The Evolutional Portfolio Optimization System (EPOS)

N. Loukeris¹,², Y. Boutalis³, I. Eleftheriadis¹

¹Department of Business Administration, University of Macedonia, Egnatias 156, Thessaloniki, Greece
²Dept. Electrical and Computer Engineers, Democritus University of Thrace, Xanthi, Greece
nloukeris@uom.edu.gr

Abstract - We introduce a new methodology that incorporates advanced higher moments evaluation in a new approach of the Portfolio Selection problem, supported by effective Computational Intelligence models. The Evolutional Portfolio Optimization System (EPOS) extracts hidden patterns out of the numerous accounting data and financial statements filtering misleading effects such as noise or fraud, offering an optimal portfolio selection method.

Keywords: Isoelastic Utility function, Higher moments, Recurrent Neural Networks, Time-Lag Recurrent Neural Networks, Jordan-Elman Neural Networks, Hybrid Networks, Portfolio Optimization

1 Introduction

The optimal selection problem is a two phase process. Firstly, the portfolios evaluation offers the feasible set, largely inefficient and mostly rejected by risk-averse investors. The remaining efficient portfolios depend on investor’s behavior, are based on a utility function, ranked and thus the model is receiving the optimal portfolios. Secondly, the risk expressed in variance, volatility, hyperkurtosis, is minimized.

The goal of this research is to examine the first phase of the optimization problem, whilst offering a general solution of the second step and an integrated system capable of selecting and optimizing portfolios employing advanced methods of Artificial Intelligence and Finance. The single period model is examined, as we evaluate various artificial intelligence models of the Recurrent Networks family: the Recurrents, the Time-Lag Recurrents, the Jordan Elmans, to the MLP, in neural or hybrid neuro-genetic forms of different topologies: each: 40 on the Recurrents, 44 on the TLRNs, 66 on the Jordan Elmans, to create the efficient portfolio set. The scope is quintuple:

I) to investigate in depth investors’ genitive behavior in higher moments, seeking more information on earnings and exposure to risk preferences,

II) to introduce an improvement of the isoelastic utility as a more optimal function that supports higher moments,

III) to further develop the Markowitz’s portfolio theory, in fundamentals evaluation, prices, or other available information, to clear the unnecessary noise, and determine healthy firms excluding manipulation, fraud, etc.

IV) to examine the efficiency of various networks of the Recurrent family in neural or neuro-genetic hybrid nets on past results concluding on the optimal classification model to a dynamic, optimized investment portfolio.

V) to introduce the integrated model EPOS as a modern solution to portfolio selection and optimization problems inspired by cutting edge technologies.

2 Higher moments

The returns distributions are not n.i.i.d. in reality, and EMH fails significantly in the markets, since various non-financial parameters affect the stock prices. As observed, Subrahmanyam (2007), investors are more sensitive to their potential losses, we will try to model the overall preferences, even those that incorporate the sub-conscious trends that guide them. The investors distribute their utility balancing perceptions and fears, on the one hand, and earnings on the other. Their logic expects a rational amount of return, but the fear of loss subconsciously magnified, produces remarkable decisions. The majority of investors are risk averse or risk neutral, hence the fear parameter is easy to manipulate behaviors. In bullish periods the fear of losing excess profits, whilst in bearish the fear of maximising losses, can influence non rational herding behaviors.

A more analytical tool that will focus in depth the details that define investing behaviors is introduced. The further higher moments detect the hidden aspects of investors’ decision making. Loukeris et al. (2014a, b) noticed that on the implied utility function of the HARA family (Hyperbolic Absolute Risk Aversion) the 5th of hyperskewness and the 6th of hyperkurtosis moments should be used in the form of

\[
U_j(R_j) = aE_j + bVar(R_j) + cSkew_j + dKurt_j + eHypSkew_j(R_j) - fHypKurt_j(R_j) + gUltraSkew_j(R_j) - hUltraKurt_j(R_j)
\]

or

\[
U_j(R_j) = aE_j + bVar(R_j) + cSkew_j + dKurt_j + eHypSkew_j(R_j) - fHypKurt_j(R_j) + gUltraSkew_j(R_j) - hUltraKurt_j(R_j)
\]

UltraKurt, \( R_{a,t} = Kurt^n_{a,t} R_{a,t} = Var^n_{a,t} R_{a,t} \)

Thus (2) as a series of higher order moments can be extended to the level of analysis that is desired. A general form of the utility function, Loukeris et al (2014a, b), is:

\[
U_j(R_j) = \sum_{k=1}^{n} (-1)^{k-1} \hat{\lambda}_k \sum_{i=1}^{n} \left( x_i - \frac{\sum_{j=1}^{n} x_j}{n} \right)^k
\]

where \( \hat{\lambda}_k \) is the depth of accuracy on investors utility preferences towards risk, depending on the behavior, \( a_{ik} \), a constant on investors profile: \( a_{ik} = 1 \) for rational risk averse
individuals, \( a_{xy} \neq 1 \) for the non-rational, \( x_i \) the return \( i \) in time \( t \). The Isoelastic Utility, a unique HARA function of Constant Relative Risk Aversion, is on the risk averse investors:

\[
U = \left[ \frac{W^{1-\lambda}}{1-\lambda} - \frac{1}{\lambda} \log(x) \right] \quad \lambda \in (0, 1) \cup (1, +\infty)
\]  
(5)

where, \( W \) the wealth, \( \lambda \) a measure of risk aversion. Loukeris et al. (2014a, b) indicated the Markowitz model can have a broader alternative relaxing its essential assumption on the normally distributed prices. The initial convex problem of quadratic utility maximization, Markowitz (1952),

\[
\min_{x} f(x) = \text{Var}(r_p)
\]
(6)

is inadequate in real markets. Maringer and Parpas (2009) incorporated higher order moments:

\[
\min_{x} f(x) = \lambda \text{Var}(r_p) - (1 - \lambda) E(r_p)
\]
(7)

\[
r_p = \sum_{i} x_i r_i
\]
(8)

\[x_i \geq 0, \quad \sum_{i} x_i = 1
\]
(9)

where \( r_p \), the portfolio return, \( x_i \) the weight of asset \( i \), \( r_i \) the return of \( i \)th asset, \( \mu \) the mean and \( \sigma^2 \) the variance.

### 3 Problem Definition

Loukeris et al. (2014a, b) indicated the necessity of further higher moments into the model, to optimally describe investors’ preferences. The problem, is:

\[
\min_{x} f(x) = \frac{1}{2} \text{Var}(x) \left[ b \text{Var}(r_j) + d \text{Kurt}(r_j) + \text{HypKurt}(r_j) + \text{UltraKurt}(r_j) \right]
\]
(10)

\[
\text{Var}(x) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2
\]
(11)

\[
E[U(w, \lambda)] = \frac{1}{n} \sum_{i=1}^{n} \left[ 1 + \exp\left(r_i x_i \right) \right] / N
\]
(12)

\[
E(U_p(w, \lambda)) = \frac{\sum \left[ 1 + \exp(r_i x_i) \right]^{1-\lambda} \lambda}{1-\lambda} / N
\]
(13)

\[
Var(x) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2
\]
(14)

\[
\text{Kur}(r_i) = \text{Kur}(r_j) < \text{Kur}(r_i)
\]
(15)

\[
\text{HypKur}(r_i) = \text{HypKur}(r_j) < \text{HypKur}(r_i)
\]
(16)

\[
\text{UltraKur}(r_i) = \text{UltraKur}(r_j) < \text{UltraKur}(r_i)
\]
(17)

The stocks that fail to fulfill all the previous superiority conditions are non-optimal stocks and are exempted from the optimal portfolio set, as a part of the efficient frontier. Thus given Loukeris et al. (2014a, b):

\[
\sum_{i=1}^{n} x_i = 1
\]
(18)

\[
E(U_p(w, \lambda)) = \frac{\sum \left[ 1 + \exp(r_i x_i) \right]^{1-\lambda} \lambda}{1-\lambda} / N
\]
(19)

\[
E(U_p(w, \lambda)) = \frac{\sum \left[ 1 + \exp(r_i x_i) \right]^{1-\lambda} \lambda}{1-\lambda} / N
\]
(20)

\[
\min_{x} f(x) = \frac{1}{2} \text{Var}(r_p) \left[ b + dz + fz^2 \right]
\]
(21)

\[
\min_{x} f(x) = \frac{1}{2} \text{Var}(r_p) \left[ b + dz + fz^2 \right]
\]
(22)

\[
\min_{x} f(x) = \frac{1}{2} \text{Var}(r_p) \left[ b + dz + fz^2 \right]
\]
(23)

### 4 The EPOS model

The integrated EPOS – Evolutional Portfolio Optimization System on the first step reads the fundamentals, the
accounting data, the market prices and the preferred optimization period t.

Then it proceeds by selecting the initial method to evaluate the companies whose stocks are candidate in the portfolio. On this step the individual investor’s risk profile is given and the λ is selected for the Isoelastic utility.

On the next step, the system examines if this is the last firm to be examined, and if the condition for the optimal portfolio as an efficient portfolio is satisfied. Else we proceed to the next of the initial evaluation that uses a Computational Intelligence model, to create two subsets: Subset A of the healthy companies, and Subset B of the distressed firms.

In the specific model we select the best network among the models, The ετ value is calculated (0, for the healthy and 1 for the distressed firms).

If ετ = 1 then the firm is distressed and it is removed, else if ετ = 0 the firm is healthy being candidate for the optimal efficient portfolio.

On the next step the U_i(R_i(t)) the utility function of (105) is calculated per firm.

Next, firms are ranked according to their utility score. Then, the Efficient Frontier is calculated.

Next, the firms with the higher utility score are selected into the efficient portfolio.

The sub-optimal firms as well as the non-optimal firms are revaluated with potential new data on the step 4 of Neural Nets evaluation, following all the steps.

Next, after the efficient portfolio is created, its Utility Function is calculated U_P(f).

Then, the optimal overall portfolio U*_P(f) whose utility is the maximum available, is detected, if possible, by all the available efficient portfolios utilities U_P(f) recorded in U*_P(f) > U_P(f).

The process stops when the time limit is reached and the PI has the optimal portfolio.

The key idea is to filter fraud and speculative noise that interfere on the price and disorient investors. Thus examining recent accounting entries and through their financial indexes we can define the real financial health of the firm. After the real healthy firms are selected then their returns are considered on the model and we proceed on the main core of the Markowitz initial approach, the detection of efficient frontier and the creation of the efficient portfolio. The model’s flow chart is in figure 1.

5 The Computational Intelligence

The emphasis on the classifier will be given on the Recurrent networks family. Specifically a thorough investigation of i) the Recurrent, ii) the Time Lag Recurrent, iii) the Jordan Elman, and the MLPs as a measure of comparison will take place. All the models will be examined in neural net and hybrid neuro-genetic form, on various topologies. In the following sub-sections a detailed presentation of the models takes place.
5.1 Recurrent neural networks

The Recurrent Neural Networks (RNN) form their connections in a directed cycle, creating a network of neurons with feedback connections. Recurrent Neural Networks operate differently from feedforward neural networks, during their computational behavior and training. The Recurrent Neural Networks may behave chaotically, thus dynamical systems theory is able to model and analyse them. The novel feedback network of Long Short Term Memory-LSTM Hochreiter and Schmidhuber (1997) overcomes the fundamental problems of traditional RNNs, and efficiently learns to solve many previously unlearnable tasks involving, Principe, deVries, Kuo and Oliveira, (1992), Principe, Euliano, and Lefebvre (2000): i) Recognition of temporally extended patterns in noisy input sequences, ii) Recognition of the temporal order of widely separated events in noisy input streams, iii) Extraction of information conveyed by the temporal distance between events, iv) Stable generation of precisely timed rhythms, smooth and non-smooth periodic trajectories, v) Robust storage of high-precision real numbers across extended time intervals, (figures 2,3,4,5).

5.1.1 Hybrid Recurrent Neuro-Genetic Networks

The Recurrent Networks had been optimised under Genetic Algorithms that determined the inputs, the neurons performance on every layer into different topologies. The On-Line learning was preferred to update the weights of the hybrid neuro-genetic Recurrent Networks, optimising the: a) quantity of Artificial Neurons, b) Step Size, and c) Momentum Rate, whilst the output layer had optimised the Step size and Momentum of the Genetics.

5.2 Time-Lag Recurrent Networks

Time-Lag Recurrent Networks-TLRN are Multi Layer Perceptrons-MLP that have been extended with short term memory formations with local recurrent synapses. TLRNs implement the Backpropagation Through Time-BPTT training algorithm, a superior algorithm to the standard Backpropagation.

5.2.1 Time Lag Recurrent Nets with Genetic Algorithms

Time Lag Recurrent Networks-TLRN were implemented in a hybrid form with Genetic Algorithms optimization. The significance of each one of the 16 financial indices, that are the inputs of the TLRN, is not predefined thus we used Genetic Algorithms to select the important inputs. The weights of hybrid neuro-genetic TLRN were chosen to be updated through On-Line learning, after the presentation of each exemplar. In contrast, Batch learning updates the weights after the presentation of the entire training set, and it was rejected. Accumulation of the gradient contributions for all data points in the training set before updating the weights is referred as batch learning, Orr (1999). In online learning, the weights are updated immediately after seeing each data point. The gradient for a single data point can be considered a noisy approximation to the overall gradient G, called stochastic (noisy) gradient descent. Online learning has a number of advantages, Orr (1999): it is often much faster, especially when the training set is redundant, it can be used when there is no fixed training set, it is better at tracking non-stationary environments, the noise in the gradient can help to escape from local minima.

A GA was used in a) Neurons, b) Step Size, and c) Momentum Rate solving the sub-problem of the optimal values for these three parameters. This form of optimization requires that the network be trained multiple times in order to find the settings that produce the lowest error. Output layer was chosen to implement Genetic Algorithms optimizing the value of the Step size and the Momentum.

5.3 Partially Recurrent Neural Networks

The Partially Recurrent Networks are MLP nets where a few recurrent connections are introduced. The input layer of Partially Recurrent Networks includes two types of neurons: the inputs, and the context neurons or neurons of state, that remember past actions and take output values from one of the layers delayed by one step. Internal states, that function as a short-term memory, of the partially recurrent neural nets, can predict time series, as they can represent information about the preceding inputs Stagge and Senho (1997). The Partial Recurrent Networks are i) the Jordan network, ii) the Elman network and iii) the Multi –Step Recurrent network.

5.4 The Jordan Network

Jordan (1986a, 1986) created the Jordan neural nets, where the context neurons receive a copy from the output neurons and from themselves, thus many context neurons are the outputs. The recurrent connections from the output layer to the context neurons have an associated parameter of constant value: m € (0, 1).

5.5 The Elman Network

Elman (1990) created the Elman nets, where the context neurons receive a copy of the networks’ hidden neurons and these connections do not need to associate any parameter. Thus the number of the context neurons is identical to the number of hidden neurons. The remaining activations are calculated as in a MLP, considering the sequence of external inputs and context neurons as the vector input to the network.
5.6 The Multi-Step Recurrent network

In the Multi-Step recurrent network, Galvan and Isasi (2001), the feedback connections are directed from the output neuron to input layer. The context neurons memorise previous outputs of the network. The number of input and context neurons is replaced in every sampling time by other neurons in cases of prediction.

5.7 The Jordan Elman Networks

The Jordan and Elman networks extend the MLP implementing neurons that remember past activity, the context units. These context units offer to the JE models the ability of extracting temporal information from the data. There are available 4 essential topologies that alter by the layers that feed the context units. The topology I provides the context units with the inputs, and builds a robust past substratum of the input by its memory traces. The topology II follows the Elman’s method and builds memory traces from the initial hidden layer. Topology III elaborates the past of the last hidden layer outputs as input to the context units. Finally topology IV works with the Jordan’s method taking the past of the output layer to create the memory traces. In this research we elaborate the initial topology I, that takes account of the inputs past. The context neuron memorizes the past of its inputs in the recency gradient, that uses oblivion in an exponential decay. Thus the memory of recent data is better than the memory of those in a more distant past. On 0 hidden layers the 1st and 2nd topologies are equivalent, as are the 3rd and the 4th. If there is 1 hidden layer, then the 2nd and the 3rd topologies are equivalent. With 2 or more hidden layers, all 4 topologies are unique. The context unit remembers the past of its inputs using a recency gradient, forgetting the past with an exponential decay, and controls the forgetting factor through the Time constant that here is selected to be the IntegratorAxon function of 0.8 s time –of a longer memory depth and a slower forgetting factor. There were standard 4 neurons per hidden layer using as the transfer function the TanhAxon, the learning rule was the Momentum function, on a value of 0.7 as a momentum and changing step size per hidden layer in a scale of 0.1

5.7.1 Genetic Algorithms in Jordan-Elman Hybrids

The significance on each one of the 16 financial inputs in all the JE networks is calculated through the Genetic Algorithms, on the Hybrid models only. These models are trained multiple times to detect the lowest error inputs. The GAs are used in four different hybrid models of different topologies: i) on the inputs layer only, ii) on the inputs and outputs layers only, iii) into all the layers, iv) into all the layers with cross validation. The Batch learning was preferred to update the weights of hybrid neuro-genetic JE, after the presentation of the entire training set. GAs optimized values in all the layers and the output in: a) the Step Size and b) the Momentum Rate. JE nets require multiple training to achieve the lowest error.

5.8 Multi-Layer Perceptrons

The Multi-Layer Perceptron–MLP is a widely used neural net
Lippman (1987), where input signals are computed in a number of layers Hornik, Stinchcombe, and White (1989), that contain artificial neurons, the Processing Elements-PEs of the network.

The classification process of the EPOS model is boosted by the appropriate model. The recurrent models and their variations offered very competitive results that can claim the role of an efficient classifier of the EPOS. Specifically the Hybrid TLRN of GA optimization in all layers and Cross Validation, with no hidden layers, was the best classifier of 99.57% and 96.6% successful classifications for the healthy and the distressed companies respectively, the highest fitness of the model to the data in r at 0.991, a very low error in MSE of 0.043, NMSE of 0.106, 5.61% error, a very low partiality risk of AIC in -2093.61 in quite similar results on the CV, and a processing time of 3h 22 minutes 35 seconds.

The second rank was given to the Hybrid TLRN of 1 layer and GA optimization in all layers, where the healthy and the distressed companies were classified correctly at 99.66% and 98.05% respectively, the correlation coefficient r was 0.986, the error was the lowest in MSE at 0.022, 0.056 on NMSE, 3.78% error, and the Akaike at -828.46, requiring 3h 34’ 11” to converge, exposed though to overfitting.

The third rank was taken by the Hybrid Recurrent net of 1 layer, GAs on the inputs only, in 98.99% and 96.6% correct classifications for the healthy and the distressed companies, very high fitness of the model to the data at 0.986, 0.038 MSE, 0.093 NMSE, 2.86% error, very high impartiality integrity in AIC -2148.5, and a fast time of 54 min. 40 s.

The EPOS model offers an integrated approach on the optimal portfolio selection problem. Its module-based architecture ensures a flexible environment that can support demanding environments. The consideration of the fundamentals as a significant parameter of the problem, whilst
Table 1. Optimal Recurrent models

<table>
<thead>
<tr>
<th>Model</th>
<th>Layers</th>
<th>Active Confusion Matrix</th>
<th>Performance</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0→0</td>
<td>0→1</td>
<td>1→0</td>
</tr>
<tr>
<td>Hybrid TLRN, GA all, CV</td>
<td>0</td>
<td>99.57</td>
<td>0.41</td>
<td>3.39</td>
</tr>
<tr>
<td>CV</td>
<td>1</td>
<td>99.24</td>
<td>0.75</td>
<td>0.91</td>
</tr>
<tr>
<td>Jordan Elman NN</td>
<td>1</td>
<td>99.91</td>
<td>0.08</td>
<td>3.20</td>
</tr>
<tr>
<td>Jordan Elman GA all, CV</td>
<td>2</td>
<td>99.66</td>
<td>0.33</td>
<td>5.50</td>
</tr>
<tr>
<td>CV</td>
<td>3</td>
<td>99.83</td>
<td>0.16</td>
<td>0.91</td>
</tr>
<tr>
<td>Hybrid TLRN, GA all, CV</td>
<td>4</td>
<td>99.24</td>
<td>0.75</td>
<td>1.93</td>
</tr>
<tr>
<td>CV</td>
<td>5</td>
<td>99.15</td>
<td>0.83</td>
<td>1.83</td>
</tr>
<tr>
<td>TLRN N. N.</td>
<td>2</td>
<td>98.90</td>
<td>1.08</td>
<td>3.39</td>
</tr>
<tr>
<td>Jordan Elman GA all</td>
<td>1</td>
<td>99.83</td>
<td>0.16</td>
<td>5.50</td>
</tr>
<tr>
<td>Jordan Elman NN, CV</td>
<td>2</td>
<td>100</td>
<td>0.00</td>
<td>6.42</td>
</tr>
<tr>
<td>CV</td>
<td>3</td>
<td>100</td>
<td>0.00</td>
<td>6.42</td>
</tr>
<tr>
<td>Jordan Elman GA inputs</td>
<td>1</td>
<td>100</td>
<td>0.00</td>
<td>8.25</td>
</tr>
<tr>
<td>Jordan Elman NN</td>
<td>2</td>
<td>99.91</td>
<td>0.08</td>
<td>4.12</td>
</tr>
<tr>
<td>TLRN N. N.</td>
<td>1</td>
<td>99.66</td>
<td>0.33</td>
<td>4.85</td>
</tr>
<tr>
<td>Jordan Elman GA inputs</td>
<td>2</td>
<td>99.83</td>
<td>0.16</td>
<td>7.33</td>
</tr>
<tr>
<td>MLP NN, GA all, CV</td>
<td>1</td>
<td>98.56</td>
<td>1.92</td>
<td>21.55</td>
</tr>
<tr>
<td>Recurrent NN</td>
<td>1</td>
<td>99.15</td>
<td>0.83</td>
<td>25.51</td>
</tr>
</tbody>
</table>

the introduction of a more complex isoeelastic utility function can provide more customer-tailored solutions in higher efficiency. Finally in terms of the classifier within the EPOS model, the Hybrid TLRN of GA optimization in all layers and Cross Validation, with no hidden layers was the most optimal hybrid system that can support the decision making process.

9 Bibliography

[12] NeuroSolutions software
[16] Stagge P. and B.Senho,(1997), An extended elman net for modelling time series. *Int.Conf. on ANN*