International Journal of Modern Manufacturing Technologies
ISSN 2067–3604, Vol. V, No. 2 / 2013

APPLICATION OF SIMULATED ANNEALING ALGORITHM TO ESTIMATION OF SPEECH FEATURES

Alexey Ermilov[1]

[1]"Higher School of Economics" National Research University, Department of Software Engineering, Russia, 101000, Moscow, Myasnickaya Street, 10

Corresponding author: Alexey Ermilov, alvalerm@mail.ru

Abstract: In this work we present a method which might be useful to construct a verbal interface in mechatronics. Inevitable step in construction of a verbal interface is somehow estimate parameters of input voice signal. Maximum likelihood estimation is a very popular approach to get estimates of unknown variables. However, in practice obtaining estiamtes is a very hard task: we need to find a global optimum value of the function, which form depends on distribution we have chosen and the data. Thus, it might be concluded that the aim function in the maximization task is neither concave nor differentiable. In order to solve the maximization task we apply a simulated annealing method (Kirkpatrick et al., 1983) which does not use gradient in its work. In our work we present both Monte – Carlo study and application of the method to real world data.

Key words: Gram – Charlier density, simulated annealing, speech features.

1. INTRODUCTION

Most of currently available commercial speech recognition systems are based on the following principle. Feature vectors, which describe input voice signal, are extracted from input utterance and given to a recognizer, which is based on Hidden Markovs Models (Rabiner, 1989). Mel - Frequency Cepstral Coefficients (MFCC, Rabiner, 1989) are used as feature vectors.

Unfortunately, this approach has the following flaw. Depending on the length of a vocal tract and its shape (it is worth mentioning that length of a vocal tract is dependent on sex and other physiologic parameters of a speaker, such as height, and can vary from 13 cm to 18 cm) frequencies of central formants are shifted. The value of the shift can be as large as 25%. This huge difference can lead to a wrong recognition of a new utterance by a previously well-trained model when the utterance was said by a new speaker, thus the system becomes speaker-dependent.

One of the possible approaches to solve this problem is to use a technique called Voice Tract Length Normalization (VTLN, Cohen, 1995), which essentially transforms input utterance in such a way that central formants are on the same frequency. However, prior to transformation one needs to estimate parameters of the transformation which are also speaker dependent, which might be impossible when a training data set is limited.

The other approach, which is considered in this paper, is to use features that do not vary from speaker to speaker. One of the possible candidates for such features is those obtained from Auditory Image Model (AIM, Monaghan, 2005). In our work we present a method to construct feature vectors using AIM, show the results of a Mote – Carlo study as well as experiments with real data.

2. AUDITORY IMAGE MODEL

In this section we give a brief introduction to Auditory Image Model. The description follows Munich and Lin, 2005.

Auditory Image Models were developed in the Lab of Roy Peterson in Cambridge University with the aim of modeling human psychoacoustics. AIM is a functional model of human auditory system, which takes into account biological information. Model constists of three consecutive modules.

1. A filter bank, which consists of gammatone filters spaced according to ERB scale. Output signal if the filterbank rougly corresponds to the movement of basal membrane of the cochlea.
2. A two dimensional adaptive thresholding mechanism. At this stage a signal from filterbank is half-wave rectified and passed through a compressive logarithmic non-linearity. After thatadaptive thresholding mechanism is applied in two dimensions: time and frequency. Temporal thresholding includes a short term memory of pact activity. The spectral thresholding is basd on interactions between neighboring frequency channels: strong activity in a channel will partially suppress activity in less strongly stimulated neighbors. Thy output signal mimics the neural activity pattern

(NAP) of the auditory nerve, which connects the cochlea to brain stem nuclei.

3. Strobing integrator. Strobing integrator is applied to NAP in order to synchronize periods between maximums of NAP.

For feature construction we used the following scheme. Activity in each channel was smoothed with lowpass filter with cut-off frequency of 100 Hz, after that NAP was framed by 10 ms window. We obtain NAP profile by summation of activity in each channel. Next obtained profile was normalized and in such a way that it can be treated as probability density function. In order to describe this probability density function we suggest using Gram – Charlier extension of normal density.

3. GRAM – CHARLIER EXPANSION

Suppose that the true density of a random variable z is unknown, and then we could look at it as a combination of two functions:

$$g(z) = p_n(z)\phi(z), \qquad (1)$$

where $\phi(z)$ is a probability density function of normal random variable, and $p_n(z)$ is chosen in such a way that $g(z)$ has the same moments as the true density of z. This approximation is called Gram – Charlier expansion.

The idea behind this expansion is to express the characteristic function of one probability density with that of another one, with known properties (say, normal).

Hermitian polynoms form an orthogonal basis with respect to schalar product based on expectation on standard normal density. This feature allows using Hermitian polynoms in function $p_n(z)$:

$$p_n(z) = 1 + \sum_{i=1}^{n} c_i H_i(z), \qquad (2)$$

Parameters c_i are closely related to cumulants of the distribution. So, it seems convenient to use such an approximation since it allows us to control overe higher order moments of the distribution, which might be important for speech recognition: some authors claim that modeling higher order moments of speech signal could lead to higher accuracy during recognition (see, for example, Salavedra et.al, 1994, Nemer et al., 2002).

Unfortunatly, obtained function is not a density: it can be negative for some parameter values. In order to overcome this difficulty it was proposed by Niguez, 2004 to use a positive density:

$$g(z) = \phi(z)(1 + \sum_{i=1}^{n} c_i H_i(z))^2 / k, \qquad (3)$$

where $k = 1 + \sum_{i=1}^{n} c_i^2 i!$. Such density is convenient not only from theoretical point of view, but also from practical: during parameter estimation via maximum likelihood logarithmic likelihood function is divisible and contains logarithms of positive values, which leads to better numerical optimization.

$$l(z_i) = \ln\phi(z) + \ln(1 + \sum_{i=1}^{n} c_i H_i(z))^2 - \\ - \ln\sum_{i=1}^{n} c_i^2 i! \qquad (4)$$

In order to obtain estimates of unknown parameters with should numerically solve the following problem:

$$l(z,\theta) = \sum_{i=0}^{N} l(z_i) \to \max_{\theta} \\ s.t.\, f(\theta) \le 0, \qquad (5)$$

where $l(z_i)$ is given in (4), θ is a vector of parameters to be estimated and $f(\theta)$ is a constraints function, which might be included in order to ensure that parameters' values satisfy some apriory given constraints. For example, some cumulants should be positive.

4. MONTE – CARLO EXPERIMENTS

At first we suggest to perform a Monte- Carlo study in order to investigate the properties of simulated annealing algorithm applied to solve (5).

In this work we considered a Gram-Charlier expansion of normal density with cumulants $\kappa_1 = 2, \kappa_2 = 3, \kappa_3 = 6, \kappa_4 = 10$. Proposed density is presented on Figure 1. We can see that the density is far from normal: it is not symmetric and its tails are too heavy.

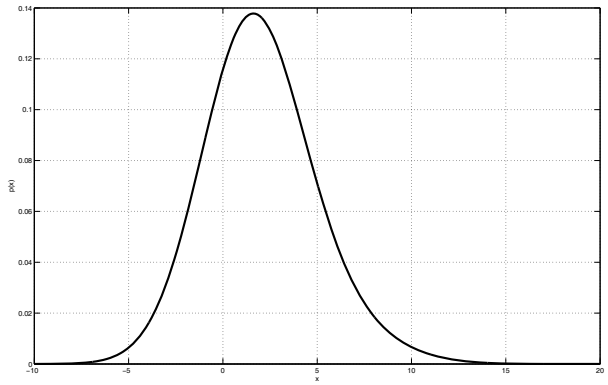

Fig. 1. Probability denstity function

We suggest using Monte-Carlo Markov's Chains (MCMC) to genereates samples from the density (3). The idea behind the method is the following. We simulate a markov chain which limiting distribution

is that of interest. We use slice samples of Neal, 2003 in order to obtain samples from the distribution with density (3). During simulation we discarded first 5000 observations (so called *burn in* period) to ensure that the markov chain reached its limiting distribution. Since algorithm produces samples which might be correlated, we also discarded 4 out of 5 produced samples to control for

Histogram of obtained samples is on Figure 2.

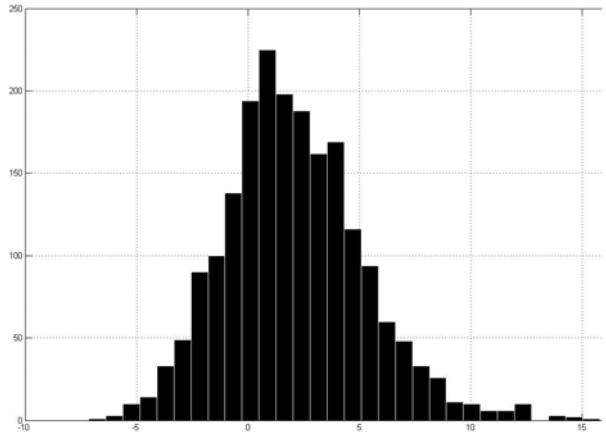

Fig. 2. Histogram from the density under consideration

indepency of produced observations. In such a way we produced 2000 samples.

We see that generated samples closely follows the density of interest.

In order to maximum likelihood estimates of parameters of interest we should numerically solve the problem (5) with respect to parameters $\{c_i\}_{i=1}^4$, which form the vector θ. So, in our case we need to find a maximum value of a function of 4 parameters. It worth mentioning that parameters c_3 and c_4 are of particular interest since they related to cumulants κ_3, κ_4, which in turn model skewness and kurtosis of distribution.

In order to demonstrate how bad behaved might be the aim function in (5) lets look at function $l(z,\theta)$ as function of c_3 and c_4 when c_1 and c_2 are held fixed. Figure 3 demonstrated negative value of $l(z,\theta)$ when $c_1 = 0, c_2 = 1$.

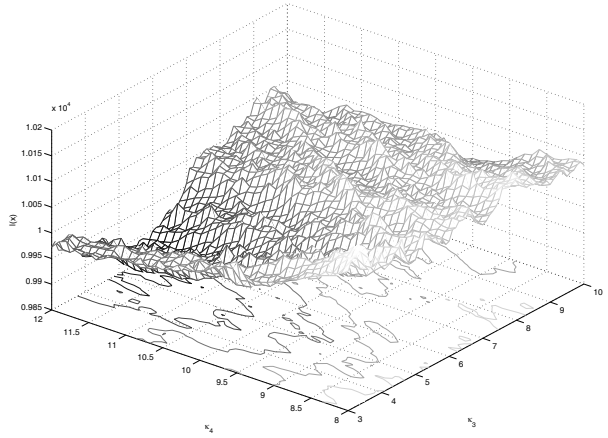

Fig. 3. Negative log likelihood with incorrect parameter values

We see that the function is neither smooth nor concave. Moreover, it has a lot of local extremums, in which gradient methods can stuck.

In fact even if correctly specified nuisance parameters (in our case it is c_1 and c_2) the aim function might still not demonstrate good properties. From Figure 4 we can see the surface of $l(z,\theta)$ with correctly specified parameters c_1 and c_2. We see that the function has local extremums too.

What is important is that in real case when observations are corrupted with noise the aim function would have even more complex structure, which might lead to even more complex surface and make estimation much more challenging.

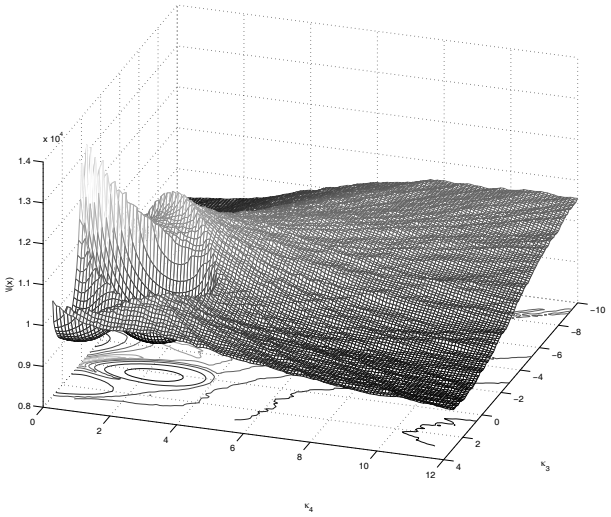

Fig. 4. Negative log likelihood with correct parameter values

As the next step let's look at parameter estimates, obtained from various numerical algorithms. We considered three methods here.

First is a method of gradient descent, which is common choice in many areas. The method relies on numerical estimation of gradients, so its performance might be enhanced by providing the algorithm with pre − specified gradient function. The advantage of this algorithm is that in case of well − behaved aim function (smooth, and concave (convex)) it would achive the optimum value after a small number of iterations.

Second method is simplex algorithm of Nelder and Mead (Lagarias et al., 1998). This algorithm has the advantage of being not gradint dependent, so local deviations from smoothness or even discontinouses of objective function can be handled by the method. On the other hand the algorithm needs a lot of function evaluation, which might be a point of great concern if objective function's evaluations are costly.

Third method is simulated annealing. This method also does not rely on gradient evaluations. The algorithm exploits the analogy between the way in which metals cools and freexes in minimum energy crystalline structure and the search for a minimum in

a more general system. Basicaly we need to define two distributions: generating distribution and acceptance distribution. Both of them depend upon temperature. The former generates states which would be explored and the latter decides probabilistically wheter to stay in a current state or to move out of it. Thuse, if the algorithm is in a local extremum then there is a chance that algorithm would leave this point by accepting a point with worse objective function value. However, like Nelder – Mead algorithm simulated annealing requires many function evaluations. Moreover, unlike two previous methods simulated annealing is a stochastic algorithm, so it is important to look at the distribution of obtained estimates. There is also a chance that given realization of search path lead to a function value that is far from true value.

Now look at the results of numerical estimation. The results are summarized in Table 1. We bounded number of function evaluations for simulated annealing algorithm by 5000. As starting point for all methods we took empirical cumulants of sample distribution. We see that all methods give good results for first and second cumulants: true values in all cases are in theirs respective confidence intervals and standard errors (given in parenthesis) are quite small.

Situation is worse for third cumulant. The true value is still in confidence interval for all methods, but standard errors increased drastically. It is worth mentioning that in this particular case we used inverse of hessian of negative log likelihood function as asymptotic covariance of maximum likelihood estimates. Moreover, the hessian was calculated using numerical differencing.

Estimates became even worse for the fourths cumulant. The standard errors are very big and only simulated annealing gave a point value which is close to true value of the parameter. However, estimated values are not significant at 95% level.

The results of the estimation suggest that such likelihood function is very hard to optimize via numerical methods. Big standard errors might be a consequence of instability of hessian calculations, which might also be exaggerated by the fact that hessian is bad conditioned and inverse operation gives inprecise results. All these suggest using analytical form of hessian for standard errors calculation.

Table 1. Estimation results

Parameter value	Gradient descent	Nelder - Mead	Simulated annealing
c_1	2.04 (0.07)	2.02 (0.07)	1.97 (0.07)
c_2	3.01 (0.05)	3.01 (0.05)	2.94 (0.05)
c_3	5.4 (0.84)	5.38 (0.85)	5.35 (0.84)
c_3	3.82 (5.1)	6.03 (5.12)	9.65 (5.84)

In order to further investigate properties of estimates obtained from simulated annealing algorithm we constructed empirical distribution of estimates. To do so we repeated estimation process 1000 times. Emprical cumulative distribution functions (cdf) are presented on Figure 5.

The empirical cdf for parameters are in line with the results obtained from the first estimation. Mean values are close to the true values for parameters c_1 and c_2 with small standard deviations from the mean. Both of them have much better properties then estimates of two other parameters. We get relatively big standard error for parameter c_3, at the same time mean value of the estimate is close to its true value. What is important is that even after big number of recalculations we cannot get reliable estimate of the last parameter.

However, it is worth mentioning that estimation of higher order moments of the distribution is enevitably connected with difficulties. In our case it is possible that we cannot simply distinguish two densities with different values of c_4 given that number of samples. The motivation for that is in the following. Since c_4 is responsible for kurtosis of the distribution, we need enough samples to fall into tails of the distribution.

5. EXPERIMENTS WITH REAL DATA

In Walters et al., 2008 the authors suggest using of mixture gaussians to model NAP profile. Then they used parameters of the mixture and corresponding parameters of the normal densities as input feature vector for automatic speech recognition system.

In this work we propose different reparametrization for NAP profile, which encorporates Gram-Charlier extension of the normal density as an element of the mixture.

Also, we propose two modifications of the simulated annealing algorithm which take into account possible parallelization of the calculations.

In the first modification we use a hybrid scheme. We start from different points in a search space and build the paths for each point independently. After each branch of the calculations converges, we compare obtained results and chose that gives the best value for the aim function given constraints.

In second modification we suggest using pre – calculation of the initial values. Thus, we could get better initial point, and the algorithms could converge faster.

Figure 6 demonstrates typical approximation of NAP profile via mixture of 3 Gram – Charlier expansions. We see that the fit is almost perfect: all peaks are correctly identified. In addition, we see that tail

behavior of our mixture is close to that of the profile. It might indicate that notwithstanding relatively bad performance in Monte – Carlo simulations algorithm provide us with as good estimates as it was necessary to properly fit the profile.

It is also interesting compare two proposed algorithms in terms of accuracy and efficiency.

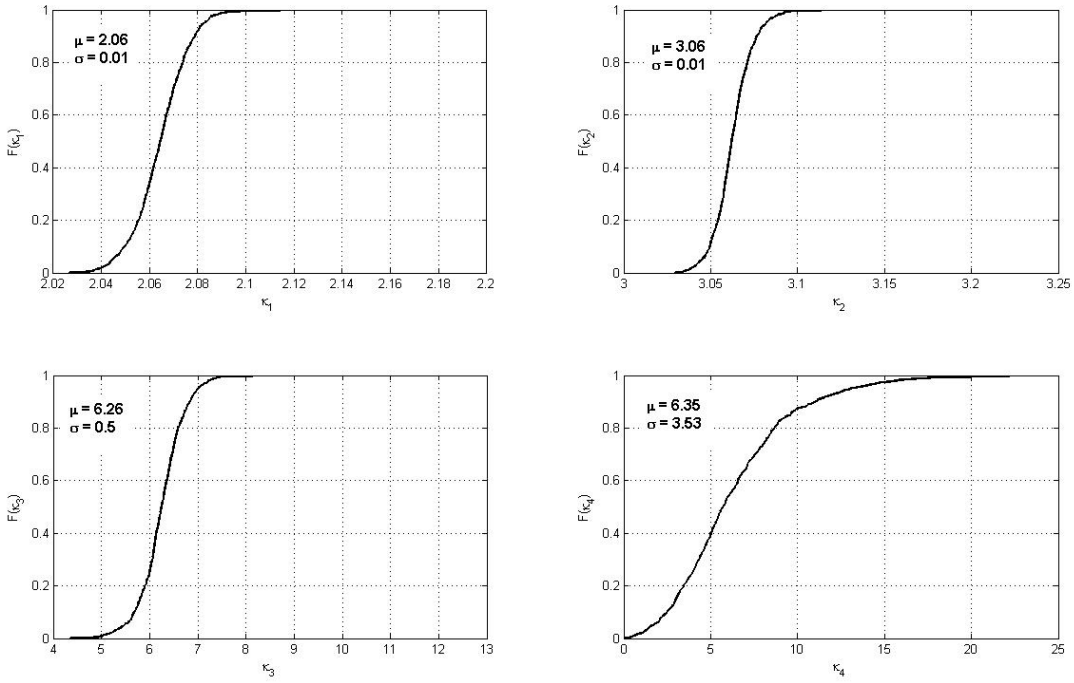

Fig. 5.Empirical distribution functions for parameter estimates

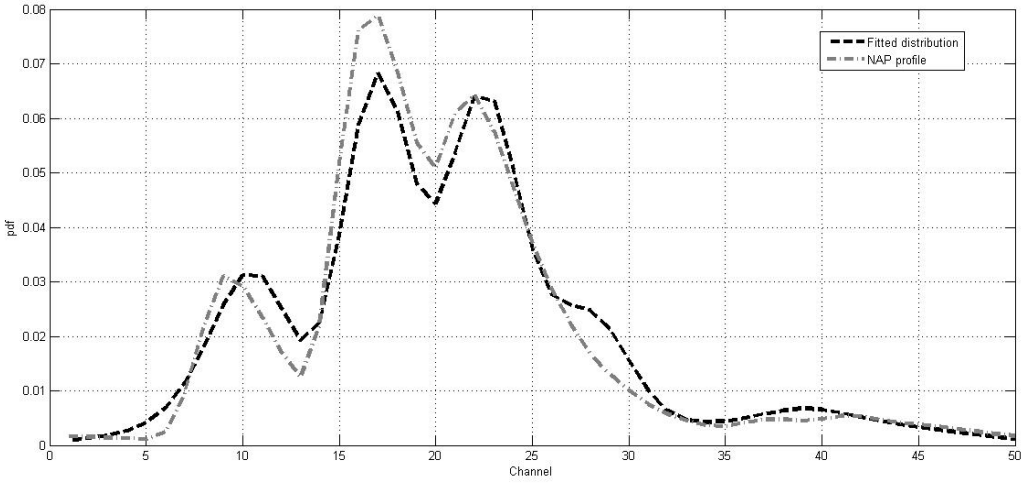

Fig. 6.Fitting of two distributions

Table 2 summarizes the time cost of parallel execution on different number of processors of two proposed algorithms.

Table 2. Time of execution for two algorithms

Number of processors	First algorithm time of execution, sec	Second algorithm time of execution, sec
1	9756	15
3	4465	13
6	2463	11

As it was expected the first algorithm is more time consuming. However, time dependence is almost linear. So, being implemented on graphical cards,

even the first algorithm could work in real time. It is not necessary, as a matter of fact, that this relationship would preserve if would increase number of processors, since in this case we would be obliged to take into account interdependence of search paths in order to prohibit visiting the same point several times. But this interdependence might be neglected in the second case. Thus, the question is what algorithm gives us a better fit.

We also compare goodness of fit of two proposed algorithms. As measures of we use Kullback – Leibler divergence and a value of log likelihood function. The results are presented in Table 3.

As we can see first algorithm has a better fit: Kullback – Leibler divergence is smaller and the aim function has a bigger value. However, the advantage of the first algorithm is not dramatic.

Table 3. Goodnes of fit for two alorithms

Measure of fit	First algorithm results	Second algorithm results
Kullback – Leibler divergence	0.117	0.140
Log - likelihood	-32935	-34289

From the comparison of two algorithms and the consideration about possible performance degradation mentioned above we could suggest that the second algorithm, yet slightly less accurate, might be of preferred use in practical applications.

6. CONCLUSIONS

In this paper we considered a new parametric model which approximates NAP profile, obtained from AIM. Our model is based on Gram-Charlier expansion of the normal density. Parameters of the model might be used in automatic speech recognition systems as feature vectors.

However, estimation of the parameters values is a challenging task. To demonstrate all the difficulties that arise during estimation process, we made a Monte-Carlo study, which demonstrated that likelihood function is bad behaved: it does not have a constant curvature and has a lot of local extremums.

To cope with this difficulty we suggest using stochastic optimization. In this work we used simulated annealing algorithm. We built empirical distribution function of parameters estimates. It could be concluded from Monte – Carlo study that proposed algorithm gives good estimates of all but one parameter. However, from comparison with out methods, both gradient and gradient free, it is clear that no method could handle this problem well.

In additional to Monte-Carlo study we test proposed methodoly on real data. To enhance algorithm performance we suggest two hybridization schemes of the algorithm and compared them in terms of efficiency and goodness of fit. This experiment shows that we obtained very good fit.

As a direction for further research we might suggest derivation of a goodness of fit test of these two methods, comparison of this approach with standard expectation-maximization algorithm. From practical point of view it is important to implement proposed algorithms on GPU, compare automatic speech recognition systems based on classical MFCC features and that on proposed features.

7. REFERENCES

1. Kamm, T., Andreou, G., Cohen, J., (1995), *Vocal Tract Normalization in Speech Recognition: Compensating for Systematic Speaker Variability,* Proc. of the 15th Annual Speech Research Symposium, pp. 161-167.

2. Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., (1983), *Optimization by simulated annealing,* Science, Vol. 220(4598), pp. 671-680.

3. Monaghan J., Feldbauer, C., Walters, T., Patterson, R., (2008), *Low-Dimensional, Auditory Feature Vectors that Improve VTL Normalization in Automatic Speech Recognition,* Journal of the Acoustical Society of America, Vol. 123, pp. 3066.

4. Del Brio, E., Ñíguez, T., Perote, J., (2011), *Multivariate semi-nonparametric distributions with dynamic conditional correlations*, International Journal of Forecasting, Vol. 27(2), pp. 347-364.

5. Radford M Neal, (2003), *Slice sampling,* Annals of Statistics, Vol. 31(3), pp.705–767.

6. Lagarias, J.C., Reeds, J. A., Wright, M. H., Wright, P. E., (1998), *Convergence Properties of the Nelder-Mead Simplex Method in Low Dimension,* SIAM Journal of Optimization, Vol. 9, pp. 112-147.

7. Rabiner, L., (1989*), A tutorial on hidden markov models and selected applica-tions in speech recognition,* Institute of Electrical and Electronics Engineers, Transactions on Information Theory, Vol. 77(2), pp. 257–284.

Received: August 20, 2013 / Accepted: December 5, 2013 / Paper available online: December 10, 2013