. The first way permits to estimate the ENN performance and the second one shows the stability (robustness) of the algorithm.

Then, we have to digitize an ideal continuous net to evolve it on a computer. The amount of beads (templates) of about 2.5 times more than the number of cities provides sufficient digitization within reasonable computing time.

Initially, the templates $\boldsymbol{y}_a$ form a small circle centered at the center of gravity of the cities $\boldsymbol{x}_i$. By the successive recalculation of coordinates $\boldsymbol{y}_a$ (see Eq. 2) the circle is gradually enlarged non-uniformly to eventually pass near all cities and, thus, to define an optimal tour. The details are described in the original paper [1].

Each point $\boldsymbol{y}_a$ on the path moves under the influence of two type of forces (see Eq. 7 and Fig. 1):

– the first one pushes it towards nearest cities;
– the second one pulls it towards its neighbors on the path aiming to minimize the total path.

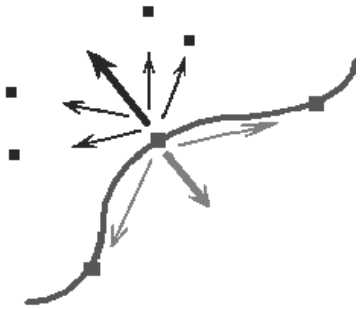


Fig. 1. Two types of forces acting on the path point.

Due to this process, each city becomes associated with a particular section of path. The tightness of this association is determined by how the force contributed from a city depends on its distance, and the degree of this dependence changes with network evolution. Initially, all cities have roughly equal influence on each point of path. Subsequently, a larger distance becomes less favored and each city gradually becomes more influenced for the points on the path closest to it: that corresponds to the decreasing of T in Eq. 8.

Usually, the elastic net method provides the solution with the path length not longer than 105% of an optimal tour, and this is plausible for many practical applications. Figure 2 shows an example of the elastic net evolution for the 100 cities problem.

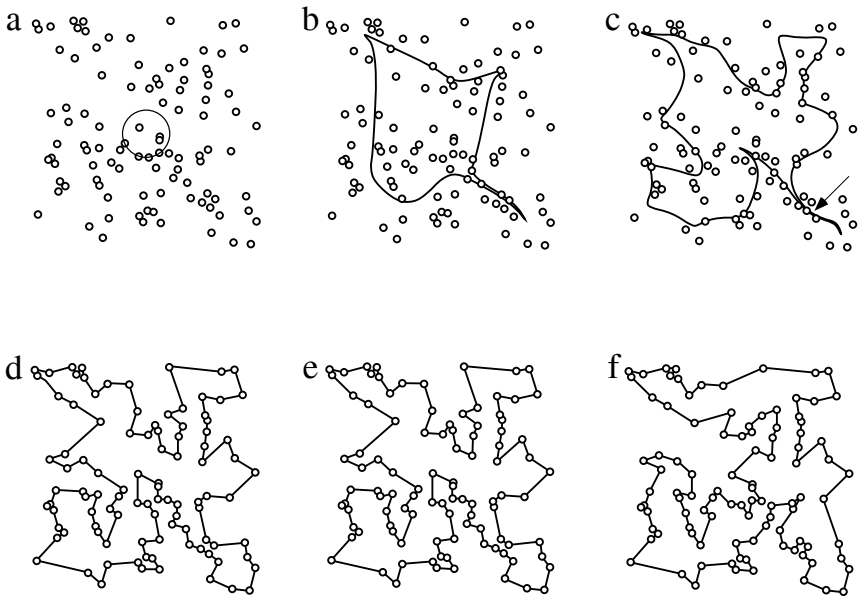


Fig. 2. Elastic net evolution for 100 cities problem

Unfortunately, the standard elastic net is too slow to be directly used for real applications. There are two disadvantages: 1) the calculation of exponential functions in Eq. 3 is very time consuming, and 2) in the case of non-uniform distributions of large number of cities the elastic net may be locked inside a local minimum, and this may require a noticeable time to escape from it.

3 The Modified ENN

In order to simplify and to speed up calculations in the ENN, we propose to modify the matching neuron v_{ia} in the deviation equation (7) as follows:

$$\Delta \boldsymbol{y}_a = \eta \left[2 \sum_i u_{ia} (\boldsymbol{x}_i - \boldsymbol{y}_a) + \gamma \cdot (\boldsymbol{y}_{a+1} - 2\boldsymbol{y}_a + \boldsymbol{y}_{a-1}) \right], \tag{9}$$

where

$$u_{ia} = \frac{\chi_{ia}}{\sum_j \chi_{ja}}, \qquad \chi_{ia} = w_i \left[\frac{\min\limits_b (1 + |\boldsymbol{x}_i - \boldsymbol{y}_b|^2)}{1 + |\boldsymbol{x}_i - \boldsymbol{y}_a|^2} \right]^4. \tag{10}$$

The weight w_i corresponding to the city i

$$w_i = \frac{\sum_k |\boldsymbol{x}_i - \boldsymbol{x}_k|}{\max\limits_j \sum_k |\boldsymbol{x}_j - \boldsymbol{x}_k|} \tag{11}$$

balances this single city with a group of cities.

In the case of the standard ENN each city interacts with the neighboring part of net inside some area which decreases with the iterations. This decrease of the interaction area should be slow enough to attract a bead by a city. For the modified weight function (Eq. 10) the area of attraction depends on the distance between the net and cities only, but does not depend on the iteration.

The modifications described above speeded up essentially the computing time. For instance, for the 52-cities problem the standard elastic net took 7 minutes, but the modified elastic net required only 2.7 seconds for the solution of the same quality.

4 The Double ENN

The minimization of the energy functional (1) is usually realized by applying the gradient descent method [17]. However, there is no guarantee that using this method we will reach the global minimum [19]. At the same time, local minima not only increase the tour length but also make worse its quality. Sometimes, in real applications, where especially curving aspects of tour are essential, this does not permit to get a reliable solution. Thus, local minima may lead to the tour intersections, which can not be corrected even by such efficient local optimization procedure, like the Or-opt algorithm (see section 6).

In general case, to minimize a function of many variables with many local minima most known minimization methods [20] start from some initial point corresponding to the minimizing functional. The minimization is performed in the vicinity of this point (Fig. 3 a). However, in this way there exist a significant non-zero probability to fall down into a local minimum. The only reliable way to escape local minima is to start from a good initial approximation. However, to obtain such an approximation is not very easy.

Contrary to standard methods, we propose to start from two initial points (Fig. 3 b) surrounding the global minimum and covering all physical regions of the parameters [13]. These points attract each other and, thus, pull each other from all local minima until the global minimum will be reached.

In this connection, we propose to simultaneously use two Modified ENNs (see Fig. 4). The attraction between nodes of different nets is determined by the third term of the deviation (7), which becomes:

$$\Delta \boldsymbol{y}_a^1 = \eta \left[2 \sum_i u_{ia} (\boldsymbol{x}_i - \boldsymbol{y}_a^1) + \gamma \cdot (\boldsymbol{y}_{a+1}^1 - 2\boldsymbol{y}_a^1 + \boldsymbol{y}_{a-1}^1) + \beta \cdot (\boldsymbol{y}_a^2 - \boldsymbol{y}_a^1) \right]. \tag{12}$$

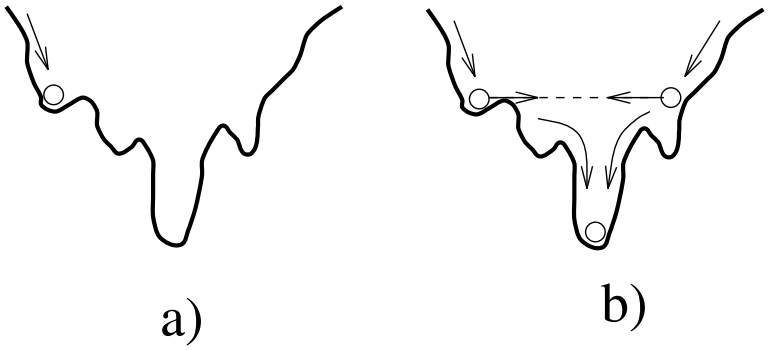


Fig. 3. Searching global minimum by standard ENN (a) and by double ENN (b)

Both nets start from the circular rings centered at the middle of cities' distribution: the first net has small radius and the second one envelopes all cities (see Fig. 4 a). Both nets evolve independently until one or both of them stop in local minimum (minima). Then, the attraction between nets is increasing until they escape a minimum (minima). After this, both nets again evolve independently.

To realize this algorithm we define the attraction force as the inverse proportion to the average bead shift δ at the last iteration:

$$\beta = \frac{T}{1 + K \cdot \delta}, \tag{13}$$

here K is a constant, and T is the parameter which initially equals to 1 and is increasing with iterations.

Fig. 4 demonstrates the evolution of double net for the 52-cities problem. Starting from the initial configuration (Fig. 4 a), both nets stop in different local minima (Fig. 4 b). Then, the attraction between nets is growing up (in Fig. 4 c there are only 3 separate tour sections) and both nets are united into one net (Fig. 4 d). This net evolves as a standard ENN.

5 The Smooth ENN

Another way to solve the local minima problem is to use the smooth properties of an optimal tour [21].

Let us connect the beads of net by a continuous curve. This curve is defined by the coordinate functions $[X(t), Y(t)]$, $t \in [0, 2\pi]$, which can be approximated by the Fourier series:

$$X(t) = \sum_{k=0}^{m} \left[a_k^X \cos(kt) + b_k^X \sin(kt)\right], \quad Y(t) = \sum_{k=0}^{m} \left[a_k^Y \cos(kt) + b_k^Y \sin(kt)\right]. \tag{14}$$

At initial iterations the low frequencies prevail in the coordinate functions, and at the finishing steps the high frequencies are added to the tour. In this

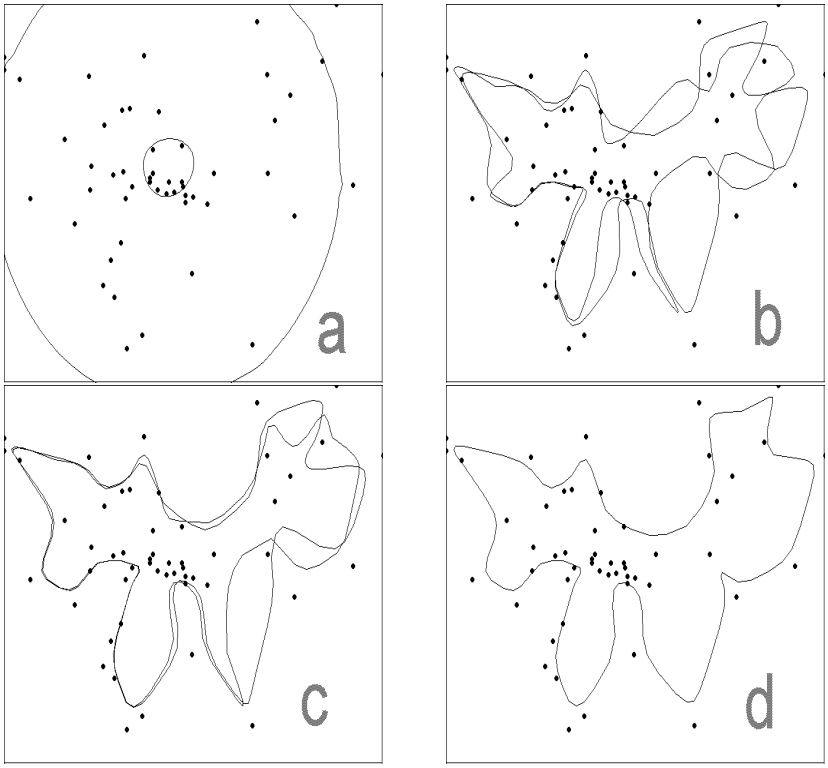


Fig. 4. Double ENN evolution for 52-cities problem

connection, the tour smoothness means decreasing the absolute values of a_k, b_k with increasing of k.

The smoothness of the net is realized by the gradual increase of the number of beads. Initially, the net consists only 4 beads and evolves as a standard ENN. When it falls into a local minimum, the number of beads is increasing, by inserting a new bead between two existing beads. In other words, we run in consecutive order the elastic nets with increasing number of beads. Each net starts from a tour performed before by the previous net.

Figure 5 demonstrates how the smooth elastic net constructs on the basis of first iterations a draft tour using rough properties of the cities distribution, and after this processes the details concerning the local sections of tour. This does not change a global behavior of tour, but makes each new net very fast and improves the quality of solution at each new stage.

6 Local optimization

Aiming to improve a tour found by the main algorithm, one often uses the local optimization algorithms (so-called post-processors). A very fast high quality algorithm is the Or-opt post-processor [2].

To understand the Or-opt procedure, we refer to Fig. 6. For each connected strings of s cities in the tour (s equals 3 first, then 2, then 1), we test if the string can be relocated between two other cities at a reduced cost. If it can, we make the appropriate changes. For $s = 3$ in Fig. 6, each string of three adjacent cities m, n, p in the current tour is considered for the insertion between all pairs of connected cities i and j outside of the string. The insertion is performed if the total cost of the edges to be erased, (a, b), (m, i) and (p, j), exceeds the cost of the new edges to be added, (i, j), (m, a) and (p, b). After considering all strings of three cities, all strings of two cities and then all strings of one city are considered. When no further exchanges improve the solution, the algorithm terminates.

7 Discussion of results

The algorithms described above were tested on various sets of cities. After application of main algorithms the Or-opt postprocessor was used to improve the tour. Table 2 presents the results of a comparative study of different ENNs algorithms. The standard ENN was not tested due to its low speed. The positions of cities were generated randomly inside the unit square. The tour lengths for different algorithms are presented in per cents of the tour length of the modified ENN. All tests were performed on SUN Spark-station 4u.

Number of cities		25	50	100	200
Tour length	Modified ENN	100.00	100.00	100.00	100.00
	Double ENN	99.94	99.34	96.35	94.08
	Smooth ENN	99.78	99.03	96.14	92.57
Tour length with postprocessor	Modified ENN	94.43	92.17	89.27	86.45
	Double ENN	94.80	92.34	89.28	85.46
	Smooth ENN	94.87	91.90	89.66	84.67
Computational time (sec.)	Modified ENN	0.52	2.71	9.06	41.17
	Double ENN	0.99	3.94	16.44	68.96
	Smooth ENN	0.39	1.25	4.04	11.16
Postprocessor time (sec.)	Modified ENN	0.09	0.56	3.12	16.40
	Double ENN	0.09	0.50	2.67	14.47
	Smooth ENN	0.10	0.52	2.54	14.10

Table 2. Comparison of ENNs algorithms on various sets of cities

¿From Table 2 one can see that for 100 or less cities the double and smooth elastic nets give similar results, for 200 cities the smooth elastic net performs better than other nets.

Table 3 presents the performance of the ENN algorithms for some distributions of cities with known optimal tour. The tour lengths are given in percents of an optimal tour length.

Number of cities		52	100	101
Main algorithm	Standard ENN	110.37	105.51	107.26
	Modified ENN	104.86	101.76	105.80
	Double ENN	104.56	101.60	104.52
	Smooth ENN	104.90	103.70	107.35
With postprocessor	Standard ENN	106.05	101.32	103.04
	Modified ENN	104.29	100.73	101.55
	Double ENN	103.36	100.97	102.01
	Smooth ENN	102.68	100.78	100.91

Table 3. Comparison of ENNs algorithms for sets of cities with known optimal tour

Table 4 shows the results of performance for some “classical” algorithms, taken from [2]. The algorithms are denoted as follows:

CC — convex hull cheapest insertion procedure;
RXD — ratio times difference insertion procedure;
GA — greatest angle insertion procedure;
CI — convex hull insertion procedure;
CCA — convex hull, cheapest insertion, angle selection procedure.

Number of cities		50	100	105
Main algorithm	CC	105.80	105.90	103.68
	RXD	108.09	105.48	104.42
	GA	106.39	107.91	106.29
	CI	102.67	103.23	105.73
	CCA	105.16	104.90	102.37
With postprocessor	CC	102.79	103.09	100.08
	RXD	102.31	101.33	100.78
	GA	101.47	102.64	100.62
	CI	102.29	101.57	101.40
	CCA	101.38	101.42	100.00

Table 4. Tour finding results for some “classical” algorithms

From Tables 3 and 4 one can conclude that the algorithms based on the elastic net method produce tours of the same quality as well known classical algorithms.

8 Conclusion

Our application of the standard ENN to the TSP problem has shown that the original algorithm is very slow and that it is unstable against local minima of the energy functional. In order to speed up the original ENN algorithm and to make it robust against local minima, we proposed three modifications of the elastic net method. The Modified (or simplified) ENN is very close to the classical algorithm, but it converges significantly faster to the optimal tour. The Smooth ENN, converges considerably faster than the original algorithm and gives solutions of better quality. The Double ENN is able to solve complex problems where global solution is the main requirement. All 3 algorithms presented here produce tours of the same quality with the well known classical algorithms.

Although the elastic nets provide a solution close to the optimal tour, it needs the application of a local optimization procedure to produce a better final tour. In this connection, we plan to modify further the elastic net in order to decrease the computational time and to incorporate the post-processor into the main algorithm.

9 Acknowledgments

We acknowledge the European Commission for partial support under the ESPRIT project 21042-CTIAC and the Belgian Government through the Interuniversity Attraction Poles. We are grateful to Prof. I.Prigogine for encouragement and support.

References

1. R. Durbin and D. Willshaw, "An Analogue Approach to the Traveling Salesman Problem Using an Elastic Net Method", *Nature* **326** 16 April (1987) 689; see also the Java applet at `http://nuweb.jinr.ru/LNP/NEMO/tspN.html`.
2. E.L. Lawler, J.K. Lenstra, A.H.G. Rinnoy Kan, D.B. Shmoys, "The Traveling Salesman Problem", John Wiley & Sons, Ltd, (1985).
3. I. Or, "Traveling Salesman-Type Combinatorial Problems and their Relation to the Logistics of Regional Blood Banking", *Ph. D. thesis*, Northwestern University, Evanston, 1976.
4. D.J. Rosenkrantz, R.E. Stearns, P.M. Lewis, "An Analysis of Several Heuristics for the Traveling Salesman Problem", *SIAM J. Comput.* **6** (1977) 563.
5. W.R. Stewart, Jr., "A Computationally Efficient Heuristic for the Traveling Salesman Problem", *Proc. 13th Annual Meeting of S.E. TIMS* (1977) 75.

6. J.P. Norback, R.F. Love,"Geometric Approaches to Solving the Traveling Salesman Problem", *Management Sci.* **23** (1977) 1208.
7. J.P. Norback, R.F. Love, "Heuristic for the Hamiltonian Path Problem in Euclidian Two Space", *J. Oper. Res. Soc.* **30** (1979) 363.
8. R.M. Bardy, "Optimization Strategies Gleaned from Biological Evolution", *Nature* **317** (1985) 804.
9. C. Peterson and B. Soderberg, "A New Method for Mapping Optimization Problems onto Neural Networks", *Int. J. Neural Syst.* **1** (1989) 3.
10. J.J. Hopfield and D.W. Thank, "Neural Computation of Decisions in Optimization Problems", *Biol. Cybern.* **52** (1985) 141.
11. C. Peterson and T. Rögnvaldsson, "Introduction to Artificial Neural Networks", *1991 CERN School of Computing*, Ystad, Sweden, 23 August – 2 September 1991, **CERN 92-02**, 1992, p. 113.
12. I. Kisel, V. Neskoromnyi and G. Ososkov, "Applications of Neural Networks in Experimental Physics", *Phys. Part. Nucl.* **24** 6, November-December 1993, p. 657.
13. I. Kisel and V. Kovalenko, "Elastic Net for Broken Multiple Scattered Tracks", *Comput. Phys. Commun.* **98** (1996) 47.
14. I. Kisel, V. Kovalenko, F. Laplanche et al. (NEMO Collaboration), "Cellular Automaton and Elastic Net for Event Reconstruction in the NEMO-2 Experiment". *Nucl. Instr. and Meth.* **A387** (1997) 433.
15. A.L. Yuille, "Generalized Deformable Models, Statistical Physics, and Matching Problems", *Biol. Cybern.* **52** (1985) 141.
16. A.L. Yuille, K. Honda and C. Peterson, "Particle Tracking by Deformable Templates", *Proc. of 1991 IEEE INNS International Joint Conference on Neural Networks*, **Vol. 1**, p. 7, Seattle, WA, 1991.
17. F.P. Vasil'ev, "Chislennie metodi resheniya ekstremal'nix zadach", Nauka, Moscow, 1988 (in Russian).
18. See at http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB.html the library TSPLIB or the directory at `ftp://softlib.rice.edu/pub/tsplib/tsp/` with lots of samples for TSP and related problems.
19. M.W. Simmen, "Parameter Sensitivity of the Elastic Net Approach to the Traveling Salesman Problem", *Neural Comp.* **3** (1991) 363.
20. W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, "Numerical Recipes: the Art of Scientific Computing". Cambridge University Press, 1986.
21. S.V. Gorbunov, I.V. Kisel, E.V. Konotopskaya and R.T. Faizullin, "Comparison of Guaranteed Smoothness and Elastic Net Methods for the Traveling Salesman Problem in Plane.", *JINR Report* **P5-97-258**, Dubna, (1997) (in Russian).

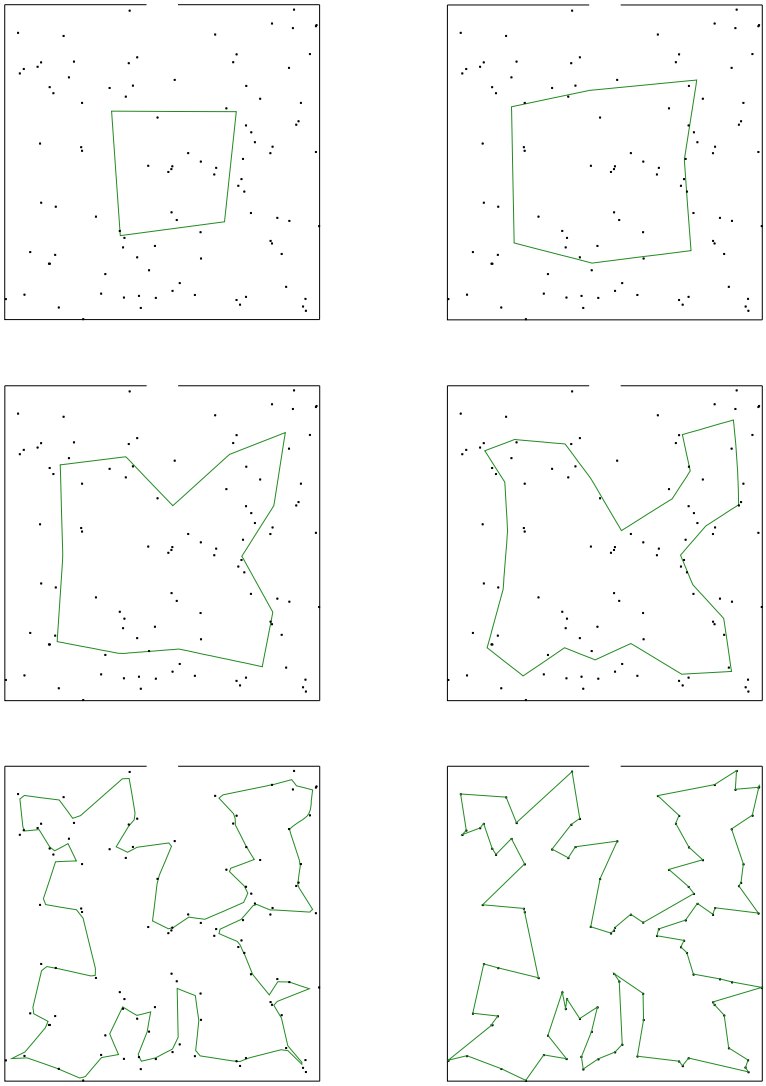


Fig. 5. Smooth ENN evolution for 100 cities problem

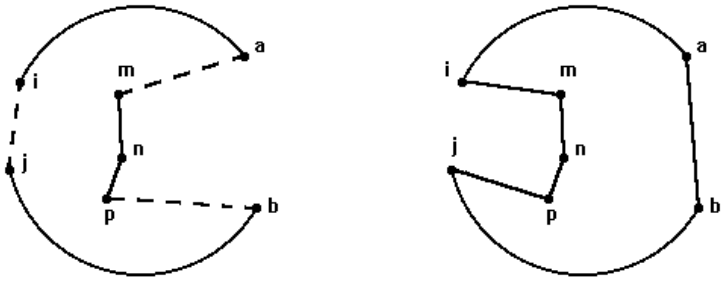


Fig. 6. Or-opt post-processor example for $s = 3$

Elastic Neural Net for the Earthquake Epicenter Search

I. Antoniou[1,2], V.V. Ivanov[1,3] and I.V. Kisel[3,4]

[1] International Solvay Institutes for Physics and Chemistry,
CP-231, ULB, Bd. du Triomphe, 1050, Brussels, Belgium
[2] Theoretische Natuurkunde, Free University of Brussels, Brussels, Belgium
[3] Joint Institute for Nuclear Research, Dubna, 141980 Russia
[4] Max-Planck-Institut für Physik München, Werner Heisenberg Institut, Föringer
Ring 6, D-80805 München, Germany

Abstract. Automatic processing of seismic data is today a key element in the efforts to achieve high quality seismic systems. Automated procedures using for locating seismic events with a network including arrays and single element seismometers usually incorporate back-azimuth estimates, arrival-time data, and associated uncertainties into a least-squares-inverse location algorithm. Such an algorithm is quite cumbersome and requires expanding a set of nonlinear equations in a Taylor series. Second-order terms usually not included in the algorithm can be important if the initial estimate is far from the solution.
We propose to use elastic neural nets (ENN) to find the initial estimation in automated procedures of locating seismic events and discuss the results for simulated seismic events. The advantages of ENN are the simplicity of the algorithm, the fast convergence and the high efficiency.

1 Introduction

With the development of digital seismograph recording systems, the automatic processing of seismic data has became increasingly important. In particular, the ever-decreasing cost and the increasing capacity of modern computers have made possible the implementation of very sophisticated automatic processing algorithms. Today, the limitations do not regard computer capacity, but rather shortcomings of the available automatic schemes.

In March 1978, the group of seismic experts established by the Conference of the Committee on Disarmament (CCD) [1] of the United Nations submitted a report with recommendations regarding an international co-operative system to detect and identify seismic events. Such a system would become a key part of the verification of a comprehensive test ban treaty. The *Ad Hoc* expert group established by the CCD in 1976 has proposed the structure of the global system (Fig. 1). This international co-operative effort involves three main elements:

1. A systematic improvement in the observations reported from more than 50 seismological observatories around the globe.

2. An international exchange of these data over the Global Telecommunications System (GTS) of the World Meteorological Organization.
3. Processing of data at special international data centers for the use of the participant states.

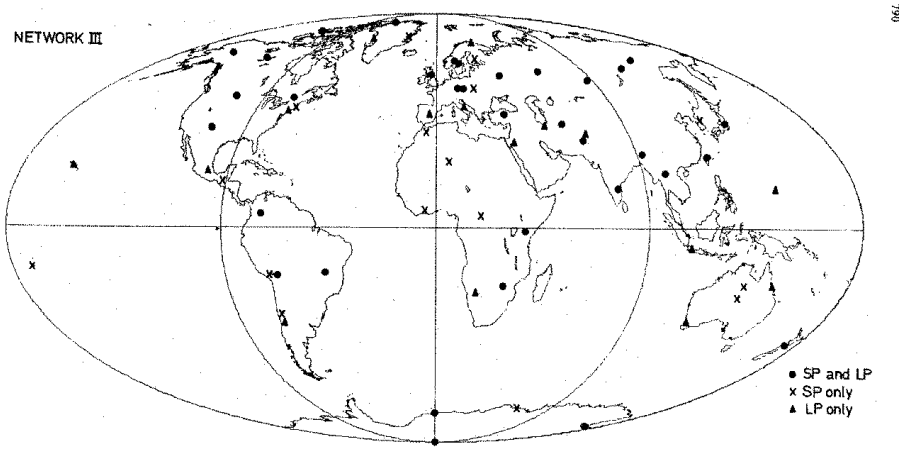


Fig. 1. A possible global network of existing and planned seismic observatories proposed by the CD *Ad Hoc* group of seismic experts

The *Ad Hoc* group considered that special International Data Centers (IDC) should be established for the global network. In order to achieve reliability acceptable to all, it was proposed that more than one international center have to be established, equipped with compatible hardware and software.

The main tasks of the international centers should be:

1. To receive data from the world network of seismic stations via the authorized Government facility of each State.
2. To apply agreed analysis procedures to available data for the estimation of the origin time, location, magnitude and depth of seismic events.
3. To associate the reported identification parameters with these events.
4. To distribute, in accordance with defined procedures and without interpretation of identification parameters, compilations of the complete results of these analyses.
5. To act as an archive for reported data and results of analysis on those data.

2. Seismic event localization procedure

The determination of the seismic event location is one of the most important functions of the IDC. The results of the localization process, such as, the epi-

center location, depth, the origin time, error estimates, form the primary data product of the IDC: the bulletin.

The procedure for event location calculates epicenters, confidence bounds, residuals and data importances using arrival times, back-azimuths, and slownesses from seismic phases recorded by stations at local, regional and teleseismic distances. Travel time, slowness, and azimuth data are, thus, incorporated simultaneously to solve the iterative non-linear least-squares problem for hypocentral locations.

The location procedure follows the approach of Jordan and Sverdrup [2] as extended by Bratt and Bache [3] to include azimuth and slowness data. The key feature of this procedure is that it allows *a priori* knowledge of the data variances to be incorporated along with the resultant *a posteriori* data residuals. This makes calculation of the error ellipses particularly robust.

In order to approximate better the event location in a truly heterogeneous media, while also maintaining the simplicity of travel time tables based on a one-dimensional velocity model, various source-specific corrections can be made. These corrections usually take the form of travel time corrections, but azimuth and slowness corrections are also possible.

Let us present the basis of the location algorithm. Geiger's [4] iterative, least-squares method and its various extensions and modern improvements (e.g., [5, 6]) are the basis of most numerical algorithms for earthquake location. A point seismic source is specified by a space-time vector x of dimension $M \leq 4$. The inequality refers to the case when certain components, such as depth, are fixed throughout the calculation. This location vector is to be constrained by an N-component vector of the observed arrival times t^0. Formally, $x \in E^M$ and $t^0 \in E^N$, where E^N is the Euclidean space of dimension N. Since the number of observation points N is usually large compared to M, the problem is over-determined, although it may still be ill-posed. From the initial estimate x_0, an arrival-time vector $\overline{t}(x_0)$ is calculated by tracing rays through an average radial earth structure; the overbar indicates the assumption of spherical symmetry. By expanding $\overline{t}$ about x_0 we can write

$$\overline{\Delta t} \equiv \overline{t}(x) - \overline{t}(x_0) = A \cdot \Delta x + O\left[|\Delta x|^2\right] \tag{1}$$

where $\Delta x \equiv x - x_0$ and A is an $N \times M$ matrix of partial derivatives evaluated at x_0. If we assume that the rank $[A] = M$, then a refined estimate of x can be uniquely obtained from the residual vector $\overline{\Delta t} \equiv \overline{t}(x) - \overline{t}(x_0)$ by minimizing the norm of the error vector

$$e^0 = \Delta t^0 - A \cdot \Delta x. \tag{2}$$

To account for the curvature of the $\overline{t}(x)$ manifold, the estimate of Δx is added to x_0 and the procedure is iterated. The convergence is usually rapid in the teleseismic case.

In certain situations, especially for epicenters determined by sparse local networks and/or when locating local and near-regional events that lie outside

the network, the nonlinear aspects of the problem may be important [5–7]. Techniques, including nonlinear contributions to local uncertainty and the shape of these nonlinear confidence bounds, may be quite complex compared to the ellipsoids resulting from the linear approximation.

It becomes obvious that the calculation of the initial estimation of x_0 is the crucial point of the full location algorithm. We shall use an elastic neural net for estimating x_0.

2 Elastic net method

The elastic net is a kind of artificial neural net [8] which has been used for event recognition in high energy physics.

The elastic neural net method can be illustrated on a simple example of the traveling salesman problem. The problem is:

> *Given the positions of N cities, what is the shortest closed tour in which each city can be visited once?*

One of the most successful approaches to the traveling salesman problem is the elastic neural net of Durbin and Willshaw [9][1], which can be understood as a number of beads connected by an elastic string to form a ring. The essence of the method is:

> *Using an iterative procedure, a circular closed path is gradually elongated non-uniformly until it eventually passes sufficiently near to all the cities to define a tour.*

Let us denote the cities by $\boldsymbol{x}_i$. We are going to match these cities with template coordinates $\boldsymbol{y}_a$ such that the sum $\sum_a |\boldsymbol{y}_a - \boldsymbol{y}_{a+1}|$ is minimized and that each $\boldsymbol{x}_i$ is matched by at least one $\boldsymbol{y}_a$. Define s_{ia} to be 1 if a is matched to i and 0 otherwise. The following energy expression [10] then minimizes a valid tour

$$E\left(s_{ia}, \boldsymbol{y}_a\right) = \sum_{ia} s_{ia} \cdot \left|\boldsymbol{x}_i - \boldsymbol{y}_a\right|^2 + \gamma \cdot \sum_a \left|\boldsymbol{y}_a - \boldsymbol{y}_{a+1}\right|^2 . \tag{3}$$

The coefficient γ governs the relative strength between matching and tour length. Then the dynamical equations can be written:

$$\Delta \boldsymbol{y}_a = \Delta \boldsymbol{y}_a^{(1)} + \Delta \boldsymbol{y}_a^{(2)} = \eta \left[2 \sum_i v_{ia} \cdot (\boldsymbol{x}_i - \boldsymbol{y}_a) + \gamma \cdot (\boldsymbol{y}_{a+1} - 2\boldsymbol{y}_a + \boldsymbol{y}_{a-1}) \right] , \tag{4}$$

where the decision value v_{ia} of a to be matched to i is given by

$$v_{ia} = \frac{e^{-|\boldsymbol{x}_i - \boldsymbol{y}_a|^2/T}}{\sum_b e^{-|\boldsymbol{x}_i - \boldsymbol{y}_b|^2/T}} . \tag{5}$$

[1] `http://nuweb.jinr.ru/LNP/NEMO/tspN.html`

The "temperature" T is decreasing during evolution.

The algorithm is a procedure for the successive recalculation of the positions of a number of points in the plane in which the cities lie. The points describe a closed path which is initially a small circle centered on the center of the distribution of cities and is gradually elongated non-uniformly to eventually pass near all the cities and thus define a tour around them (for details see the original paper [9]). Each point on the path moves under the influence of **two types of force** (see Eq. 4):

1. the first pulls towards the nearest cities;
2. the second pulls towards neighboring points on the path, acting to minimize the total path length.

By this process, each city becomes associated with a particular section on the path. The tightness of the association is determined by how the force contributed from a city depends on its distance, and the nature of this dependence changes as the algorithm progresses. Initially all cities have roughly equal influence on each point of the path. Subsequently a larger distance becomes less favored and each city gradually becomes more influenced for the points on the path closest to it.

Usually the ENN finds a solution close to the best one, that is practically enough in many applications. Thus, in our case we should find a solution, which could be used for making a decision about a vertex existence and for using it as an initial approximation for the least squares method.

An example of the TSP solution for 100 cities, using ENN is presented in Fig. 2.

3 Elastic net for the earthquake problem

Let us formulate the problem of searching for an initial estimation of a seismic event epicenter. Data from a network of arrays and single element seismometers include arrival times, back-azimuths, and slownesses from seismic phases recorded by stations at local, regional and teleseismic distances. This information could be represented by the distance and the direction towards the earthquake center registered by seismometers with corresponding errors that give confidence ellipsoids for each station. Parameters and location of an ellipsoid depend on the technical characteristics of the station and the distance to the event source.

For testing the elastic net algorithm, we simulate data from 6 "correct" stations and 3 "noise" stations (see Fig. 4). This usually happens analyzing seismic data at IDC. A "correct" radius (detection error) is random up to 5.0 relative units. A "correct" center location is generated with a random shift within the generated detection error with respect to the generated event location and with random angle distribution. "Noise" detection error is the same as

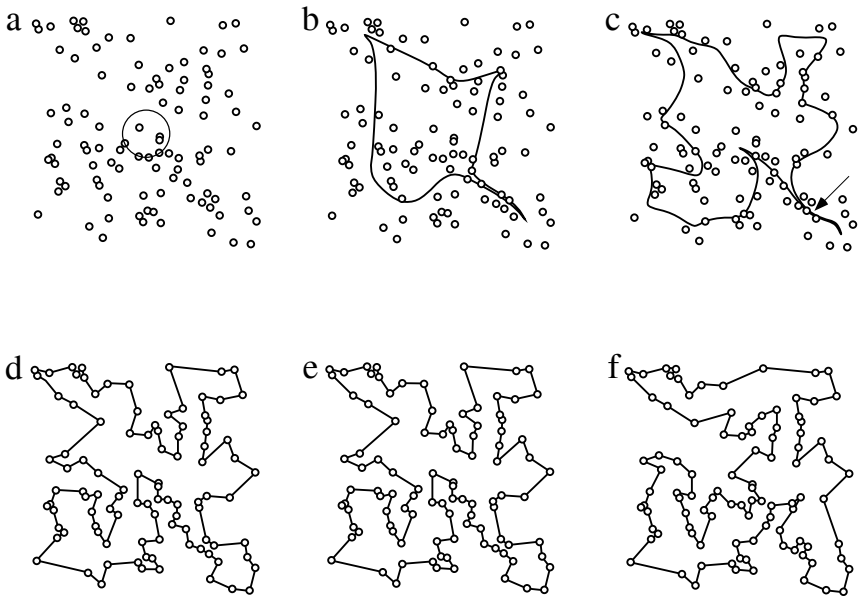


Fig. 2. Example of the progress of the elastic net method for 100 cities

for "correct" stations but locations are shifted randomly in addition up to 5.0 relative units to be far from the true location[2].

Let us

Define an epicenter of a seismic event as the geometrical point with a maximum density of confidence ellipsoids.

The earthquake location procedure at presence of noise data can be considered as a problem of minimization of a function of many variables with many local minima. All known standard methods of minimization [12] start with a point on the function surface and minimize the function in a local area around it, at each step of the iteration procedure. In this way there is a significantly non-zero probability to stop at a local minimum. It is possible to build methods against local minima to save the standard methods, but to do so the complexity of our program increases significantly. The only reliable way is to start with a good initial approximation, which cannot be easily found in our case. So the use of the standard minimization methods is not suitable.

Contrary to the standard methods, we propose to use an elastic ring surrounding the global minimum and covering the region where the parameters

[2] Similar problem appears searching for the reaction vertex in high energy physics experiments [11].

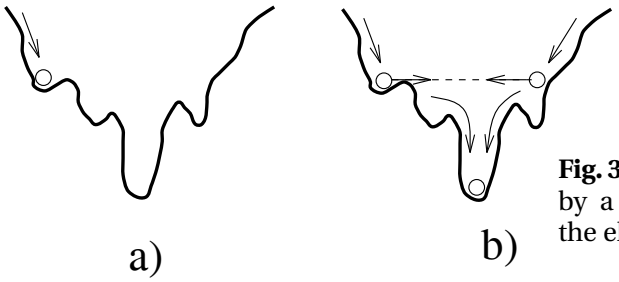


Fig. 3. Global minimum search by a standard method (a) and the elastic net method (b)

are likely to be found. This ring will be squeezed, catching on all local minima until the global minimum will be reached [13]. Graphically, this method is represented in Fig. 3.

We construct the algorithm on the basis of an elastic ring by introducing **two types of force**:

1. *Attraction of the ring to all ellipsoids placing it at the condensation area of the ellipsoids.*
   Here confidence ellipsoids play a role of cities in the traveling salesman problem. The attraction force of the ring to an ellipsoid depends on its distance and location accuracy given by a station. The sensitivity region of the ring is decreasing during iterations being more confident at the end of the process.
2. *Squeezing of the ring leading to localization of the source region.*
   The influence of this force increases with algorithm iterations. The radius of the ring corresponds to the most probable area surrounding the epicenter.

4 Results and conclusion

Fig. 4 shows the result of the application of the ENN to the earthquake location searching using data from 9 seismic stations. Here the star denotes the epicenter of the simulated earthquake and 6 solid circles are simulated confidence circles given by seismic stations. Other stations give noisy data and one can see 3 circles far from the epicenter. Dashed circles demonstrate the iterative process of the elastic net that converges near the epicenter. This found position can be used as the initial estimation in the procedure of the epicenter calculation (Eq. 2).

The advantages of the method are the simplicity of the algorithm, the fast convergence and the high and stable reconstruction efficiency. One can see on Fig. 5 that the algorithm is able to find the epicenter even with 50% wrong seismic data.

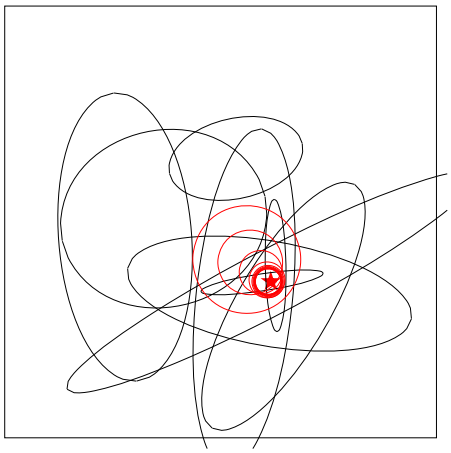


Fig. 4. An evolution of the elastic net on a data from 6 correct and 3 noise seismic stations

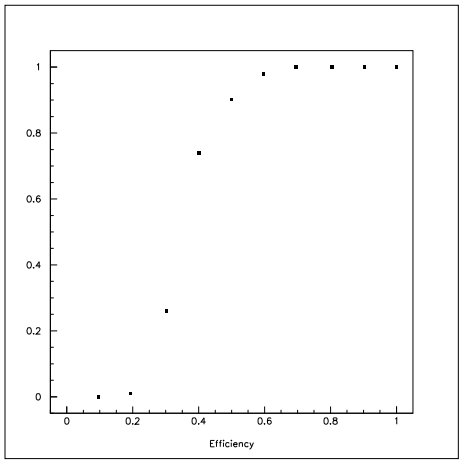


Fig. 5. Efficiency of the epicenter searching depending on seismic data efficiency

5 Acknowledgments

We acknowledge the International Solvay Institutes for Physics and Chemistry and the Belgian Government through the Inter-university Attraction Poles for financial support. We are grateful to Prof. I.Prigogine for encouragement and support.

References

1. Conference of the Committee on Disarmament paper CCD/558, 1978, Geneva.
2. T.H. Jordan and K.A. Sverdrup, Teleseismic location techniques and their application to earthquake clusters in the south-central Pacific, *Bull. Seism. Soc. Am.* **71** (1981) 1105.
3. S.R. Bratt and T.C. Bache, Locating events with a sparse network of regional arrays, *Bull. Seism. Soc. Am.* **78** (1988) 780.
4. L. Geiger, Herdbestimmung bei Erdbeben ans den Ankunftzeiten, *K. Gessell. Wiss. Goett.* **4** (1910) 331.
5. E.A. Flinn, Confidence regions and error determinations for seismic event location, *Rev. Geophys.* **3** (1965) 157.
6. R. Buland, The mechanics of locating earthquakes, *Bull. Seism. Soc. Am.* **66** (1976) 173.
7. W. Spence, Relative epicenter determination using P-wave arrival-time differences, *Bull. Seism. Soc. Am.* **70** (1980) 171.
8. I. Kisel, V. Neskoromnyi and G. Ososkov, Applications of neural networks in experimental physics, *Phys. Part. Nucl.* **24** (6), November–December 1993, p. 657.

9. R. Durbin and D. Willshaw, An analogue approach to the traveling salesman problem using an elastic net method, *Nature* **326** 16 April (1987) 689.
10. C. Peterson and T. Rögnvaldsson, Introduction to artificial neural networks, *1991 CERN School of Computing*, Ystad, Sweden, 23 August – 2 September 1991, **CERN 92-02**, 1992, p. 113.
11. I. Kisel, E. Konotopskaya and V. Kovalenko, Elastic neural net for track and vertex search, *Fifth International Workshop on Software Engineering, Artificial Intelligence, Neural Nets, Genetic Algorithms, Symbolic Algebra and Automatic Calculation AIHENP'96*, Lausanne, Switzerland, 1–6 September 1996; *Nucl. Instr. and Meth.* **A389** (1997) 167.
12. W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, *Numerical Recipes: the Art of Scientific Computing*. Cambridge University Press, 1986.
13. I. Kisel and V. Kovalenko, Elastic net for broken multiple scattered tracks, *Comp. Phys. Commun.*, **98** (1996) 45.

On a Binomial Model of Forecasting Option Prices

I. Antoniou[1,2], V.V. Ivanov[1,3] and A.V. Kryanev[1,4]

[1] International Solvay Institutes for Physics and Chemistry, CP-231, ULB, Bd. du Triomphe, 1050, Brussels, Belgium
[2] Theoretische Natuurkunde, Free University of Brussels, Brussels, Belgium
[3] Laboratory of Information Technologies, Joint Institute for Nuclear Research, 141980, Dubna, Russia
[4] Moscow Engineering and Physical Institute, 115409, Moscow, Russia

Abstract. The conventional models of option pricing are based on the artificial proposition of the market neutrality to the risk, which is equivalent to the assumption that the expected return μ equals the risk-free return r. However, in real market μ may significantly differ from r, and this may lead to the noticeable difference between model and real values of option prices. In the present work we consider a modification of the Cox-Ross-Rubenstein binomial model that takes into account the dependence of option prices with respect to μ. After proving the extended local theorem of de Moivre-Laplace and applying transition to the continuous limit we obtained the same option price formula as for the continuous model in [AIK]. We show that option prices based on this formula and on Bachelier's model (for which the dependence on μ and the discount factor were taken into account) give close results. At the same time, the option prices obtained by the Black-Scholes formula gives different results. This difference is increasing together with the growth of difference between μ and r may reach 20–30%.

1. Introduction

The conventional models for the determination of the option prices [CR1–Hau] are based on the assumption of market neutrality to the risk. In particular, the famous Black-Scholes option pricing formula:

$$C_{BS} = S \cdot \Phi\left(\frac{\ln\frac{S}{X} + rT + \frac{\sigma^2}{2}T}{\sigma\sqrt{T}}\right) - X\exp(-rT)\cdot\Phi\left(\frac{\ln\frac{S}{X} + rT - \frac{\sigma^2}{2}T}{\sigma\sqrt{T}}\right), \quad (1)$$

was obtained for the continuous model [BS1] and for the discrete binomial model [CRR–NeRa] under the assumption that the expected return μ of the underlying assets equals the risk-free return r.

Here C_{BS} is the option price, S is the stock price, σ is the volatility of the stock price per annum, X is the option exercise price (strike price), T is the option expiration date (maturity), and

$$\Phi(\alpha) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\alpha}\exp(-\frac{S^2}{2})dS$$

is the standard normal cumulative distribution function.

However, in a real risky market μ may differ significantly from r. For example, it is known that on the average for the stock market μ is about two times greater than the return r on a risk-free investment, i.e. $\mu \approx 2r$ (see [Hul], p. 219). As a result we may have a noticeable difference between the predicted and real values of the option prices [AIK].

We introduce in Section 2 a modification of the Cox-Ross-Rubenstein discrete binomial model [CRR] taking into account the dependence of option prices on μ. We realize in Section 3 the transition to the continuous limit when $\mu = r$ using the de Moivre-Laplace theorem and when $\mu \neq r$ by extending the de Moivre-Laplace theorem in Section 4.

2. Modification of the Cox-Ross-Rubenstein model for the case $\mu \neq r$

We consider a modification of the Cox-Ross-Rubenstein model in the form, which allows for the case $\mu \neq r$.

For the determination of the option price we use the equality

$$C = d(T) \cdot E(\max(S(T) - X, 0)), \tag{2}$$

where $d(T)$ is the discount factor and E is the operator of mathematical expectation (without any assumptions).

Following [CRR], we divide the time interval [0,T] into n subintervals with duration $\Delta t = \frac{T}{n}$. During each subinterval the stock price may grow by a factor $u_n > 1$ with probability p_n, or to diminish by a factor $d_n < 1$ with probability $q_n = 1 - p_n$.

We assume, moreover, that u_n, d_n, p_n depend neither on the location of the subinterval $[m\,\Delta t, (m+1)\,\Delta t]$, nor on the previous change of stock price. This actually means that the stock price change is described by a stationary Markov chain.

For this model the call option price C_n satisfies the binomial law [CRR]:

$$C_n = \sum_{m=0}^{n} \frac{n!}{m!(n-m)!} \cdot \frac{p_n^m\, q_n^{n-m}}{(1 + r\frac{T}{n})^n} \cdot \max\{0, S u_n^m d_n^{n-m} - X\}, \tag{3}$$

where $S u_n^m d_n^{n-m}$ is the stock price to the option execution date in the case of m times price increase and $(n-m)$ times price decrease. The probability of realizing m times increase and $(n-m)$ times decrease of the stock price during n steps starting from the current moment $t = 0$ till the option execution date $t = n \cdot \Delta t = T$ is:

$$P(m,n) = \frac{n!}{m!(n-m)!} \cdot p_n^m q_n^{n-m}.$$

The values of u_n, d_n, p_n are found from the continuous Samuelson model:

$$dS = \mu\, S(t)\, dt + \sigma\, S(t)\, dz(t), \tag{4}$$

where $S(t)$ is the stock price at time t, $dz(t)$ is the stochastic differential of the normalized Wiener process with zero mean and variance dt.

Let us consider the stochastic function $\eta(t) = \ln \frac{S(t)}{S}$, where $S(0) = S$ is assumed to be a deterministic variable.

Then, according to Ito's formula, we have

$$\begin{aligned} E\{\eta(t)\} &= (\mu - \frac{\sigma^2}{2})\, t, \\ \sigma^2\{\eta(t)\} &= \sigma^2\, t. \end{aligned} \tag{5}$$

On the other hand, for the above considered discrete binomial model we have

$$\ln \frac{S(T)}{S} = m\, \ln \frac{u_n}{d_n} + n\, \ln d_n.$$

Then, for the discrete binomial model, we have

$$\begin{aligned} E\{\ln \frac{S(T)}{S}\} &= E\{m\} \ln \frac{u_n}{d_n} + n\, \ln d_n = p_n\, n\, \ln \frac{u_n}{d_n} + n\, \ln d_n, \\ \sigma^2\{\ln \frac{S(T)}{S}\} &= \sigma^2\{m\}\, \ln^2 \frac{u_n}{d_n} = n\, p_n\, q_n\, \ln^2 \frac{u_n}{d_n}. \end{aligned} \tag{6}$$

In order that formulae (5) and (6) be compatible it is necessary to fulfill the conditions:

$$\begin{aligned} \lim_{n\to\infty} n\, (p_n\, \ln \frac{u_n}{d_n} + \ln d_n) &= (\mu - \frac{\sigma^2}{2})\, T, \\ \lim_{n\to\infty} n\, p_n\, q_n\, \ln^2 \left(\frac{u_n}{d_n}\right) &= \sigma^2\, T. \end{aligned} \tag{7}$$

Equations (7) are fulfilled if we assume that

$$\begin{aligned} u_n &= 1 + \sigma \sqrt{\frac{T}{n}} + \frac{\sigma^2}{2} \cdot \frac{T}{n}, \\ d_n &= 1 - \sigma \sqrt{\frac{T}{n}} + \frac{\sigma^2}{2} \cdot \frac{T}{n}, \\ p_n &= \frac{1}{2} + \frac{1}{2} \left(\frac{\mu}{\sigma} - \frac{\sigma}{2}\right) \cdot \sqrt{\frac{T}{n}}, \quad q_n = 1 - p_n. \end{aligned} \tag{8}$$

Remark 1: We assume that n is so large that $d_n,\, p_n \in\ (0, 1)$.

Remark 2: u_n and d_n in (8) with accuracy $\mathrm{O}(\frac{1}{n})$ coincide with $\exp(\sigma\sqrt{\frac{T}{n}})$ and $\exp(-\sigma\sqrt{\frac{T}{n}})$, correspondingly.

Let us denote by m^* the least integer m for which the inequality

$$S u_n^m d_n^{n-m} \geq X \tag{9}$$

is fulfilled.

For sufficiently large n we have

$$m^* = \left[\frac{\ln \frac{X}{S} + \sigma\sqrt{n\,T}}{2\sigma\sqrt{\frac{T}{n}}} \right], \tag{10}$$

where $[M]$ is the integer part of the number M.

Remark 3: As $m^* \to +\infty$ while $n \to +\infty$, then for sufficiently large n and at the limit $n \to +\infty$ m^* is close to the real number:

$$m^* \simeq \frac{\ln \frac{X}{S} + \sigma\sqrt{n\,T}}{2\sigma\sqrt{\frac{T}{n}}}.$$

From the definition of m^* and formula (3) follows:

$$C_n = \sum_{m=m^*}^{n} \frac{P(m,n)}{(1+r\frac{T}{n})^n} \cdot (S u_n^m d_n^{n-m} - X), \tag{11}$$

where p_n, u_n, d_n are given by equalities (8).

3. The limit transition when $\mu = r$ based on the de Moivre-Laplace theorem

First, let assume that condition $\mu = r$ is justified. Then, from (11) follows the equality:

$$C_n = \sum_{m=m^*}^{n} \frac{n!}{m!(n-m)!} \cdot \left(S \hat{p_n}^m \hat{q_n}^{n-m} - X \frac{\tilde{p_n}^m \tilde{q_n}^{n-m}}{(1+r\frac{T}{n})^n} \right), \tag{12}$$

where

$$\tilde{p_n} = \frac{1}{2} + \frac{1}{2}\left(\frac{r}{\sigma} - \frac{\sigma}{2}\right)\sqrt{\frac{T}{n}}, \quad \tilde{q_n} = 1 - \tilde{p_n},$$

$$\hat{p_n} = \frac{p_n u_n}{1+r\frac{T}{n}} = \frac{\frac{1}{2} + \frac{1}{4}\sigma\sqrt{\frac{T}{n}} + \frac{r}{2\sigma}\sqrt{\frac{T}{n}} + \frac{1}{2}r\frac{T}{n} + \frac{r\sigma}{4}\left(\frac{T}{n}\right)^{\frac{3}{2}} - \frac{\sigma^3}{8}\left(\frac{T}{n}\right)^{\frac{3}{2}}}{1+r\frac{T}{n}},$$

$$\hat{q_n} = \frac{q_n d_n}{1+r\frac{T}{n}} = \frac{\frac{1}{2} - \frac{1}{4}\sigma\sqrt{\frac{T}{n}} - \frac{r}{2\sigma}\sqrt{\frac{T}{n}} + \frac{1}{2}r\frac{T}{n} - \frac{r\sigma}{4}\left(\frac{T}{n}\right)^{\frac{3}{2}} + \frac{\sigma^3}{8}\left(\frac{T}{n}\right)^{\frac{3}{2}}}{1+r\frac{T}{n}}, \quad \hat{q_n} = 1 - \hat{p_n}.$$

Therefore, in order to find $C = \lim_{n\to+\infty} C_n$, where C_n is determined by (12), it is possible to use the local limit theorem of de Moivre-Laplace [Fel], due to which

$$\frac{n!}{m!\cdot(n-m)!}\tilde{p_n}^m \tilde{q_n}^{n-m} = \frac{1}{\sqrt{2\pi n \tilde{p_n}\tilde{q_n}}} \exp(-\frac{\tilde{x_m}^2}{2}) \left(1 + O(\frac{1}{\sqrt{n}})\right),$$

$$\frac{n!}{m!\cdot(n-m)!}\hat{p_n}^m\hat{q_n}^{n-m}=\frac{1}{\sqrt{2\pi n\hat{p_n}\hat{q_n}}}\exp(-\frac{\hat{x_m}^2}{2})\left(1+O(\frac{1}{\sqrt{n}})\right),$$

where

$$\tilde{x_m}=\frac{m-n\cdot\tilde{p_n}}{\sqrt{n\tilde{p_n}\tilde{q_n}}},\qquad \hat{x_m}=\frac{m-n\cdot\hat{p_n}}{\sqrt{n\hat{p_n}\hat{q_n}}}$$

and $\sqrt{n}\,O(1/\sqrt{n})\to C<\infty$ while $n\to+\infty$.

We have

$$\tilde{x_m}^*=\frac{m^*-n\cdot\tilde{p_n}}{\sqrt{n\tilde{p_n}\tilde{q_n}}}\to\frac{\ln\frac{X}{S}-rT+\frac{\sigma^2}{2}T}{\sigma\sqrt{T}},$$

$$\hat{x_m}^*=\frac{m^*-n\cdot\hat{p_n}}{\sqrt{n\hat{p_n}\hat{q_n}}}\to\frac{\ln\frac{X}{S}-rT-\frac{\sigma^2}{2}T}{\sigma\sqrt{T}},$$

$$\tilde{x_n}=\frac{n-n\cdot\tilde{p_n}}{\sqrt{n\tilde{p_n}\tilde{q_n}}}\to+\infty,\quad \hat{x_n}=\frac{n-n\cdot\hat{p_n}}{\sqrt{n\hat{p_n}\hat{q_n}}}\to+\infty,$$

while $n\to+\infty$.

Thus, due to the integral theorem of Moivre-Laplace, we get $C=\lim_{n\to+\infty}C_n$, where C is determined by the Black-Scholes option pricing formula (1).

4. The limit transition when $\mu\neq r$. Extension of the de Moivre-Laplace theorem

In the general case, when condition $\mu=r$ is not fulfilled, we have:

$$C_n=\sum_{m=m^*}^{n}\frac{n!}{m!(n-m)!}\cdot p_n^m q_n^{n-m}\cdot\left(Su_n^m d_n^{n-m}-X\right)\cdot\frac{1}{(1+r\frac{T}{n})^n},\tag{13}$$

where p_n, q_n, u_n, d_n are determined by equations (8).

We have

$$x_{m^*}=\frac{m^*-n\cdot p_n}{\sqrt{np_nq_n}}\to\frac{\ln\frac{X}{S}-\mu T+\frac{\sigma^2}{2}T}{\sigma\sqrt{T}},$$

$$x_n=\frac{n-n\cdot p_n}{\sqrt{np_nq_n}}\to+\infty,\quad \text{for}\quad n\to+\infty.$$

Therefore, due to the integral theorem of Moivre-Laplace, we have the relation:

$$\sum_{m=m^*}^{n}\frac{n!}{m!(n-m)!}\cdot p_n^m q_n^{n-m}\to\Phi\left(\frac{\ln\frac{S}{X}+\mu T-\frac{\sigma^2}{2}T}{\sigma\sqrt{T}}\right).\tag{14}$$

Thus, in order to find the $\lim_{n\to+\infty}C_n$, where C_n is determined by equation (13), it is enough to find the limit:

$$\lim_{n\to+\infty}\frac{1}{(1+r\frac{T}{n})^n}\sum_{m=m^*}^{n}\frac{n!}{m!(n-m)!}\cdot(p_nu_n)^m\cdot(q_nd_n)^{n-m}.\tag{15}$$

As the inequalities $p_n u_n + q_n d_n \neq 1$ and $p_n u_n + q_n d_n \neq 1 + r\frac{T}{n}$ are fulfilled for $\mu \neq r$, it is not possible to apply the local theorem of Moivre-Laplace for the limit (15). Therefore, we shall extend the local theorem of Moivre-Laplace, namely:

Theorem 1. The following asymptotic expression holds for large n:

$$I_{n,m} = \frac{n!}{m!(n-m)!} \cdot \bar{p_n}^m \bar{q_n}^{n-m} = \frac{1}{\sqrt{2\pi n \ddot{p}_n \ddot{q}_n}} \exp(\mu T) \exp(-\frac{\ddot{x_m}^2}{2}) \left(1 + O(\frac{1}{\sqrt{n}})\right), \tag{16}$$

where

$$\ddot{p_n} = \frac{1}{2} + \frac{1}{2}\left(\frac{\mu}{\sigma} + \frac{\sigma}{2}\right)\sqrt{\frac{T}{n}}, \quad \ddot{q_n} = 1 - \ddot{p_n},$$

$$\ddot{x_m} = \frac{m - n \cdot \ddot{p_n}}{\sqrt{n \ddot{p}_n \ddot{q_n}}}, \quad \bar{p_n} = p_n u_n, \quad \bar{q_n} = q_n d_n.$$

Proof. We have

$$m = n\ddot{p_n} + \ddot{x_m}\sqrt{n \cdot \ddot{p_n}\,\ddot{q_n}}, \quad n - m = n\ddot{q_n} - \ddot{x_m}\sqrt{n \cdot \ddot{p_n}\,\ddot{q_n}}.$$

Using the Stirling formula and the equalities:

$$\bar{p_n} = \ddot{p_n} + \frac{\mu}{2} \cdot \frac{T}{n} + O(\frac{1}{n^{3/2}}),$$

$$\bar{q_n} = \ddot{q_n} + \frac{\mu}{2} \cdot \frac{T}{n} + O(\frac{1}{n^{3/2}}),$$

we get

$$\begin{aligned}
\ln\, I_{n,m} = {} & \ln\sqrt{2\pi\, n} + n\ln n - n - ln\sqrt{2\pi\,(n\ddot{p_n} + \ddot{x_m}\sqrt{n \cdot \ddot{p_n}\ddot{q_n}})} - \\
& (n \cdot \ddot{p_n} + \ddot{x_m}\sqrt{n \cdot \ddot{p_n}\ddot{q_n}}) \cdot \ln(n\ddot{p_n} + \ddot{x_m}\sqrt{n \cdot \ddot{p_n}\ddot{q_n}}) + \\
& n \cdot \ddot{p_n} + \ddot{x_m}\sqrt{n \cdot \ddot{p_n}\ddot{q_n}} - \ln\sqrt{2\pi(n\ddot{q_n} - \ddot{x_m}\sqrt{n \cdot \ddot{p_n}\ddot{q_n}})} - \\
& (n \cdot \ddot{q_n} - \ddot{x_m}\sqrt{n \cdot \ddot{p_n}\ddot{q_n}}) \cdot \ln(n\ddot{q_n} - \ddot{x_m}\sqrt{n \cdot \ddot{p_n}\ddot{q_n}}) + \\
& n \cdot \ddot{q_n} - \ddot{x_m}\sqrt{n \cdot \ddot{p_n}\ddot{q_n}} + (n\ddot{p_n} + \ddot{x_m}\sqrt{n \cdot \ddot{p_n}\ddot{q_n}} \times \\
& \ln(\ddot{p_n} + \tfrac{\mu}{2} \cdot \tfrac{T}{n} + O(\tfrac{1}{n^{3/2}})) + (n\ddot{q_n} - \ddot{x_m}\sqrt{n \cdot \ddot{p_n}\ddot{q_n}}) \times \\
& \ln(\ddot{q_n} + \tfrac{\mu}{2} \cdot \tfrac{T}{n} + O(\tfrac{1}{n^{3/2}})) + O(\tfrac{1}{n}).
\end{aligned}$$

From the Taylor expansion we have:

$$\ln(n\ddot{p_n} + \ddot{x_m}\sqrt{n \cdot \ddot{p_n}\ddot{q_n}}) = \ln(n\ddot{p_n}) + \ddot{x_m}\sqrt{\frac{\ddot{q_n}}{n\ddot{p_n}}} - \frac{\ddot{x_m}^2}{2} \cdot \frac{\ddot{q_m}}{n\ddot{p_n}} + O(\frac{1}{n^{3/2}}),$$

$$\ln(n\ddot{q_n} - \ddot{x_m}\sqrt{n \cdot \ddot{p_n}\ddot{q_n}}) = \ln(n\ddot{q_n}) - \ddot{x_m}\sqrt{\frac{\ddot{p_n}}{n\ddot{q_n}}} - \frac{\ddot{x_m}^2}{2} \cdot \frac{\ddot{p_m}}{n\ddot{q_n}} + O(\frac{1}{n^{3/2}}),$$

$$\ln(\ddot{p_n} + \frac{\mu}{2} \cdot \frac{T}{n} + O(\frac{1}{n^{3/2}})) = \ln \ddot{p_n} + \frac{\mu T}{2n\,\ddot{p_n}} + O(\frac{1}{n^{3/2}}),$$

$$\ln(\ddot{q_n} + \frac{\mu}{2} \cdot \frac{T}{n} + O(\frac{1}{n^{3/2}})) = \ln \ddot{q_n} + \frac{\mu T}{2n\,\ddot{q_n}} + O(\frac{1}{n^{3/2}}).$$

Putting the last relations into the expression for $\ln I_{n,m}$ we get:

$$\ln I_{n,m} = \ln \frac{1}{\sqrt{2\pi n \ddot{p_n} \ddot{q_n}}} - \frac{\ddot{x_m}^2}{2} + \mu T + O(\frac{1}{n^{1/2}}),$$

which proves theorem 1.

Theorem 2.

$$\lim_{n\to+\infty} \sum_{m=m^*}^{n} \frac{n!}{m!(n-m)!} \cdot \bar{p_n}^m \bar{q_n}^{n-m} = \exp(\mu T) \cdot \Phi\left(\frac{\ln \frac{S}{X} + \mu T + \frac{\sigma^2}{2} T}{\sigma\sqrt{T}}\right).$$

Proof. We have

$$\ddot{x_m^*} = \frac{m^* - n \cdot \ddot{p_n}}{\sqrt{n \cdot \ddot{p_n} \ddot{q_n}}} \to \frac{\ln \frac{X}{S} - \mu T - \frac{\sigma^2}{2} T}{\sigma\sqrt{T}},$$

$$\ddot{x_n} = \frac{n - n \cdot \ddot{p_n}}{\sqrt{n \cdot \ddot{p_n} \ddot{q_n}}} \to +\infty \qquad \text{for} \quad n \to +\infty.$$

Due to the theorem 1 we have

$$\sum_{m=m^*}^{n} \frac{n!}{m!(n-m)!} \cdot \bar{p_n}^m \bar{q_n}^{n-m} = \exp(\mu T) \frac{1}{\sqrt{2\pi}} \sum_{m=m^*}^{n} \exp(-\frac{\ddot{x_m}^2}{2}) \Delta x_n \left(1 + O(\frac{1}{\sqrt{n}})\right),$$

where $\Delta x_n = \ddot{x}_{m+1} - \ddot{x_m} = 1/\sqrt{n \cdot \ddot{p_n} \ddot{q_n}} \to 0$ for $n \to +\infty$.

Therefore, from the last equation we get

$$\lim_{n\to+\infty} \sum_{m=m^*}^{n} \frac{n!}{m!(n-m)!} \cdot \bar{p_n}^m \bar{q_n}^{n-m} = \exp(\mu T) \frac{1}{\sqrt{2\pi}} \int_{\frac{\ln \frac{X}{S} - \mu T - \frac{\sigma^2}{2} T}{\sigma\sqrt{T}}}^{+\infty} \exp(-\frac{S^2}{2}) dS =$$

$$\exp(\mu T) \Phi\left(\frac{\ln \frac{S}{X} + \mu T + \frac{\sigma^2}{2} T}{\sigma\sqrt{T}}\right).$$

The theorem 2 is proved.

Applying theorem 2 and (14) we can determine the limit value $C = \lim_{n\to+\infty} C_n$ of the call option price in the framework of the discrete binomial model:

$$C = \exp(-r\,T) \cdot \left[\exp(\mu\,T) \cdot \Phi\left(\frac{\ln \frac{S}{X} + \mu T + \frac{\sigma^2}{2} T}{\sigma\sqrt{T}}\right) - X \cdot \Phi\left(\frac{\ln \frac{S}{X} + \mu T - \frac{\sigma^2}{2} T}{\sigma\sqrt{T}}\right)\right]. \tag{17}$$

For instance, assuming condition $\mu = r$ we get the Black-Scholes option pricing formula (1).

Formula (17) determines the call option price in the limits of the continuous Samuelson model (4) and was previously obtained [AIK]. We have shown [AIK] that when, condition $\mu = r$ is not fulfilled, the call option price calculated by the Black-Scholes option pricing formula (1) may give a noticeable bias.

5. Conclusions

Using the assumption $\mu \approx 2r$ and typical parameters (see below), we calculated the option prices applying the Black-Scholes formula (1) and formula (17) C and obtained systematic and essential difference between C_{BS} and C. This difference remains on the same level when the discount factor $d(T) = \exp(-rT)$ in (17) is changing to $\exp(-\mu T)$.

The following parameters were used in our calculations: $r = 0.08$, $\mu = 0.16$, $\sigma = 0.2$, $S = 100$, $X = 102$, $T = 0.25$. Using the Black-Scholes formula we obtained $C_{BS} = 4.00$. Using formula (17) with $d(T) = \exp(-rT)$ we obtained $C = 5.13$, which differs from C_{BS} by 28%, and, if we will use $d(T) = \exp(-\mu T)$, our formula gives 5.02, which differs from C_{BS} by 25.5%.

Here the stock prices changes were based on the Samuelson model (4). However, the following model can be used:

$$S(t) = S(0) + \mu t + \sigma z(t). \tag{18}$$

This model for $\mu = 0$ coincides with a model proposed and first used for the determination of the option prices in the Bachelier's paper [BaL]. Maybe, in this sense (18) is quoted as the Bachelier's model (see, for example, [Shir]). Equation (18) can be rewritten in the form of the stochastic process (compare it with (4)):

$$dS = \mu dt + \sigma dz(t). \tag{19}$$

Using formula (2) and model (18) one can obtain the following formula for the call option prices:

$$C_{BA} = d(T)\left[(S + \mu T - X)\Phi\left(\frac{S + \mu T - X}{\sigma\sqrt{T}}\right) + \frac{\sigma\sqrt{T}}{\sqrt{2\pi}} e^{-\frac{(S+\mu T-X)^2}{2\sigma^2 T}}\right]. \tag{20}$$

It must be noted that Eq. (20) is not explicitly presented in [BaL]. However, the scheme in [BaL] (under the assumption $\mu = 0$) precisely corresponds to (20) (see also [Shir]). Moreover, in [BaL] no account was taken of the discount factor of the option prices on the expiration date, i.e. $d(T) = 1$.

One can see that formula (22) in [AIK] presents the specific case of (20) for $d(T) = \exp(-rT)$.

Using formula (20) with the same parameters as mentioned above we obtained the following results: $C_{BA} = 4.98$ for $d(T) = \exp(-rT)$ and $C_{BA} = 4.88$

for $d(T) = \exp(-\mu T)$. They are close to those obtained with the help of formula (17) and significantly differ from those obtained for the Black-Scholes scheme.

In order to compare the actual option price (the payoff) at the expiration date T with the market option price, with the option price corresponding to the Black-Scholes model and to our model (17), we used the option price database. The call option payoffs were calculated using the formula $P = \max[S(T) - X, 0]$. The values of the expected return μ in (17) were calculated on the basis of the daily closing prices of basic stocks using the least squares method.

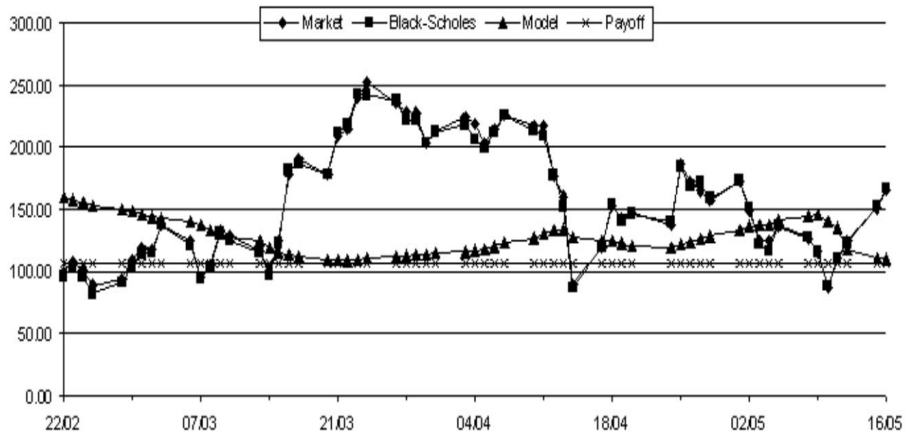


Fig. 1. Dynamics of call option payoffs for the S&P500 index, for market option prices, for option prices determined by the Black-Scholes model and by model (17) for the period 22.02.2000–16.05.2000

We see in Fig. 1 that the market prices and the prices corresponding to the Black-Scholes model coinside in most cases of the presented dates. However, they significantly differ from the call option payoffs. At the same time, the estimation of option prices by formula (17) is more close to call option payoffs compared with the market option prices and the option prices corresponding to the Black-Scholes model. Similar results were obtained for other options.

Acknowledgements

We are grateful to G.V. Lukin for his help in calculations of option prices presented in the paper.

We are grateful to Prof. I. Prigogine for encouragement and support. We acknowledge the support of the International Solvay Institutes for Physics and

Chemistry and the Belgian Government through the Interuniversity Attraction Poles.

References

[CR1] Cox J.C., Rubenstein M. *Options Markets*. Englewood Cliffs.N.J., Prentice-Hall, 1985.

[Mer] Merton R.C. *Continuous-Time Finance*. Basil Blackwell Inc., Oxford, 1990.

[LaG] Lawrence Galitz. *Financial Engineering*. Pitman Publishing, London, 1996.

[Hul] Hull J. *Options, Futures and Other Derivatives*. Prentice-Hall, 1997.

[OBS] O'Brien T., Srivastava S. *Financial Analysis and Security Trading*. Carnegie Mellon University, 1995.

[Hau] Haugen R.A. *Modern Investment Theory*. Prentice-Hall, 1996.

[AIK] Antoniou I., Gorshkov Yu.S., Ivanov V.V. and Kryanev A.V.: Forecasting Financial Derivative Prices, *Chaos, Solitons and Fractals*, 11, 2000, pp. 223-229.

[BS1] Black F. and Scholes M. The Pricing of Options and Corporate Liabilities. *J.Political Economy*, 81, no. 3, 1973, pp. 637-654.

[CRR] Cox J.C., Ross S.A., Rubenstein M. Option Pricing: A Simplified Approach. *Journal of Financial Economics*, 7, 3, 1979, pp. 229-263.

[Sha] Sharpe W.F. *Investments*. Prentice Hall, 1978.

[RjB] Rendleman R.J., Jr., and Bartter B.J. Two-State Option Pricing. *Journal of Finance,* **34**, 5, 1979, pp. 1093-1110.

[NeRa] Nelson D.B. and Ramaswamy K. Simple Binomial Processes as Diffusion Approximations in Financial Models. *Review of Financial Studies*, **3**, 3, 1990, pp. 393-430.

[Fel] Feller W. *An Introduction to Probability Theory and Its Applications*, v. 1, Ed. John Wiley, 1970.

[BaL] Bachelier L. Theorie de la Spéculation. *Ann. Ecole Norm. Sup.*, 1900, v.17, p. 21-86 (Reprinted in: *Random Character of Stock Market Prices*. Ed. by P.H.Cootner, Cambridge, Massachusetts: MIT Press, 1964, p. 17-78).

[Shir] Shiryaev A.N. Probabilistic-statistical models of financial indexes dynamics. *Review of applied and industrial mathematics*. Moscow, TVP publ., vol. **2**, N% 4, 1995, pp. 527-555 (in Russian).

Wavelet Neural Network and its Application to ECG Analysis

A.Yu. Bonushkina[1], V.V. Ivanov[1,2], A.V. Selin[1] and P.V. Zrelov[1]

[1] Laboratory of Information Technologies, Joint Institute for Nuclear Research, Dubna, 141980, Russia
[2] International Solvay Institute for Physics and Chemistry, CP-231, ULB, Campus Plaine, Bd. du Triomphe, 1050, Brussels, Belgium

Abstract. A new approach for the approximation of raw ECG data and for the detection of ECG extremums of preassigned widths is proposed. It is based on the usage of a specialized feed-forward neural network representing the ECG signal in the form of wavelet series with specially chosen scaling function. This provides the effective smoothing of ECG data, filtering a high-frequency noise, fixing the position of main extremums and the compression of raw data. The efficiency of the approach is demonstrated on model and real data.

Introduction

In recent years there is a constant interest to the analysis of electrocardiograms (ECGs) applying modern mathematical methods based on artificial neural networks and wavelets [1-5].

The ECG time-series represent measurements $f_1, f_2, \ldots, f_N$ taken sequentially in equal time intervals. The problem is to construct the approximation function in such a way that the contribution of a random noise would be significantly suppressed, and the main extremums corresponding to P-wave, QRS-complex and T-wave would be described by a smooth curve. Moreover, such approximation should allow to perform most possible identification of these waves.

Among the most convenient technique for dealing with this problem is the wavelet analysis, which permits to represent a function approximating raw ECG-data as wavelet series expansion. Choosing wavelets with specific properties one can filter out a high frequency noise, extract extremums of definite width and amplitude, and fix their position in time.

1 Method

1.1 Wavelet Decomposition

Here we present the main concept of the wavelet analysis. The details can be found, for instance, in [6, 7].

The function $f(x) \in L_2(R)$ can be represented in terms of shifts and dilations of a low-pass scaling function $\phi(x)$ and bandpass wavelet $\psi(x)$:

$$f(x) = \sum_{k \in Z} c_k^J \phi(2^J x - k) + \sum_{j \geq -J,\, k \in Z} \sum d_k^j \psi(2^j x - k). \quad (1)$$

The first term in (1) is an approximation to $f(x)$ at resolution J. The higher J corresponds to the analysis with a higher resolution. The reduced sum of the expansion (1) at level J is:

$$f_J(x) = \sum_k c_k^J \phi(2^J x - k). \quad (2)$$

The expansion coefficients in (1) depend on the choice of wavelet basis. For orthogonal wavelets they can be represented as:

$$c_k^J =< f, \phi(2^J x - k) >, \quad d_k^j =< f, \psi(2^j x - k) > . \quad (3)$$

In the case of biorthogonal wavelets they are:

$$c_k^J =< f, \phi^*(2^J x - k) >, \quad d_k^j =< f, \psi^*(2^j x - k) >, \quad (4)$$

here symbol "*" denotes so-called dual functions.

The function $\phi(x)$ can be chosen as the m-th order cardinal B-splines N_m, where m is an arbitrary positive integer [6]. In this case the scaling function implies the biorthogonal basis and the expansion coefficients can be calculated using (4).

The first order cardinal B-spline N_1 is the characteristic function of the unit interval [0,1]. The higher order splines N_m for $m \geq 2$ are recursively obtained using the (integral) convolution:

$$N_m(x) = \int_0^1 N_{m-1}(x - t)dt.$$

The m-order B-spline has a compact support $[0, m]$, is positively determined and is symmetrical regarding the center of support (where it has the maximum). In each interval $[k, k+1]$ $(0 \leq k < m)$ $N_m(x)$ is the polynomial of $(m-1)$-order. These peculiarities provide remarkable smoothing properties of B-splines (see details in [6]).

Fig. 1 and 2 show the scaling function $\phi(x)$ and the dual function $\phi^*(x)$ of the cubic B-spline.

1.2 Algorithm

Suppose the ECG-signal in the range of extremums can be fitted by the second order polynomial $f(x) \approx a + b(x - \xi)^2$. The corresponding scaling function is the 3-rd order B-spline:

$$N_3(x) = \begin{cases} x^2/2, & \text{if } x \in [0, 1); \\ -x^2 + 3x - 3/2, & \text{if } x \in [1, 2); \\ x^2/2 - 3x + 9/2 & \text{if } x \in [2, 3); \\ 0 & \text{elsewhere.} \end{cases}$$

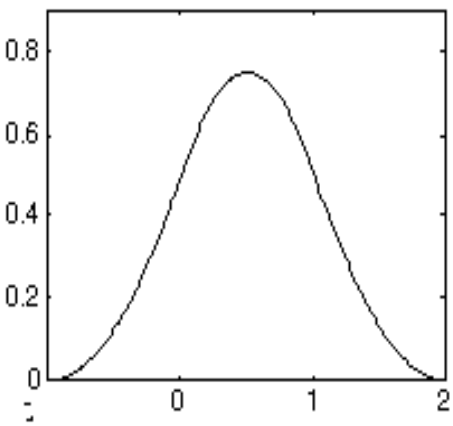


Fig. 1. Biorthogonal B-spline scaling function

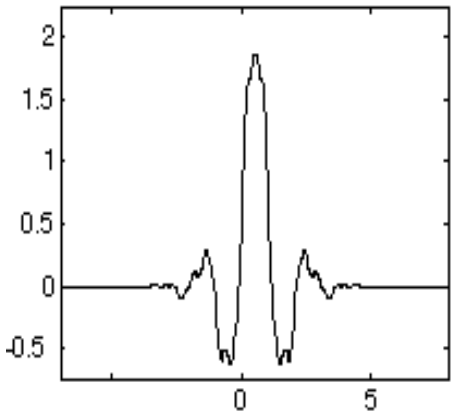


Fig. 2. Dual scaling function of B-spline

A general scheme for raw data processing is as follows.

Level J corresponding to the approximation with maximal accuracy can be estimated using the following expression:

$$J = [\log_2 N] + 1,$$

where N is the number of points f_i in the analyzing sample.

However, aiming to suppress a high-frequency noise, we choose the lower level $j_1 < J$ which must correctly describe extremums containing n points. This level j_1 can be estimated using the formula:

$$j_1 = [\log_2 \frac{N}{n}] + 1.$$

Applying now the "moving average" scheme [6] or using a neural network (see below) the expansion coefficients (2), corresponding to the chosen level j_1, are calculated. All coefficients smaller than some fixed threshold are assumed to be zero. As each wavelet has a limited support, the thresholds can be verified depending on the current time (position). The filtered signal, formed after such "cleaning" procedure, is free of a random noise.

An algorithm for the searching of extremums is the following. The number of non-zero coefficients is limited and is equal to $2^{j_1} + 2$. For each $k = -2, -1, \ldots, 2^{j_1} - 1$ the value

$$\xi_k = \frac{1}{2^{j_1}} \left(k + \frac{c_{k-1} - c_{k-2}}{2c_{k-1} - c_k - c_{k-2}} \right)$$

is calculated. In the case when

$$\xi_k \in \left[\frac{k}{2^{j_1}}, \frac{k+1}{2^{j_1}}\right],$$

we may have an extremum at the coordinate ξ_k. When $c_{k-1} - c_{k-2} > 0$, then we have a maximum, and, if $c_{k-1} - c_{k-2} < 0$, we have a minimum. When the numerator and the denominator are both equal to zero, the signal does not change in $[\frac{k}{2^{j_1}}, \frac{k+1}{2^{j_1}}]$, and there is no an extremum in this interval.

Then a level $j_0 < j_1$ corresponding to "slow" changes of the ECG signal with a period approximately equal to the inter-beat interval is chosen, and in the analogues way corresponding coefficients are calculated. In the set of the obtained extremums only those are taken into account for which absolute values of difference between "fast" and "slow" approximations exceed a certain threshold.

Details of application of this scheme to the analysis of real ECG data are given below.

1.3 The wavelet neural network

According the first formula in (4), the calculation of coefficients of the decomposition (2) needs the integration with function represented in Fig.2. There is no the explicit analytical presentation of this function – it can be obtained only numerically applying the Fourier transform for each set of new points.

In order to avoid complicated problems connected with the numerical calculation of the dual function and with construction of the quadrature formulas for this function, we may use another approach.

As we have a limited number N of values $f(x)$ ($f_p = f(x_p), p = 1, 2, ..., N$), concerning the analyzing sample of the ECG signal, the expansion coefficients can be obtained by minimizing the following functional:

$$E_N(\{\omega_k\}) = \sum_{p=1}^{N} \left[f_p - \sum_k \omega_k N_3(2^j x_p - k) \right]^2, \tag{5}$$

where ω_k denotes the coefficients c_k^J.

The functional (5) can be interpreted as the error functional of a feed-forward neural network of a specific architecture (see below). The series approximating the ECG signal at the level j

$$f_j(x) = \sum_k \omega_k N_3(2^j x - k). \tag{6}$$

can be re-written in terms of the three-layered perceptron. An argument, time x, is feed to its input and the filtered signal is taken from its output. The architecture of network is given in Fig. 3. It looks similar to a neural network developed in [10].

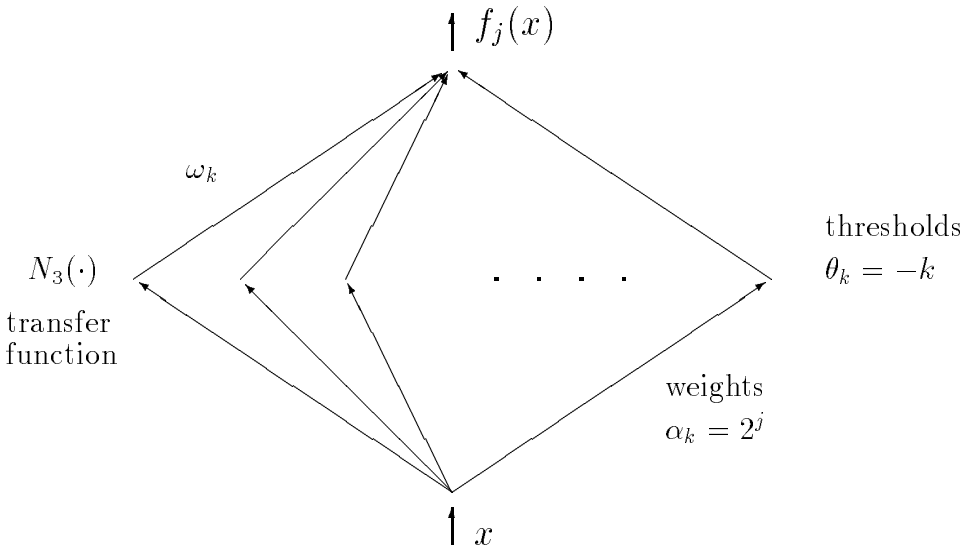


Fig. 3. Wavelet neural network

The number of neurons in the hidden layer is equal to the number of terms in the expression (6) and each k-th neuron corresponds to the k-th term in (6). Weights and thresholds between the input layer and the hidden layer are fixed and are correspondingly equal to $\alpha_k = 2^j$, $\theta_k = -k$. The transfer functions of hidden neurons are the 3-rd order B-splines. The thresholds between the hidden layer and the output neuron are taken equal to zero, and each k-th coefficient corresponds to the k-th weight.

The weights ω_k, which play the role of expansion coefficients, are obtained by the neural network "training"on data corresponding to the analyzing sample and applying the standard back-propagation method [11].

We transform the analyzing sample of the ECG data to the time interval $[0,1]$. Only terms concerning $k = -2,-1,\ldots,2^j - 1$ contribute into the expression (6), and the coefficients corresponding to other k are assumed to be zero.

2 Application to model and real data

2.1 Model example

To estimate the accuracy of an extremum localization, and to evaluate the robustness of the algorithm we considered the following model example.

Let a signal changes in time as

$$f(x) = e^{-\left(\frac{x-0,45}{0,1}\right)^2}, \tag{7}$$

i.e. it has the maximum at the point $x = 0.45$ with amplitude 1 and width 0.2. Using (7) we calculated 50 "measurements" of the signal at equidistant moments of time interval $x \in [0,1]$. A random Gaussian noise with mean value 0 and dispersion 0.1 was added to each term of this time-series; this corresponds to the ratio of noise/signal (in extremum) 10%. The neural network was trained on these data and its learning was going quit quickly: practically after 30 – 50 training epochs a mean squared error between the network output and a target value was less 0.02 %.

In Fig. 4 the approximation of simulated data are shown by the continuous curve. The approximation level j was taken equal to 3. It can be calculated using formula $j = [\log_2 \frac{T}{\Delta}] + 1$, where $T = 1$ is the interval where the function is defined, $\Delta = 0.2$ is the maximum width. The coordinate of the global extremum for the approximating curve equals to 0.441, which corresponds to very small relative error.

2.2 Real data analysis

Figures 5 demonstrate some details concerning the ECG data analysis: a) few successive periods of raw ECG-data, b) the $j_1 = 6$ approximation (solid curve) superimposed with the $j_0 = 0$ approximation (dot-and-dashed curve), c) the difference between approximations with $j_1 = 6$ and $j_0 = 0$, e) the relative error corresponding to the approximation with $j_1 = 6$.

Figures 6 a÷6 d show the details concerning the approximation of real data and fixing of the ECG main extremums. The extremums we are looking for are marked by "1","2","3" and "4". The size N of sample using for the analysis depends on a frequency of discretization and in our case it was equal to 250 points. This is somewhat greater than a distance between two neighboring peaks "1". In addition, it was assumed that the width of the narrowest peak must contain not less than 5 points, i.e. the needed resolution is given by the approximation of level 6: $[\log_2 \frac{250}{5}] + 1$.

The dot-and-dashed curve shows the approximation with $j_0 = 0$. This approximation plays a special role in the analysis. It can be considered as a so-called "baseline" or "null-line" accepted in ECG analysis. And hence the amplitudes of main extremums should be measured relative to this curve.

Additional identification of extremums can be performed using, for instance, values of amplitude and curvature and also their order in time. Thus, the extremums "1","2","4" in Fig. 6 could be classified using thresholds between levels $j_0 = 0$ and $J_1 = 6$: for "1" it equals 400, for "2" and "4" it is 100. However, the problem of extremums identification goes out of limits of this work.

The supplementary result of the developed method is the compression of raw ECG data, because only 65 coefficients should be used (and hence can be stored) in average to effectively represent of 250 points of the ECG. Thus, the compression factor equals to about 4.

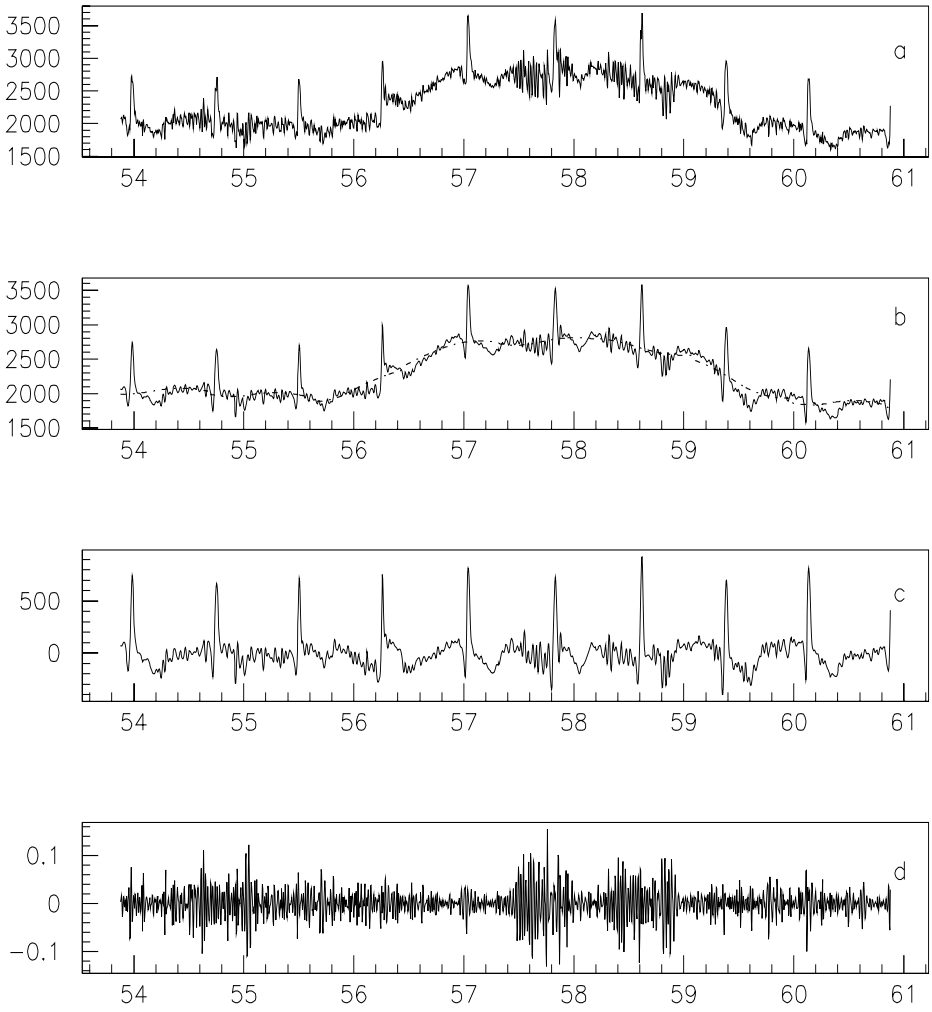


Fig. 4. Results on the ECG data analysis: a) few successive periods of real ECG, b) $j_1 = 6$ approximation (solid line) superimposed by $j_0 = 0$ approximation (dot-and-dash line), c) difference between the approximations with $j_1 = 6$ and $j_0 = 0$, d) relative error corresponding to $j_1 = 6$ approximation

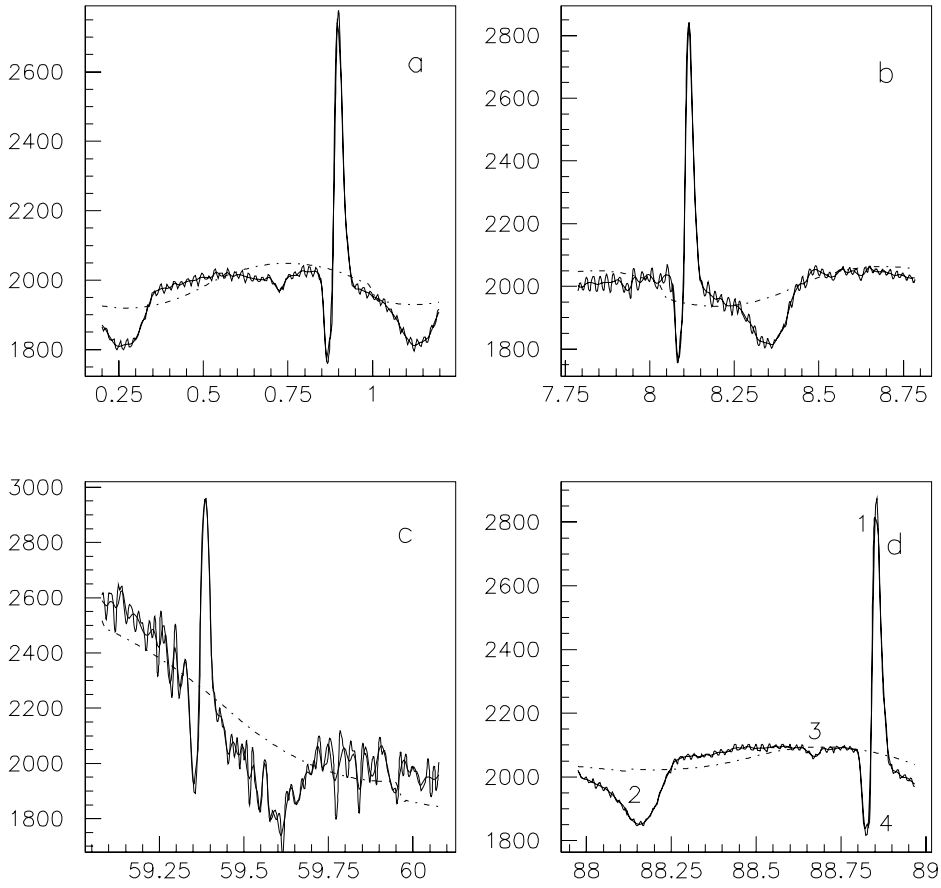


Fig. 5. The approximation of the ECG data in different periods

Conclusions

The results of this work can be summarized as follows.

A new approach for the ECG-data analysis has been developed. It is based on the specialized feed-forward neural network representing the ECG signal in the form of wavelet series with specially chosen scaling function. This permits: a) to filter out a high-frequency noise, b) to fix the positions of main extremums, and c) to compress raw ECG data with compression factor 4 and

without losing the significant information. The efficiency of the network has been demonstrated on model and real data.

This approach permits to organize the uninterrupted processing of analyzing ECG data and provides also the detection of the extremums of preassigned widths.

Acknowledgments

This work has been partly supported by the Commission of the European Community within the framework of the EU – RUSSIA Collaboration, in accordance with ESPRIT Project CTIAC-21042.

References

1. A. Iwata, Y. Nagasaka, S. Kuroyanagi, N. Suzumura: *"Real time ECG data compression using dual three layered neural networks for digital holter monitor"*, Artificial Neural Networks, Elsevier Science Publishers B.V. (North–Holland), 1991, p.1673 – 1676.
2. K. Zhu, P.D. Noakes and A.D.P. Green: *"Training Neural Networks for ECG Feature recognition"*, in Proceedings of International Neural Network Conference, July 9 – 13, 1990, Paris, p.137 – 140.
3.] C.D.Nugent, J.A.C.Webb, M.McIntyre, N.D.Black, G.T.H.Wright: *Computerized electrocardiology employing bi-group neural networks*, Artificial Intelligence in Medicine 13, (1998), p.167.
4. Q.Xue, Yu.H.Hu, W.J.Tompkins: *Neural-Network-Based Adaptive Matched Filtering for QRS Detection*, IEEE Transactions on Biomedical Engineering, vol. 39, No. 4, April 1992.
5. H.Holst, M.Ohlsson, C.Peterson, L.Edenbrandt: *A Confident Decision Support System for Interpreting Electrocardiograms*, LU-TP 98-31, 1998.
6. Ch. Chui: *"An Introduction to Wavelets"*, Academic Press, Inc., 1992.
7. I. Daubechies: *"Orthonormal Bases of Compactly Supported Wavelets"*, Comm. on Pure and Appl. Math., vol. XLI 909-996, 1988.
8. M. Ueda and S. Lodha: *"Wavelet: An Elementary Introduction and Examples"* Baskin Center for Computer Engineering & Information Sciences, University of California, Santa Cruz, CA 95064, USA, UCSC-CRL 94-47, January 17, 1995.
9. Y.C. Paty and P.S. Krishaprasad: *"Analysis and Synthesis of Feed-forward Neural Networks Using Discrete Affine Wavelet Transformation"*, IEEE Trans. On Neur. Network, vol. 4, no. 1, 1993.
10. V. Basios, A.Yu. Bonushkina and V.V. Ivanov: *"A Method for Approximating One-Dimensional Functions"*, Comput. Math. Appl.", Vol. **34**, No. 7/8, p.687, 1997.
11. D.E.Rumelhart, G.E.Hinton, R.J.Williams: *"Learning Internal Representations by Error Propagation"* in D.E.Rumelhart and J.L.McClelland (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations. MIT Press (1986).

Similarity Searches Using a Recursive String Parsing Algorithm

Ulrich Günther

Department of Computer Science, University of Auckland, Auckland, New Zealand

Abstract. Similarity queries present a computational problem of great practical significance. Its applications range from Internet search engines and other large databases to genetic research and computer vision. Similarity searches compare objects from a large repository (e.g., a database) to each other or to a specified sample. This paper focuses on string objects, such as text files, or search terms submitted to an Internet search engine. Similarity queries require a (dis)similarity measure that takes two strings as input and outputs a number that indicates how "similar" the two strings are. The conventional approach to this problem transforms the strings into a feature space, where a metric is applied. This paper presents a fundamentally different approach, derived from a string measure developed by Titchener [18, 17, 21, 20]. The resulting similarity measure seems to be well suited to a wide range of practical problems. It is directly related to the extent of pattern sharing between two strings.

1 Introduction

Many real-world search applications (such as large databases, Internet search engines, DNA matching) require or desire a facility that permits similarity searches. For example, a user that can't quite remember whether New Zealand's biggest city is spelled "Auckland" or "Aukland" (a common misspelling) could simply submit the latter to an Internet search engine and still get all the links to the former. Similarly, one might search for "Auckland University" and still find pages relating to "The University of Auckland". Another scenario is the automated "clustering" of records in a database [5], which is often done to permit users to find "a document similar to this one", but may also in turn be used to improve the performance of other similarity search tools [7].

We may assume that database searches or clustering operations are carried out on strings from some finite alphabet, e.g., ASCII or binary. At the heart of the task is the need for a similarity measure that takes two strings as input and returns a number that ranks the strings according to their similarity.

Let x and y be the strings that are to be compared. E.g., y could be the text content of a web page and x a search string entered by a search engine user. Many conventional approaches (e.g., [14, **?**,12]) to similarity measures

are based on a two-step process: first, both x and y are transformed into a feature space. In the second step, a metric (often Euclidean) is used to measure similarity.

The feasibility of such similarity measures thus depends on the choice of a suitable transform. This choice is often a heuristic one, and as a general rule it is easier to transform into high-dimensional feature spaces than into low-dimensional ones. Unfortunately, the complexity of computing a meaningful metric increases with the dimension of the feature space, a phenomenon that has become known as the *dimensionality curse* [22, 10].

It is thus not surprising that other approaches (e.g., [4]) have been proposed. The author of [22] states that geometrically optimal similarity measures are routinely replaced with less precise metrics, at the risk of yielding false hits. The measure presented here goes in this direction. However, unlike conventional approaches it uses neither a transform into a feature space, nor a Euclidean metric. It does however acknowledge the underlying principle: that similarity is a fuzzy concept, and that we may thus have to accept false hits.

This paper is based on a fundamental assumption: we assume that y contains information similar to the information in x if a large proportion of the substrings (or patterns) in x can be found in y. The term *large proportion* is fuzzy by default, whereas the term *substrings* is at least well defined.

We could thus test all substrings in x for occurrence in y, but this is often impractical for long x, as the number of distinct substrings in x is proportional to $|x|^2$. In many real-life cases (clustering), it is simply impossible to search for all these substrings in y. This is where conventional approaches use a feature transform, which tends to involve heuristics, i.e., the de-facto exclusion of many possible pattern combinations, or the restriction to certain substring patterns, e.g., whole words in a phrase.

The method presented here requires no such heuristics and instead relies on a recursive parsing mechanism. Compared to other methods, it is not overly computationally intensive ($\approx O(|xy|)$), and a large deal of the computation may be done in advance during indexing. The method is based on the T-complexity/T-information measures developed by Titchener [18, 17, 21, 20].

2 T-Codes, T-decomposition, T-complexity, and T-information

The similarity measure proposed in this paper is a direct derivate of Titchener's T-information measure. The T-information measure computes from his T-complexity measure, which in turn computes from the construction parameters of T-Code sets. T-Code sets feature an interesting duality relationship with the set of all finite strings. So let's start with T-Codes.

2.1 T-Codes

About 15 years ago, Titchener proposed a system of recursively constructed codes, the T-Codes [2], which he subsequently extended to the *generalized* T-Codes [8]. These are complete coding trees (similar to Huffman coding trees) that are constructed from finite alphabets using a copy-and-append technique called T-augmentation. A detailed treatment of T-Codes and T-augmentation may be found in [19]. T-Code sets are constructed from a finite alphabet S. For applications such as text similarity searches one would typically use ASCII or the binary alphabet $\{0,1\}$. S acts as the T-Code set at *T-augmentation level* $n = 0$, such that all the letters in S are primitive *codewords*. The construction of higher-level T-Code sets from S requires a series of *T-augmentations*: Given a T-Code set $S^{(k_1,k_2,\ldots,k_n)}_{(p_1,p_2,\ldots,p_n)}$ at level n, a T-augmentation constructs the set $S^{(k_1,k_2,\ldots,k_{n+1})}_{(p_1,p_2,\ldots,p_{n+1})}$ at level $n+1$ as follows:

$$
\begin{aligned}
&S^{(k_1,k_2,\ldots,k_{n+1})}_{(p_1,p_2,\ldots,p_{n+1})} = \\
&\left\{ z | z = p^i_{n+1} a, \text{ where } a \in S^{(k_1,k_2,\ldots,k_n)}_{(p_1,p_2,\ldots,p_n)} \backslash \{p_{n+1}\}, 0 \le i \le k_{i+1} \right\} \\
&\cup \left\{ p^{k_{n+1}+1}_{n+1} \right\}, \qquad (1)
\end{aligned}
$$

where $p_{n+1} \in S^{(k_1,k_2,\ldots,k_n)}_{(p_1,p_2,\ldots,p_n)}$ and k_{n+1} is a positive integer.

Example: Table 1 illustrates the construction of the T-Code set $S^{(1,1,3)}_{(0,1,01)}$ from the binary alphabet S by means of three T-augmentations with T-prefixes 0, 1, and 01, and T-expansion parameters 1, 1, and 3 respectively. At T-augmentation level 1 and 2, the intermediate T-Code sets $S^{(1)}_{(0)}$ and $S^{(1,1)}_{(0,1)}$ are generated in the course of the construction.

2.2 T-Codes and finite strings – two sides of the same coin

The example already illustrates an important property of T-Codes: Consider the two longest codewords of the top-level T-Code set. These contain, in sequence, all the T-prefixes that we have chosen in the construction of the set. Moreover, the number of times that they appear adjacent to each other reflects the T-expansion parameters chosen[1].

Nicolescu [6, **?**] showed that the longest codewords are unique for each T-Code set. He further showed that, given one of the longest codewords, it is always possible to derive the associated T-Code set. Nicolescu proposed an

[1] This is, strictly speaking, not always true, as many T-Code sets can be constructed by more than one set of T-prefixes and T-expansion parameters (called a *T-prescription*). However, these T-prescriptions are equivalent and Nicolescu showed that one can always derive one where all T-expansion parameters are one less than a prime number. This T-prescription is unique and results from the T-decomposition algorithm explained in this paper [6, **?**].

	T-augmentation level			
n	0	1	2	3
k_n	$k_0 = 1$	1	1	3
set	S	$S^{(1)}_{(0)}$	$S^{(1,1)}_{(0,1)}$	$S^{(1,1,3)}_{(0,1,01)}$
	0	∅	–	–
	1	1	1	–
		00	00	00
		01	01	01
			–	–
			11	11
			100	100
			101	101
				–
				–
				0100
				01
				–
				0111
				01100
				01101
				–
				–
				010100
				01
				–
				010111
				0101100
				0101101
				–
				–
				01010100
				01010101
				–
				01010111
				010101100
				010101101

Table 1. The construction of the T-Code set $S^{(1,1,3)}_{(0,1,01)}$ from the binary alphabet S by means of three T-augmentations with T-prefixes $p_1 = 0$, $p_2 = 1$, and $p_3 = 01$, and T-expansion parameters $k_1 = 1$, $k_2 = 1$, and $k_3 = 3$.

algorithm now called *T-decomposition*, which will be explain shortly. He also showed that for each finite string, there is a T-Code set such that that string is one of its longest codewords. It is apparent from the example that the longest codewords of a T-Code set only differ in their last symbol.

2.3 T-decomposition

Using Nicolescu's method, one may take a string and deduce its associated T-Code set. This is done as follows: let z be a string from S^+ and $a \in S$. Further let za be the string that we wish to decompose in order to find the T-Code set for which it is one of the longest codewords. Then proceed as follows:

1. make S the current T-Code set.
2. decode za left-to-right and identify the codeword boundaries with respect to the current T-Code set. If the string decodes as a single codeword, stop here. If it decodes as more than one codeword, identify the second-to-last codeword in the decoding and make it the T-prefix for the next T-augmentation.
3. count the length of the run of the T-prefix that we have just identified, counting the second-to-last codeword and all adjacent copies to the left of it. This is the T-expansion parameter.
4. T-augment your current T-Code set with the T-prefix and T-expansion parameter you just found.
5. make the resulting set your current T-Code set and continue at step 2.

Example [19]: Let $z = 011000101010$ and $a = 1$, s. t. $za = 0110001010101$. If we decode za over $S = \{0,1\}$, we obtain the following codeword boundaries indicated by a dot:

$$za = 0.1.1.0.0.0.1.0.1.0.1.0.1.$$

from which we identify the second-to-last codeword, which is $p_1 = 0$. There are no further adajacent copies of p_1 further to the left, hence $k_1 = 1$. Our level 1 intermediate set is thus $S^{(1)}_{(0)}$. If za is decoded over $S^{(1)}_{(0)}$, we obtain the codeword boundaries:

$$za = 01.1.00.01.01.01.01.$$

The second to last codeword is p_2, in this case $p_2 = 01$. Because there are another two adjacacent copies of p_2 to the left of the second-to-last codeword, we obtain $k_2 = 3$ as the corresponding T-expansion parameter. Hence, the level 2 set is $S^{(1,3)}_{(0,01)}$. Decoding za over it, we get:

$$za = 011.00.01010101.$$

such that $p_3 = 00$, $k_3 = 1$. The next iteration yields $p_4 = 011$ and $k_4 = 1$, respectively. The reader may wish to verify that $za = 0110001010101$ is indeed one of the longest codewords of $S^{(1,3,1,1)}_{(0,01,00,011)}$.

The last symbol (a) of the string za does not influence its T-decomposition. However, we want all symbols in our strings x and y to be taken into account in our similarity measure. Hence, for the purposes of this paper, we will regard strings under T-decomposition as *T-handles* [6], i.e., we will add a dummy symbol to their right end before carrying out a T-decomposition.

N.B.: The above algorithm is actually more verbose than it needs to be. For didactic purposes, the above example always decodes the *entire* string in each pass. In each of the passes, however, it suffices to decode the entire run of the second-to-last codeword in the string. This is all the information we need in order to deduce the next higher level set.

Let the T-Code decoder start part-way through the string, some distance to the left of the run of the second-to-last codeword that we wish to find. If full synchronisation (decoding of the correct codeword boundaries with respect to the current T-Code set) can be achieved before the run of the second-to-last codeword is decoded in a pass, then the results will be just as valid as if the entire string had been decoded.

T-Codes are known for their inherent, well-understood self-synchronisation mechanism. In fact, a T-Code decoder is able to tell when it is synchronised. Thus, it can start decoding at a reasonable distance from the end of the string. If this distance proves to be insufficient, the decoder can simply increase the distance and try again, although this would be unnecessary in most practical cases. For most strings, this results in the T-decomposition having a computational complexity of $O(|z|)$.

2.4 T-complexity

Given this duality between strings and T-Code sets, Titchener observed that one could think of T-augmentation not just as a process for the construction of T-Code sets, but also for the construction of strings. Following the example of Lempel and Ziv [1], who used their string parsing algorithm to derive a "production complexity" for strings, Titchener derived a measure C_T from the T-decomposition algorithm and called it *T-complexity*. Lempel and Ziv chose to define their production complexity in terms of the number of steps required to construct the string z using the algorithm they had proposed. Similarly, Titchener chose to define T-complexity as the weighted number of steps (T-augmentations) required to construct a string. $C_T(z)$ is defined as:

$$C_T(z) = \sum_{i=1}^{n} \log_2(k_i + 1), \tag{2}$$

where the weights $\log_2(k_i+1)$ are given by the T-expansion parameters, which are obtained from the T-decomposition[2] of za, where $a \in S$.

[2] N.B.: $C_T(z)$ is invariant under a change of T-prescription.

2.5 T-information

Further, Titchener observed empirically that the maximum T-complexity for all strings of length ℓ, $max\{C_T(z)|z \in S^+ \wedge |z| = \ell\}$, converges towards the logarithmic integral of ℓ as ℓ increases. Strings with such maximum C_T for a given length (or approximate length) are easily constructed by always choosing 1 as the T-expansion parameter and (one of) the shortest available codeword(s) in the respective current T-Code set as the next T-prefix.

Titchener applied the inverse logarithmic integral to $C_T(z)$, calling the resulting measure $I_T(z)$ the *T-information* of za. He showed that this approximate proportionality also holds for other classes of string, including strings generated by natural random processes or pseudorandom algorithms.

3 T-information and similarity

The T-complexity (and hence the T-information) of a string of a given length depends heavily on the repeated occurrence of patterns within the string. If certain substrings are repeated within the string, the T-Code decoder will generally synchronize within these substrings during T-decomposition passes. Hence, these substrings will be parsed in a similar way. In later passes, when the rightmost results of this parsing have become the new T-prefixes, the copies of the original substring further to the left will consist mostly of longer codewords because they contain T-prefixes that have already been identified. "Longer codewords" implies "fewer codewords", which in turn implies "fewer total number of passes" and thus "lower T-complexity" and "lower T-infomation".

Consider the concatenated string xy. Following the previous argument, if y contains many patterns that also appear in x, we would expect to find that $C_T(xy)$ and $I_T(xy)$ are lower than if x and y shared few patterns.

For example (based on an ASCII alphabet and rounded to two digits after the decimal point), C_T("Auckland") $= 8$, C_T("Aukland") $= 7$, C_T("Brussels") $= 7$. However, C_T("AuklandAuckland") $= 9$ whereas C_T("AuklandBrussels") $= 12$! This is the underlying principle that this paper exploits.

The effect becomes even stronger when one considers the T-information for the above examples:
I_T("Auckland") $= 14.58$, I_T("Aukland") $= 12.00$, I_T("Brussels") $= 12.00$, I_T("AuklandAuckland") $= 17.35$, I_T("AuklandBrussels") $= 26.58$.

How can this be turned into a useful measure for similarity comparisons? One problem that appears almost immediately is that C_T and I_T generally increase as the length of the string increases. If we want to look for pattern sharing between our query string x and a range of data strings y of differing length, it becomes difficult to compare the results without taking the length of the strings into account.

This suggests that the increase of the T-information divided by the length of the strings involved could be a feasible measure m_1:

$$m_1(x,y) = \frac{I_T(x) + I_T(y) - I_T(xy)}{|xy|}. \tag{3}$$

Experimental evidence does indeed show that m_1 is a feasible similarity measure. The author has set up a web site that demonstrates its use on a set of HTML files:

`http://www.tcs.auckland.ac.nz/~ulrich/cgi-bin/similarity.cgi`

The web site permits users to enter a search term or phrase, searches a set of files, and returns a list of those files ranked by similarity with the phrase entered by the user.

4 Conclusions

The measure m_1 introduced in this paper may be used as a similarity measure. As a very crude rule of thumb, it performs best when $|x| \gg \log(|y|)$. The reason is that shorter search patterns have a higher probability of occurrence in random y, which may happen when the repository of strings in the database is very large. Thus, if x is too short, the likelihood of false hits increases.

It is conceivable that other measures involving T-information could work better than m_1 (hence the subscript), which merely serves to demonstrate the underlying principle at this stage. Comparison with other similarity search algorithms is difficult as many specialize on a particular type of data, and performance in general may depend on the specific data repository involved. The measure proposed here looks like a good all-rounder, in particular when long search strings are involved.

Just before submission of this paper, Mark Titchener reported from Berlin in a personal communication that he had experimented in a similar direction, looking at similarity between biosequences. He reports that his experiments show "good initial results".

5 Acknowledgements

A big thanks to Mark Titchener for providing the spark and many helpful discussions on T-information. Thanks go also to Scott Wackrow for implementing the T-decomposition algorithm in software, and to Pat Riddle for putting the similarity search world into perspective for me.

References

1. A. Lempel and J. Ziv: *On the Complexity of Finite Sequences*. IEEE Trans. Inform. Theory", 22(1), January 1976, pp. 75-81.

2. M. R. Titchener, *Technical Note: Digital Encoding by Way of New T-codes*, IEE Proceedings — Computers and Digital Techniques, 131(4), 1984, pp. 151–153.
3. D. Shasha and T.L. Wang: *New Techniques for Best-Match Retrieval*, ACM Transactions on Information Systems, 8(2), April 1990, pp. 140–158.
4. K. C. Gowda and E. Diday: *Symbolic Clustering Using a New Similarity Measure*, IEEE Transactions on Systems, Man, and Cybernetics, 22(2), March/April 1992, pp. 368–378.
5. D. R. Cutting, J. O. Pedersen, D. Karger, and J. W. Tukey: *Scatter/gather: A cluster-based approach to browsing large document collections*, Proceedings of SIGIR'92, New York, 1992, pp. 318–329.
6. R. Nicolescu: *Uniqueness Theorems for T-Codes*. Technical Report. Tamaki Report Series no.9, The University of Auckland, 1995.
7. M. A. Hearst and J.O. Pedersen: *Reexamining the cluster hypothesis*, Proceedings of SIGIR'96, Zürich, 1996, pp. 76–84.
8. M. R. Titchener: *Generalized T-Codes: an Extended Construction Algorithm for Self-Synchronizing Variable-Length Codes*, IEE Proceedings – Computers and Digital Techniques, 143(3), June 1996, pp. 122-128.
9. U. Guenther: *Data Compression and Serial Communication with Generalized T-Codes*, Journal of Universal Computer Science, V. 2, N 11, 1996, pp. 769-795. `http://www.iicm.edu/jucs_2_11`
10. S. Blott and R. Weber: *A simple Vector-Approximation File for Similarity Search in High-Dimensional Vector Spaces*, ESPRIT Technical Report TR19, ca. 1997, available online from `http://www.ced.tuc.gr/Research/Reports/HERMES/Reports.htm`
11. U. Guenther, P. Hertling, R. Nicolescu, and M. R. Titchener: *Representing Variable-Length Codes in Fixed-Length T-Depletion Format in Encoders and Decoders*, CDMTCS Research Report no.44, Centre of Discrete Mathematics and Theoretical Computer Science, The University of Auckland, August 1997. `http://www.cs.auckland.ac.nz/research/CDMTCS/docs/pubs.html`.
12. P. Ciaccia, M. Patella, and P. Zezula: *Processing Complex Similarity Queries with Distance-based Access Methods*, ESPRIT Technical Report TR22, September 15, 1997, available online from `http://www.ced.tuc.gr/Research/Reports/HERMES/Reports.htm`
13. U. Guenther, P. Hertling, R. Nicolescu, and M. R. Titchener: *Representing Variable-Length Codes in Fixed-Length T-Depletion Format in Encoders and Decoders*, Journal of Universal Computer Science, 3(11), November 1997, pp. 1207–1225. `http://www.iicm.edu/jucs_3_11`.
14. T. Seidl and H.-P. Kriegel: *Optimal Multi-Step k-Nearest Neighbor Search*, Proceedings SIGMOD'98, Seattle, 1998, pp. 154-165.
15. M. R. Titchener: *A Deterministic Theory of Complexity, Information and Entropy*, *IEEE Information Theory Workshop*, February 1998, San Diego.
16. R. Nicolescu and M. R. Titchener, *Uniqueness Theorems for T-Codes*, Romanian Journal of Information Science and Technology, 1(3), March 1998, pp. 243–258.
17. M. R. Titchener, *A novel deterministic approach to evaluating the entropy of language texts*, *Third International Conference on Information Theoretic Approaches to Logic, Language and Computation*, June 16-19, 1998, Hsi-tou, Taiwan.
18. M. R. Titchener, *Deterministic computation of string complexity, information and entropy*, *International Symposium on Information Theory*, August 16-21, 1998, MIT, Boston.
19. U. Guenther: *Robust Source Coding with Generalized T-Codes*. PhD Thesis, The University of Auckland, 1998. `http://www.tcs.auckland.ac.nz/~hd.ps.gz`

20. M. R. Titchener, P. M. Fenwick, and M. C. Chen: *Towards a Calibrated Corpus for Compression Testing*, Data Compression Conference, DCC-99, Snowbird, Utah, March 1999.
21. M. R. Titchener: *A measure of Information*, IEEE Data Compression Conference, Snowbird, Utah, March 2000.
22. V. Pestov: *On the geometry of similarity search: Dimensionality curse and concentration of measure*, Information Processing Letters, 73, 2000, 47-51.

Random Fields and Multiple Markov Properties

Si Si

Faculty of Information Science and Technology
Aichi Prefectural University,
Aichi-ken 480-1198, Japan

Abstract. We have so far discussed the multiple Markov propereties for Gaussian processes and for Gaussian random fields. In this paper we introduce an extension of this notions to some class of non- Gaussian random fields; namely for a random field $X(C)$ of homogeneous chaos. We assume that $X(C)$ is expressed as an integral of a homogeneous chaos over the domain enclosed by a smooth ovaloid C with a canonical kernel introduced in the Gaussian case. Then, an N-ple Mrkov property is defined in terms of the conditional expectations which are nonlinear (unlike Gaussian case) functions of the $X(C)$ that are observed. A characterization of the N-ple Markov propery is given in such a way that the canonical kernel is a Goursat kernel of order N. As a result we can speak of computability of the best predictors.

1 Introduction

When we discuss the analysis of a stochastic process $X(t)$ and a random field $X(C)$, we usually consider operations acting on the values of $X(t)$ or $X(C)$; for example the best predictor is formed by taking a function, in general nonlinear function, of the observed values like $\varphi(X(t), t \in T)$ of $X(t)$ and $\varphi(X(C), C \in \mathbf{C})$ of random fields $X(C)$.

In order to establish the analysis of those functions, we propose to take the *innovation* of $X(t)$ or that of $X(C)$ and express the original random functions as functionals of the innovation, so that they are ready to be analyzed. Often, actually in many favorable cases, the innovation is taken to be a *white noise*. Thus, the white noise analysis has become a basic tool for the investigation of stochastic processes and random fields.

With this innovation approach in mind, we aim at defining multiple Markov properties for some non Gaussian processes or fields and studying their characterization. This notion is a generalization of the Markov properties established for Gaussian process, although we need a new idea for the definition. There, we remind that as in the case of a Gaussian process, the innovation will play a dominant role.

In what follows some background will be prepared, and see the significance of taking random fields compared to stochastic processes from the viewpoint of information theory and the theory of dependency of random complex phenomena. Needless to say, we always keep computability in mind.

2 Operations acting on processes and fields

Various kinds of operations on stochastic processes and random fields help us in the investigation of the way of dependency of those random phenomena in question, in particular multiple Markov properties.

Linear and nonlinear functions.

Assume that $X(t)$'s are Gaussian in distribution. It is not so easy to analyse functions $\varphi(X(t), t \in T)$ of those Gaussian variables directly. The most useful method is, as was mentioned before, that we first form the innovation of $X(t)$, which is always taken to be white noise, of a given process discussed in this paper, then functions of $X(t)$'s can be rephrased as functionals of the innovation. Now, one is ready to analyze those functionals. Note that the causality always holds in this case and that every operation acting on those Gaussian functionals is linear. In such a case, it is easy to check if the quantity in question is computable.

It is well known that for a weakly stationary stochastic processes the Fourier analysis plays an important role (see the Wiener theory). In particular, the spectrum is one of the characteristics of stationary stochastic processes. Also in this case, only linear transformations of the process $X(t)$ or $X(C)$ are involved; namely it is easily checked if the given quantity is computable or not.

Our aim is to discuss non-Gaussian case, so that we need a review of some known results and further background.

Transformations of time parameter

Another kind of operation for $X(t)$ and $X(C)$ is change of time parameter t and of space-time parameter C, respectively.

(1) Time shift

The simplest and in fact most important example is the time shift $T_t : s \to s+t$:

$$T_t X(s) = X(s+t). \tag{1}$$

To fix the idea let us take $X(t)$ to be a white noise which is now viewed as the time derivative of a Brownian motion $B(t)$; the $\dot{B}(t)$ is a white noise. In this case we call $\{T_t, t \in R\}$, after S. Kakutani, the *flow of Brownian motion*.

This flow describes the evolution of the random pheneomena that are represented as functionals of white noise. More precisely, let $\varphi(\dot{B})$ be a white noise functional in Hilbert space (L^2) which contains white noise functionals with finite variance, then time evolution is expressed in terms of the untary operator U_t as

$$U_t\varphi(\dot{B}) = \varphi(T_t\dot{B}) \tag{2}$$

We can therefore speak of the spectral multiplicity of the unitary group $\{U_t\}$ on a subspace of (L^2), the choice of which depends on the problem to be discussed. For instance, if the entire space (L^2) is concerned, the multiplicity is infinite, while the subspace spanned by linear functionals of white noise has unit multiplicity. Computability necessarily requires finite multiplicity.

For the case of a stochastic process with higher dimensional parameter, say R^d parameter, the notion of multiplicity can also be introduced by taking the course of propagation to be the radial direction. The following assertion is almost obvious.

Let $\{X(a), a \in R^d\}$ be Lévy's Brownian motion in the sense that

1) it is Gaussian,

2) $X(a) - X(b)$ has mean 0 and variance $|a-b|$,

3) $X(O) = 0$, where O is the origin.

Proposition *Lévy's Brownian motion has infinite multiplicity.*

As a result, we claim that finitely many channels can not send the full information contained in Lévy's Brownian motion.

If we wish to describe finer way of dependency, then multiple Markov properties in a weak sense, defined like in the Gaussian case, can be defined in each cyclic subspace.

(2) The whiskers as time change operators

White noise analysis has an aspect of infinite dimensional harmonic analysis since the white noise measure μ is invariant under the infinite dimensional rotation group. It has significant one-parameter subgroups which come from one-parameter families of diffeomorphism of the parameter space. They are called *whiskers*. The shift discussed above is a good example of a whisker. We can say that the isotropic dilation, which is another whisker, has infinitely many cyclic subspaces, that is they have similar spectral properties to the shift. The same is true for the special conformal transformation.

Such an observation is helpful when whiskers that come from the conformal tarnsformations acting on the parameter space are discussed in connection with the computability. We emphasize the significance of the role of whiskers when we form the innovation from the variation of random field, which leads us to define multiple Markov property with the help of innovation.

(3) Random time parameter

The so-called *subordination* is an interesting operation for a stochastic process $X(t)$. The time parameter t is replaced by an increasing additive process $Y(t)$. The case where $Y(t)$ is taken to be an increasing stable process with exponent α, with $0 < \alpha < 1$, has been discussed. (This topic was discussed in [3] from the viewpoint of the information theory.)

If we consider computability, we may discuss the case where $Y(t)$ is taken to be a Poisson process, so that $X(t)$ is observed at random times that appear successively with exponential holding time. If, in particular, $X(t)$ is taken to be a Brownian motion, we can see the probability distribution of the subordinated process $X(Y(t))$. Subordination is an interesting topic, but we do not go into details since there is no direct connection with Markov property, which is our main topic.

It is noted that the facts discussed in this section are quite different from those discussed in Section 1 from the viewpoint of computation.

3 Information theoretical study

We now come to a random field $X(C)$ indexed by a manifold C in R^d, the parameter space of white noise. Before we come to the study of information theoretical properties of $X(C)$, we wish to emphasize the significant advantages of taking a field $X(C)$ instead of a process $X(t), t \in R^d$, as a mathematical model of random complex systems. Namely,

A) $X(C)$ carries more information than $X(t)$ when the parameter moves.

We know that $X(t)$ for $d = 1$ has usually unit multiplicity but $X(C)$ has infinite multiplicity except degenerated cases. Further, we see from an intuitive observation that t is 0-dimensional and runs through a finite dimensional space, while the dimension of C is at least one and moves in an infinite dimensional space $\mathbf{C}$. It means that $X(C)$ expresses more complex random phenomena than $X(t)$.

B) For C we have complex ways of deformation. In some important examples (e.g. [3]) the innovation of the field can be obtained by deformations of C (including the mappings from C onto itself). As a result, various ways of dependence can be discussed, in particular multiple Markov properties.

Let us have a quick review of the representation theory of a Gaussian process $X(t)$ and give a remark so that we extend the results to random fields which are either Gaussian or non Gaussian expressed as a homogeneous chaos. A representation of $X(t)$ in terms of a white noise $\dot{B}(t)$ such that

$$X(t) = \int_0^t F(t,u)\dot{B}(u)du, \tag{3.1}$$

is called *canonical* if the conditional expectation $E[X(t)|X(u), u \leq s]$ with $s \leq t$ is given by

$$\int_0^s F(t,u)\dot{B}(u)du. \tag{3.2}$$

In such a case, we can obtain the innovation $\dot{B}(t)$ by a causal and linear operator; formally speaking by the inverse of the integral operator F. The idea behind such an observation is that we define a stochastic process as a random function that gains a new information (actually, expressed by the innovation) at each instant. The probabilistic structure of a process is determined by the variation of the field involving innovation and, of course, past values.

If $X(t)$ is a nonlinear function of white noise, then it is no more Gaussian and we can not say that innovation is obtained in a similar manner, as is easily understood (see 1 in the next section).

However, if a random field $X(C)$ formed by a homogeneous chaos in the form

$$X(C) = \int_{(C)^n} F(C,u_1,u_2,...,u_n) : x(u_1)x(u_2)...x(u_n) : du^n, \tag{3.3}$$

where (C) is the domain enclosed by C.

Then the innovation can be obtained from $\delta X(C)$ by using nonlinear operations (see [2]). There we have assumed that the kernel F is the canonical kernel.

In order to carry on the calculus, we have to assume that C runs through a certain class of smooth ovaloids.

With this remark we are now able to define a generalization of multiple Markov properties.

4 Multiple Markov properties for non-Gaussian case

For a Gaussian case, the N-ple Markov property can be characterized by the canonical kernel; namely it should be a Goursat kernel of order N.

We now think of the non-Gaussian case which is restricted to be a homogeneous chaos.

1. Stochastic process $X(t)$

Suppose that $X(t)$ is not a Gaussian process, say homogeneous chaos (of white noise) of order greater than 1. Then it is impossible to obtain the innovation $\dot{B}(t)$ from the variation $\delta X(t)$. For example, let

$$X(t) = \int_0^t \int_0^t F(t; u_1, u_2) : x(u_1)x(u_2) : du_1 du_2 \tag{4.1}$$

be given. Then its variation is

$$\begin{aligned}\delta X(t) &= dX(t) \\ &= dt \int_0^t \int_0^t \frac{\partial}{\partial t} F(t, u_1, u_2) : x(u_1)x(u_2) : du_1 du_2 + 2dt x(t) \int_0^t F(t, t, u_2)x(u_2)du_2.\end{aligned}$$

Here we see that the second term has a different order, but $\int_0^t F(t, t, u_2)x(u_2)du_2$ is not a conditional expectation of $X(t + dt)$. since it is orthogonal to $X(s), s \leq t$; hence not a function of the $X(s)$.

2. Random field $X(C)$

We now come to the case of random field which is not Gaussian but homogeneous chaos. To fix the idea let us consider the case of quadratic chaos (take $n = 2$ for $X(C)$ in the last section),

$$X(C) = \int_{(C)^2} F(C; u_1, u_2) : x(u_1)x(u_2) : du_1 du_2. \tag{4.2}$$

Assume that F is the canonical kernel. Note that canonical property can be defined in the same manner to the case of Gaussian random field. Of course the uniqueness of the canonical kernel is guaranteed.

¿From the result in the paper [2] we can obtain the innovation $\{x(s), s \in C\}$ from $\delta X(C)$ and also the conditional expectation $E[X(C)|\mathbf{B}(C')]$, where $\mathbf{B}(C')$ is the sigma-field of events determined by $X(C''), C'' < C'$. Here $C'' < C'$ means that C'' is inside of C'. There is an important fact; namely, the conditional expectation is a nonlinear function of the $X(C'')$ with $C'' < C'$ (note that no more Gaussian case). It is therefore a nonlinear function, in reality a quadratic function, of the $x(u), u \in (C')$.

Since the conditional expectation is the projection to the space spanned by the non linear functions of $X(C''),\ C'' < C'$. Hence, it is the projection to nonlinear function of $x(u), u \in (C')$. Thus we have

$$E[X(C)|\mathbf{B}_{C'}(X)] = \textstyle\int_{(C')^2} F(C, u_1, u_2) : x(u_1)x(u_2) : du_1 du_2. \tag{4.3}$$

The results obtained so far hold for a homogeneous chaos of any order.

Thus, multiple Markov properties for $X(C)$ of homogeneous chaos can be defined in a similar manner to the case of Gaussian fields since conditional expectation is formed by the innovation. To make sure, we give

Definition Let $X(C)$ be given by

$$X(C) = \int_{(C)^n} F(C, u_1, u_2, ..., u_n) : x(u_1)x(u_2)\cdots x(u_n) : du^n \qquad (4.4)$$

with a canonical kernel F. For any choice of C_i's such that $C_0 \leq C_1 < ... < C_N < C_{N+1}$,

1. $E[X(C_i)|\mathbf{B}_{C_0}(X)], i = 1, 2, ..., N$, are linearly independent and
2. $E[X(C_i)|\mathbf{B}_{C_0}(X)], i = 1, 2, ...N + 1$ are linearly dependent

then, $X(C)$ is said to be *N-ple Markov*.

Theorem *If a random field $X(C)$ of homogeneous chaos is N-ple Markov, then its canonical kernel is a Goursat kernel of order N.*

Proof. For proof we only note that the conditional expectation is a nonlinear function of the known values, unlike Gaussian case. For the rest of the proof we can follow the method given in [4].

Corolary *The predictor of an N-ple Markov random field of homogeneous chaos is computable. More precisely, the best predictor is a linear combination of the random variable obtained from the values of the past.*

To close this paper, we note that the multiple Markov properties indicate not only way of dependency, but also suggest computability of the best predictor.

References

[1] T. Hida, Canonical representation of Gaussian processes and their applications. Mem. College of Sci. Univ. Kyoto, A 33 (1960),109-155.

[2] T. Hida and Si Si, Innovation for random fields. Infinite Dimensional Analysis, Quantum Probability and Related Topics. 1 (1998), 499-509.

[3] Si Si, Random irreversible phenomena. Entropy in subordination. Proceedings of Les Treilles Conference (1999), to appear.

[4] Si Si, Gaussian processes and Gaussian random fields. Quantum Information II (2000) World Scientific, 195-204.

[5] L. Accardi and Si Si, Innovation approach to multiple Markov properties of homogeneous chaos. Preprint.

[6] Si Si, Representations and transformations of Gaussian random fields, to appear in the proceeding of the International Conference on Quantum Information III.