Research Article

Cost-Sensitive Estimation of ARMA Models for Financial Asset Return Data

Minyoung Kim

*Department of Electronics & IT Media Engineering, Seoul National University of Science & Technology,
Seoul 139-743, Republic of Korea*

Correspondence should be addressed to Minyoung Kim; mikim21@gmail.com

Received 11 June 2015; Revised 30 August 2015; Accepted 2 September 2015

Academic Editor: Meng Du

Copyright © 2015 Minyoung Kim. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The autoregressive moving average (ARMA) model is a simple but powerful model in financial engineering to represent time-series with long-range statistical dependency. However, the traditional maximum likelihood (ML) estimator aims to minimize a loss function that is inherently symmetric due to Gaussianity. The consequence is that when the data of interest are asset returns, and the main goal is to maximize profit by accurate forecasting, the ML objective may be less appropriate potentially leading to a suboptimal solution. Rather, it is more reasonable to adopt an asymmetric loss where the model's prediction, as long as it is in the same direction as the true return, is penalized less than the prediction in the opposite direction. We propose a quite sensible asymmetric cost-sensitive loss function and incorporate it into the ARMA model estimation. On the online portfolio selection problem with real stock return data, we demonstrate that the investment strategy based on predictions by the proposed estimator can be significantly more profitable than the traditional ML estimator.

1. Introduction

In modeling time-series data, capturing the underlying statistical dependency of the variables of interest at current time on the historic data is central to accurate forecasting and faithful data representation. For financial time-series data especially (e.g., daily asset prices or returns) where a large amount of potential prognostic indicators is available, the development/analysis of sensible dynamic models as well as effective parameter estimation algorithms has been investigated significantly.

To account for statistical properties specific to financial sequences, several sophisticated dynamic time-series models have been developed: fairly natural autoregressive and/or moving average models [1], the conditional heteroscedastic models that represent dynamics of volatilities (variances) of the asset returns [2–4], and nonlinear models [5, 6] including bilinear models [7], threshold models [8], and regime switching models [9, 10].

Among those, the autoregressive moving average (ARMA) model [1] is the simplest while essential in the sense that most other models are equipped with at least the basic

ARMA components. The ARMA models appear in a wide spectrum of applications recently including filter design in signal processing [11], time-series analysis and model selection in computational statistics [12], and jump (large changes) modeling for asset prices in quantitative finance [13], to name just a few. For a time-series $\mathbf{y} = y_1, \dots, y_T$ (e.g., y_t is the asset return at the t th day), the ARMA(p, q) determines y_t by

$$y_t = \alpha^\top y_{t-p}^{t-1} + \beta^\top e_{t-q}^{t-1} + \beta_0 e_t + \rho, \quad (1)$$

where x_a^b indicates the vector $[x_a, x_{a+1}, \dots, x_b]^\top$. Here $e_t \sim \mathcal{N}(0, \gamma^2)$ is the stochastic Gaussian error term at time t where we assume iid across t 's. In (1) α , β , β_0 , and ρ are the model parameters. That is, y_t is dependent on p previous asset returns, q historic errors, and the current error e_t .

In this paper we consider a more general, recent stochastic extension of ARMA (abbreviated as sARMA) [14] (in contrast to the deterministic equation (1)) that adds a Gaussian noise to (1). Moreover, the extra covariates c_t (called *cross predictors*) are assumed available at time t ; for instance, they are typically economic indicators, market indices, and/or the previous

returns of other related assets. The sARMA(p, q) model can be specifically written as

$$y_t \mid y_{t-p}^{t-1}, e_{t-q}^t, c_t \sim \mathcal{N}(u_t, \sigma^2), \quad (2)$$

where

$$u_t = \alpha^\top y_{t-p}^{t-1} + \beta^\top e_{t-q}^{t-1} + \beta_0 e_t + \eta^\top c_t + \rho. \quad (3)$$

Here η is the weight vector (model parameters) for the cross predictor.

Hence, sARMA deals with Gaussian noisy observation (with variance σ^2), and it exactly reduces to the ARMA model in the limiting case $\sigma \rightarrow 0$. The noisy observation modeling of sARMA is beneficial in several aspects: not only does it merely account for the underlying noise process in the observation but also the model becomes fully stochastic, which allows principled probabilistic inference and model estimation even with missing data [14].

Given the observed sequence data, the parameters $\theta = \{\gamma^2, \alpha, \beta, \beta_0, \eta, \rho, \sigma^2\}$ of the sARMA model can be estimated by the expectation maximization (EM) algorithm [15]. Compared to the traditional Levenberg-Marquardt method for ARMA model estimation [1], the EM algorithm is beneficial for dealing with latent variables (i.e., the error terms) as well as any missing observations in an efficient and principled way. However, both estimators basically aim to achieve data likelihood maximization (ML) under the Gaussian model (2) [14] (with $\sigma \rightarrow 0$ corresponding to ARMA).

Due to the Gaussian observation modeling in sARMA, the ML estimation inherently aims to minimize a *symmetric* loss. In other words, letting $\hat{y}_t$ and $\bar{y}_t$ be the model forecast and the true value at time t , respectively, incorrect prediction $\hat{y}_t$ with the prediction error $\xi = |\hat{y}_t - \bar{y}_t|$ incurs the same amount of loss for both $\hat{y}_t = \bar{y}_t + \xi$ and $\hat{y}_t = \bar{y}_t - \xi$ (i.e., regardless of over- or underestimation). This strategy is far from being optimal especially for the asset return data as argued in the following.

The main goal is to maximize profit by accurate forecasting with the asset return data that encode signs (directions) toward profits. Traditional maximum likelihood (ML) estimator aims to minimize a loss function that is inherently symmetric and hence unable to exploit the property of the asset return data, leading to a suboptimal solution.

Suppose that our data $\mathbf{y}$ forms a sequence of daily stock log-returns, encoded as $y_t > 0$ (< 0) indicating that the stock price moves up (declines) on the t th day against the previous day. Now, consider a portfolio selection algorithm that makes an investment based on the forecast $\hat{y}_t$ given the information up to time t . The investment yields positive revenue when the signs of $\hat{y}_t$ and $\bar{y}_t$ are equal and the other way around. Hence, the prediction loss should be inherently asymmetric. Furthermore, when $\bar{y}_t < 0$, having $\hat{y}_t < 0$, even its underestimation (i.e., $\hat{y}_t < \bar{y}_t$) should be penalized less than the prediction with the opposite direction (i.e., $\hat{y}_t > 0$) because the former does not incur any loss in revenue but the latter does.

To address this issue, we propose a reasonable cost function that effectively captures the above idea of the intrinsic asymmetric profit/loss structure regarding asset return

data. Our cost function effectively encodes the goodness of matching in directions between true and model predicted asset returns, which is directly related to ultimate profits in the investment. We also provide an efficient optimization strategy based on the subgradient descent using the trust-region approximation, whose effectiveness is empirically demonstrated for the portfolio selection problem with real-world stock return data.

It is worth mentioning that there have been several other asymmetric loss functions proposed in the literature similar to ours. However, existing loss models merely focus on the asymmetry with respect to the ground-truth value point. For instance, the linex function [16, 17] is defined to be linear-exponential function of difference between predicted and ground-truth values. The linlin method [18] adopts a piecewise linear function where the change point is simply the ground-truth value. To the best of our knowledge, we are the first to derive the loss based on the matching the directions (signs) of the predicted and ground-truth returns. This effectively enables incorporating the critical information about directions of profits/losses, in turn leading to a more accurate forecasting model.

The rest of the paper is organized as follows. In the next section we suggest a novel sARMA estimation algorithm based on the cost-sensitive loss function: beginning with the overall objective, we derive the one-step predictor for the sARMA model in Section 2.1, provide details of the proposed cost function in Section 2.2, and state the optimization strategy in Section 2.3. The statistical inference algorithm for the sARMA model is also provided in full derivations in Section 2.4. In the empirical study in Section 3, we demonstrate the effectiveness of the proposed algorithm on the online portfolio selection problem with real data, where the significantly higher total profit is attained by the proposed approach than the investment based on the traditional ML-estimated sARMA model.

2. Cost-Sensitive Estimation

The proposed estimator for sARMA is based on the cost-sensitive loss of the model predicted one-step forecast value (denoted by $\hat{y}_t$) at each time t with respect to the true one (denoted by $\bar{y}_t$) available from data. More specifically, for a given data sequence $\mathbf{y} = y_1, \dots, y_T$, we aim to solve the optimization problem:

$$\min_{\theta} \sum_{t=R+1}^T C(\hat{y}_t(\theta), \bar{y}_t) + \lambda N(\theta), \quad (4)$$

where $R = \max(p, q)$.

Here $C(\hat{y}_t, \bar{y}_t)$ is the cost of predicting the asset return as $\hat{y}_t$ when the true value is $\bar{y}_t$. In Section 2.2 we define a reasonable cost function that faithfully incorporates the idea of asymmetric cost-sensitive loss discussed in the introduction.

In the objective, we also simultaneously minimize $N(\theta)$, the parameter regularizer that typically penalizes a nonsmooth sARMA model while preferring a smooth model (effectively achieved by encouraging the regression

parameters in θ close to 0) model. Specifically we use the L2 penalty, $N(\theta) = \|\alpha\|^2 + \|\beta\|^2 + \beta_0^2 + \|\eta\|^2$. The constant λ (>0) trades off the regularization against the prediction error cost.

Note also that in (4) we use the notation $\hat{y}_t(\theta)$ to emphasize the dependency of the model predicted $\hat{y}_t$ on θ . We use the principled maximum a posteriori (MAP) predictor estimated under the sARMA model, which is fully described in Section 2.1. The predictor is evaluated based on the inference on the latent error terms, which can be computed recursively where we give detailed derivations for the inference in Section 2.4.

2.1. One-Step Predictor for sARMA. Under the sARMA model, the predictive model at time t , given all available information $(y_1^{t-1}, e_1^R, c_1^t)$, is $P(y_t | y_1^{t-1}, e_1^R, c_1^t)$ for $t = R+1, R+2, \dots$. From this predictive model, one can make deterministic decision on the asset return at t , typically as the maximum-a-posteriori (MAP) estimation:

$$\hat{y}_t = \arg \max_{y_t} P(y_t | y_1^{t-1}, e_1^R, c_1^t). \quad (5)$$

Note that in the sARMA model, it is always assumed that we have at least R previous observations y_1^R and R previous error terms e_1^R . The error terms are simply assumed to be $e_1 = \dots = e_R = 0$ throughout the paper. Due to the linear Gaussianity of the sARMA's local conditional densities, we have Gaussian $P(y_t | y_1^{t-1}, e_1^R, c_1^t)$, and the MAP predictor (5) exactly coincides with the mean $\mathbb{E}[y_t | y_1^{t-1}, e_1^R, c_1^t]$.

In this section we derive the MAP (or mean) prediction $\hat{y}_t$ as a function of the sARMA model parameters θ , which can then be used in gradient evaluation for the optimization in (4). As is shown, the predictive distributions heavily resort to the posterior distributions of the error terms, namely, $P(e_{R+1}^t | y_1^t, e_1^R, c_1^t)$ for $t = R+1, \dots, T$. They are also Gaussians, and we denote them by (μ_t, Σ_t) in

$$P(e_{R+1}^t | y_1^t, e_1^R, c_1^t) = \mathcal{N}(\mu_t, \Sigma_t), \quad (6)$$

for $t = R+1, \dots, T$. Note that μ_t and Σ_t have dimensions $((t-R) \times 1)$ and $((t-R) \times (t-R))$, respectively. The full derivation of the error term posteriors is provided in Section 2.4.

In deriving $P(y_t | y_1^{t-1}, e_1^R, c_1^t)$, one may need to differentiate three cases for t : (i) $t = R+1$, (ii) $R+1 < t \leq R+q+1$, and (iii) $t > R+q+1$. The first case simply forms the initial condition which immediately follows from the local conditional model with marginalization of e_{R+1} . That is, when $t = R+1$,

$$P(y_{R+1} | y_1^R, e_1^R, c_1^{R+1}) = \mathcal{N}(b_{R+1}, \beta_0^2 \gamma^2 + \sigma^2), \quad (7)$$

where we define $b_t = \alpha^\top y_{t-p}^{t-1} + \eta^\top c_t + \rho$ for $t = R+1, \dots, T$.

We distinguish the second and third cases for the following reason: at time t , the previous q error terms are *fully* included in the time window $[R+1, t]$ in the latter case, while they are *partially* included in the former. Hence in the second case, we additionally deal with the error terms e_{t-q}^R which are always given as 0. Specifically, in the second case

($R+1 < t \leq R+q+1$), the terms e_{t-q}^t are partitioned into $(e_t, e_{R+1}^{t-1}, e_{t-q}^R = 0)$, and we have

$$P(y_t | y_1^{t-1}, e_1^R, c_1^t) = \sum_{e_{R+1}^t} P(y_t, e_{R+1}^t | y_1^{t-1}, e_1^R, c_1^t) \quad (8)$$

$$= \sum_{e_{R+1}^t} P(y_t | y_{t-p}^{t-1}, e_{t-q}^t, c_t) \cdot P(e_{R+1}^t | y_1^{t-1}, e_1^R, c_1^t) \quad (9)$$

$$= \sum_{e_{R+1}^t} \mathcal{N}(y_t; b_t + \beta^\top e_{t-q}^{t-1} + \beta_0 e_t, \sigma^2) \cdot P(e_t) \quad (10)$$

$$\cdot \mathcal{N}(e_{R+1}^{t-1}; \mu_{t-1}, \Sigma_{t-1})$$

$$= \sum_{e_{R+1}^t} \mathcal{N}(y_t; b_t + \beta^\top e_{t-q}^{t-1}, \beta_0^2 \gamma^2 + \sigma^2) \quad (11)$$

$$\cdot \mathcal{N}(e_{R+1}^{t-1}; \mu_{t-1}, \Sigma_{t-1})$$

$$= \mathcal{N}(y_t; b_t + \beta_2^\top \mu_{t-1}, \beta_2^\top \Sigma_{t-1} \beta_2 + \beta_0^2 \gamma^2 + \sigma^2). \quad (12)$$

In (12), we let β_2 be the subvector of β corresponding to e_{R+1}^{t-1} .

In the third case ($t > R+q+1$), we only need to deal with error terms e_{t-q}^t , and the predictive density is derived as follows:

$$P(y_t | y_1^{t-1}, e_1^R, c_1^t) = \sum_{e_{t-q}^t} P(y_t | y_{t-p}^{t-1}, e_{t-q}^t, c_t) \cdot P(e_t) \quad (13)$$

$$\cdot P(e_{t-q}^{t-1} | y_1^{t-1}, e_1^R, c_1^{t-1})$$

$$= \sum_{e_{t-q}^t} \mathcal{N}(e_{t-q}^{t-1}; \tilde{\mu}_{t-1}, \tilde{\Sigma}_{t-1}) \quad (14)$$

$$\cdot \mathcal{N}(y_t; b_t + \beta^\top e_{t-q}^{t-1}, \beta_0^2 \gamma^2 + \sigma^2)$$

$$= \mathcal{N}(y_t; b_t + \beta^\top \tilde{\mu}_{t-1}, \beta^\top \tilde{\Sigma}_{t-1} \beta + \beta_0^2 \gamma^2 + \sigma^2). \quad (15)$$

In (14), we introduce $\tilde{\mu}_{t-1}$ and $\tilde{\Sigma}_{t-1}$ as submatrices of μ_{t-1} and Σ_{t-1} taking the indices from $(t-q)$ to $(t-1)$ only.

In summary, the one-step predictor $\hat{y}_t(\theta)$ at t with all available information $(y_1^{t-1}, e_1^R, c_1^t)$ can be written as

$$\begin{aligned} \hat{y}_t(\theta) &= \begin{cases} \alpha^\top y_{t-p}^{t-1} + \eta^\top c_t + \rho & \text{if } t = R+1 \\ \alpha^\top y_{t-p}^{t-1} + \eta^\top c_t + \rho + \beta_2^\top \mu_{t-1} & \text{if } R+1 < t \leq R+q+1 \\ \alpha^\top y_{t-p}^{t-1} + \eta^\top c_t + \rho + \beta^\top \tilde{\mu}_{t-1} & \text{if } t > R+q+1. \end{cases} \quad (16) \end{aligned}$$

Note here that the means of the error term posteriors μ_{t-1} (and their subvectors $\tilde{\mu}_{t-1}$) have also dependency on the model parameters θ .

2.2. Proposed Cost Function. In this section we propose a cost function $C(\hat{y}_t, \bar{y}_t)$ (used in (4)) that effectively encodes the intrinsic asymmetric profit/loss structure regarding asset return data. To meet the motivating idea discussed in Section 1, we deal with two outstanding cases: the case when

the true $\bar{y}_t$ is positive and the case when $\bar{y}_t$ is negative. In each case, we further consider a certain margin ϵ (small positive, e.g., $\epsilon = 0.005$), where observing $\hat{y}_t > \epsilon$ indicates positive return with high certainty; on the other hand, having $0 \leq \bar{y}_t \leq \epsilon$ can be regarded differently as weak positivity and might be considered as noise. For the negative return, we have similar two regimes of different certainty levels.

We discuss the first case, $\bar{y}_t > \epsilon$. Depending on the value of $\hat{y}_t$, the cost functional changes over the four intervals: (i) $\hat{y}_t < 0$ incurs the highest loss with a super-linear penalty along the magnitude of $\hat{y}_t$ (we particularly choose a convex quadratic function), (ii) $\hat{y}_t \geq \bar{y}_t$, that is, overestimation, should be penalized the least, and we opt for an increasing linear function with a small slope, (iii) $\epsilon \leq \hat{y}_t < \bar{y}_t$ is an underestimation, but the prediction has certainty greater than a margin and thus is penalized less (we choose a linear function with slope slightly higher than the second case), and (iv) $0 \leq \hat{y}_t < \epsilon$ makes prediction in correct direction, but due to the weak certainty below the margin, we penalize it more severely than previous two regimes.

Our specific cost definition for $\bar{y}_t > \epsilon$ is as follows:

$$C(\hat{y}_t, \bar{y}_t) = \begin{cases} \frac{1}{2}s_a\hat{y}_t^2 - s_0\hat{y}_t + g_1 & \text{if } \hat{y}_t < 0 \\ -s_0(\hat{y}_t - \epsilon) + g_0 & \text{if } 0 \leq \hat{y}_t < \epsilon \\ -s_{\text{in}}(\hat{y}_t - \bar{y}_t) & \text{if } \epsilon \leq \hat{y}_t < \bar{y}_t \\ s_{\text{out}}(\hat{y}_t - \bar{y}_t) & \text{if } \hat{y}_t \geq \bar{y}_t \end{cases} \quad (17)$$

(when $\bar{y}_t > \epsilon$),

where we choose the constants as follows: $s_a = 4$, $s_0 = 2$, $s_{\text{in}} = 0.5$, $s_{\text{out}} = 0.3$. To make the cost function continuous, we define two offsets as follows: $g_0 = -s_{\text{in}}(\epsilon - \bar{y}_t)$ and $g_1 = g_0 + \epsilon s_0$.

In the case of $\bar{y}_t < -\epsilon$, we exactly penalize the prediction in the same way as the first situation. Specifically, the cost definition for $\bar{y}_t < -\epsilon$ is

$$C(\hat{y}_t, \bar{y}_t) = \begin{cases} \frac{1}{2}s_a\hat{y}_t^2 + s_0\hat{y}_t + h_1 & \text{if } \hat{y}_t > 0 \\ s_0(\hat{y}_t + \epsilon) + h_0 & \text{if } -\epsilon < \hat{y}_t \leq 0 \\ s_{\text{in}}(\hat{y}_t - \bar{y}_t) & \text{if } \bar{y}_t < \hat{y}_t \leq -\epsilon \\ -s_{\text{out}}(\hat{y}_t - \bar{y}_t) & \text{if } \hat{y}_t \leq \bar{y}_t \end{cases} \quad (18)$$

(when $\bar{y}_t < -\epsilon$),

where the same constants are used, and the offsets are now set as $h_0 = -s_{\text{in}}(\epsilon + \bar{y}_t)$ and $h_1 = h_0 + \epsilon s_0$.

For the uncertain (within the margin ϵ) return, we still conform to the strategy of encouraging the same direction as the true return. In the case of $0 < \bar{y}_t \leq \epsilon$, we assign small penalty for overestimation as long as it is in the correct direction, while rapidly growing quadratic loss for the

prediction toward opposite direction. To summarize, the cost for $0 < \bar{y}_t \leq \epsilon$ is

$$C(\hat{y}_t, \bar{y}_t) = \begin{cases} \frac{1}{2}s_a\hat{y}_t^2 - s_0\hat{y}_t + k_1 & \text{if } \hat{y}_t < 0 \\ -s_0(\hat{y}_t - \bar{y}_t) & \text{if } 0 \leq \hat{y}_t < \bar{y}_t \\ s_{\text{out}}(\hat{y}_t - \bar{y}_t) & \text{if } \hat{y}_t \geq \bar{y}_t \end{cases} \quad (19)$$

(when $0 < \bar{y}_t \leq \epsilon$),

where we set (for continuity) $k_1 = s_0\bar{y}_t$. The other case of $-\epsilon \leq \bar{y}_t < 0$ is similarly defined as

$$C(\hat{y}_t, \bar{y}_t) = \begin{cases} \frac{1}{2}s_a\hat{y}_t^2 + s_0\hat{y}_t + l_1 & \text{if } \hat{y}_t > 0 \\ s_0(\hat{y}_t - \bar{y}_t) & \text{if } \bar{y}_t < \hat{y}_t \leq 0 \\ -s_{\text{out}}(\hat{y}_t - \bar{y}_t) & \text{if } \hat{y}_t \leq \bar{y}_t \end{cases} \quad (20)$$

(when $-\epsilon \leq \bar{y}_t < 0$),

where $l_1 = -s_0\bar{y}_t$ for continuity of the cost function.

Finally when $\bar{y}_t = 0$, we can have a symmetric loss; for instance, $C(\hat{y}_t, \bar{y}_t = 0) = s_0|\hat{y}_t|$.

2.3. Optimization Strategy. In this section we briefly describe the optimization strategy for (4). We basically follow the subgradient descent [19, 20] where the derivative of the cost function with respect to θ can be derived as

$$\frac{\partial C(\hat{y}_t, \bar{y}_t)}{\partial \theta} = \frac{\partial \hat{y}_t}{\partial \theta} \cdot \frac{\partial C(\hat{y}_t, \bar{y}_t)}{\partial \hat{y}_t}. \quad (21)$$

Here, due to the nondifferentiability of the cost function (albeit continuous), we use the *subgradient* in place of the second part of RHS of (21).

Evaluating the first part, that is, the gradient of $\hat{y}_t$ with respect to θ , requires further endeavor. According to the functional form of $\hat{y}_t$ in (16), it has complex recursive dependency on θ mainly due to the error posterior means μ_{t-1} . Instead of exactly computing the derivative of μ_{t-1} , we address this issue by evaluating an approximate gradient by treating μ_{t-1} as a constant (constant evaluated at the current iterate θ). In consequence, we have a linear function of θ , and the gradient can be computed easily. However, the approximation (i.e., constant μ_{t-1} with respect to θ) is only valid in the vicinity of the current θ . Hence, to reduce the approximation error, we restrict the search space to be not much different from the current iterate (i.e., we search the next θ within the small-radius ball centered at the current θ , specifically $\|\theta - \theta_{\text{curr}}\| \leq \kappa$ for some small $\kappa > 0$). Our optimization strategy is closely related to the trust-region method [21], where the objective is approximated in the vicinity of the current parameters.

2.4. Inference in sARMA. In this section we give full derivations for statistical inference on the latent error variables e_{R+1}^t for each t , conditioned on the historic observations y_1^t and the cross predictors c_1^t in the sARMA model. That is, the posterior

densities $P(e_{R+1}^t | y_1^t, e_1^R, c_1^t)$ for $t = R+1, \dots, T$ are fully derived. In essence, these are all Gaussians, and as denoted in (6), we find the recursive formulas for the means and covariances (μ_t, Σ_t) . We also denote the inverse covariance Σ_t^{-1} by S_t .

Similarly as one-step predictive distributions, we consider three cases: (i) initial $t = R+1$, (ii) $R+1 < t \leq R+q+1$ where e_{t-q}^t fully contains what we need to infer, that is, e_{R+1}^t , and (iii) $t > R+q+1$ where we have to infer three groups of variables $(e_{R+1}^{t-q-1}, e_{t-q}^{t-1}, e_t)$. The initial case ($t = R+1$) is straightforwardly derived as follows:

$$P(e_{R+1} | y_1^{R+1}, e_1^R, c_1^{R+1}) \propto P(y_{R+1} | y_{R+1-p}^R, e_{R+1-q}^{R+1}, c_{R+1}) \quad (22)$$

$$\begin{aligned} & \cdot P(e_{R+1} | y_1^R, e_1^R, c_1^R) \\ &= \mathcal{N}(y_{R+1}; b_{R+1} + \beta^\top e_{R+1-q}^R + \beta_0 e_{R+1}, \sigma^2) \\ & \cdot \mathcal{N}(e_{R+1}; 0, \gamma^2) \end{aligned} \quad (23)$$

$$= \mathcal{N}\left(e_{R+1}; \frac{\beta_0 \gamma^2}{\beta_0^2 \gamma^2 + \sigma^2} (y_{R+1} - b_{R+1}), \frac{\sigma^2 \gamma^2}{\beta_0^2 \gamma^2 + \sigma^2}\right). \quad (24)$$

In (23), $b_t = \alpha^\top y_{t-p}^{t-1} + \eta^\top c_t + \rho$ as before, and we use $e_{R+1-q}^R = 0$. The theorem of product of two Gaussians is applied to yield (24) from (23). This forms the initial posterior mean and inverse covariance as follows:

$$\begin{aligned} \mu_{R+1} &= \frac{\beta_0 \gamma^2 (y_{R+1} - b_{R+1})}{\beta_0^2 \gamma^2 + \sigma^2}, \\ S_{R+1} (= \Sigma_{R+1}^{-1}) &= \frac{\beta_0^2 \gamma^2 + \sigma^2}{\sigma^2 \gamma^2}. \end{aligned} \quad (25)$$

We next deal with the second case; that is, $R+1 < t \leq R+q+1$. We partition e_{t-q}^t into three parts: $E_1 = e_{t-q}^R$, $E_2 = e_{t-q}^{t-1}$, and $E_3 = e_t$. The parameter vector β for e_{t-q}^{t-1} is accordingly divided into subvectors β_1 (for E_1) and β_2 (for E_2). We only need to infer e_{R+1}^t ; thus E_2 and E_3 and the conditional density can be derived as follows:

$$\begin{aligned} P(e_{R+1}^t | y_1^t, e_1^R, c_1^t) &= P\left(\begin{bmatrix} E_2 \\ E_3 \end{bmatrix} | y_1^t, e_1^R, c_1^t\right) \\ &\propto \mathcal{N}(y_t; b_t + \beta_1^\top E_1 + \beta_2^\top E_2 + \beta_0 E_3, \sigma^2) \cdot \mathcal{N}(E_2; \end{aligned} \quad (26)$$

$$\begin{aligned} & \mu_{t-1}, S_{t-1}^{-1}) \cdot \mathcal{N}(E_3; 0, \gamma^2) \\ &\propto \exp\left(-\frac{E_3^2}{2\gamma^2} - \frac{1}{2} (E_2 - \mu_{t-1})^\top S_{t-1} (E_2 - \mu_{t-1}) \right. \\ & \quad \left. - \frac{1}{2\sigma^2} \left(\underbrace{(y_t - b_t)}_{=a_t} - \beta_2^\top E_2 - \beta_0 E_3\right)^2\right) \end{aligned} \quad (27)$$

$$\begin{aligned} &\propto \exp\left(-\frac{1}{2} \left(\frac{1}{\gamma^2} + \frac{\beta_0^2}{\sigma^2}\right) E_3^2 \right. \\ & \quad \left. - \frac{1}{2} E_2^\top \left(S_{t-1} + \frac{\beta_2 \beta_2^\top}{\sigma^2}\right) E_2 - E_2^\top \left(\frac{\beta_2 \beta_0}{\sigma^2}\right) E_3 \right. \\ & \quad \left. + \left(S_{t-1} \mu_{t-1} + \frac{\beta_2 a_t}{\sigma^2}\right)^\top E_2 + \frac{\beta_0 a_t}{\sigma^2} E_3\right). \end{aligned} \quad (28)$$

To derive μ_t and S_t , we rearrange the exponent of (28) as a canonical quadratic form in terms of (E_2, E_3) . It is not difficult to have the following formulas after some algebra:

$$\mu_t = S_t^{-1} \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix}, \quad (29)$$

$$S_t = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^\top & Q_{22} \end{bmatrix},$$

where

$$\begin{aligned} Z_1 &= S_{t-1} \mu_{t-1} + \frac{\beta_2 a_t}{\sigma^2}, \\ Z_2 &= \frac{\beta_0 a_t}{\sigma^2}, \\ Q_{11} &= S_{t-1} + \frac{\beta_2 \beta_2^\top}{\sigma^2}, \\ Q_{12} &= \frac{\beta_2 \beta_0}{\sigma^2}, \\ Q_{22} &= \frac{1}{\gamma^2} + \frac{\beta_0^2}{\sigma^2}. \end{aligned} \quad (30)$$

Finally, for the third case ($t > R+q+1$), the variables to be inferred (i.e., e_{R+1}^t) are partitioned into three groups of variables: $E_1 = e_{R+1}^{t-q-1}$, $E_2 = e_{t-q}^{t-1}$, and $E_3 = e_t$. Here, E_1 and E_2 , when concatenated, yield a vector of the same dimension as μ_{t-1} , and we partition μ_{t-1} accordingly as $\mu_{t-1}(1)$ and $\mu_{t-1}(2)$. Similarly, S_{t-1} is partitioned into (2×2) blocks, and we denote them by $S_{t-1}(i, j)$ for $i, j \in \{1, 2\}$. The posterior can then be written as

$$\begin{aligned} P(e_{R+1}^t | y_1^t, e_1^R, c_1^t) &= P\left(\begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix} | y_1^t, e_1^R, c_1^t\right) \\ &\propto \mathcal{N}(y_t; b_t + \beta^\top E_2 + \beta_0 E_3, \sigma^2) \cdot \mathcal{N}\left(\begin{bmatrix} E_1 \\ E_2 \end{bmatrix}; \mu_{t-1}, \right. \\ & \quad \left. S_{t-1}^{-1}\right) \cdot \mathcal{N}(E_3; 0, \gamma^2) \end{aligned} \quad (31)$$

$$\propto \exp \left(-\frac{E_3^2}{2\gamma^2} - \frac{1}{2\sigma^2} \left(\underbrace{(y_t - b_t)}_{=a_t} - \beta^\top E_2 - \beta_0 E_3 \right)^2 \right) - \frac{1}{2} \quad (32)$$

$$\begin{aligned} & \cdot \sum_{i=1}^2 \sum_{j=1}^2 (E_i - \mu_{t-1}(i))^\top S_{t-1}(i, j) (E_j - \mu_{t-1}(j)) \\ & \propto \exp \left(-\frac{1}{2} \left(\frac{1}{\gamma^2} + \frac{\beta_0^2}{\sigma^2} \right) E_3^2 - \frac{1}{2} E_1^\top S_{t-1}(1, 1) E_1 - \frac{1}{2} \right. \\ & \quad \cdot E_2^\top \left(S_{t-1}(2, 2) + \frac{\beta\beta^\top}{\sigma^2} \right) E_2 - E_1^\top S_{t-1}(1, 2) E_2 \\ & \quad - E_2^\top \left(\frac{\beta\beta_0}{\sigma^2} \right) E_3 + (S_{t-1}(1, 1) \mu_{t-1}(1) \\ & \quad + S_{t-1}(1, 2) \mu_{t-1}(2))^\top E_1 + \left(S_{t-1}(2, 2) \mu_{t-1}(2) \right. \\ & \quad \left. + S_{t-1}(2, 1) \mu_{t-1}(1) + \frac{\beta a_t}{\sigma^2} \right)^\top E_2 + \frac{\beta_0 a_t}{\sigma^2} E_3 \Big). \end{aligned} \quad (33)$$

Similar to the second case, we derive μ_t and S_t by rearranging the exponent of (33) as a canonical quadratic form in terms of (E_1, E_2, E_3) . The resulting formulas are as follows:

$$\begin{aligned} \mu_t &= S_t^{-1} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix}, \\ S_t &= \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{12}^\top & Q_{22} & Q_{23} \\ Q_{13}^\top & Q_{23}^\top & Q_{33} \end{bmatrix}, \end{aligned} \quad (34)$$

where

$$\begin{aligned} Z_1 &= S_{t-1}(1, 1) \mu_{t-1}(1) + S_{t-1}(1, 2) \mu_{t-1}(2), \\ Z_2 &= S_{t-1}(2, 2) \mu_{t-1}(2) + S_{t-1}(2, 1) \mu_{t-1}(1) + \frac{\beta a_t}{\sigma^2}, \\ Z_3 &= \frac{\beta_0 a_t}{\sigma^2}, \\ Q_{11} &= S_{t-1}(1, 1), \\ Q_{12} &= S_{t-1}(1, 2), \\ Q_{13} &= 0, \\ Q_{22} &= S_{t-1}(2, 2) + \frac{\beta\beta^\top}{\sigma^2}, \end{aligned}$$

$$Q_{23} = \frac{\beta\beta_0}{\sigma^2},$$

$$Q_{33} = \frac{1}{\gamma^2} + \frac{\beta_0^2}{\sigma^2}.$$

(35)

3. Empirical Study

In this section we empirically test the effectiveness of the proposed sARMA estimation method. In particular we deal with the task of portfolio selection on the real-world dataset comprising daily closing prices from Dow Jones Industrial Average (DJIA).

We consider the task of online portfolio selection (OLPS) problem with real stock return data. We begin with a brief description of the OLPS problem. Assuming there are M different stocks to invest in daily basis, at the beginning of day t , the historic closing stock prices up to day $t-1$, denoted by $\{P_\tau\}_{\tau=0}^{t-1}$, are available, where P_τ is M -dim vector whose j th element $P_\tau(j)$ is the price of the j th ticker. Using the information, you decide the portfolio allocation vector x_t , a nonnegative M -dim vector that sums to 1 (i.e., $\sum_{j=1}^M x_t(j) = 1$). Assuming no short positioning is allowed, $x_t(j)$ is the proportion of the whole budget to be invested in the j th stock for $j = 1, \dots, M$.

The portfolio strategy is thus a function that maps the historic market information (say $\{P_\tau\}_{\tau=0}^{t-1}$) to the price prediction P_t . The sARMA-based portfolio strategy can be built by estimating sARMA models, one for each stock ticker j , for the stock log-return data; namely, $y_t = \log P_t - \log P_{t-1}$ (here, we drop the dependency on j for simplicity). Then the predicted $\hat{y}_t$ can be used to decide the proportion of the budget to be invested in the j th ticker at time t . A reasonable strategy is to make no investment (i.e., $x_t(j) = 0$) if $\hat{y}_t < 0$, while forcing $x_t(j)$ to be proportional to $\hat{y}_t$ if $\hat{y}_t > 0$.

To evaluate the performance of a portfolio strategy, we use the popular (running) relative cumulative wealth (RCW) defined as $RCW(t) = A_t/A_0$, where A_t is the total budget at time t . Thus $RCW(t)$ indicates the total budget return at time t compared to the initial budget, and the portfolio strategy that yields high $RCW(t)$ for many epochs t 's is regarded as a good strategy. Assuming that there is no transaction cost, it is not difficult to see that $RCW(t) = \prod_{\tau=1}^t r_\tau^\top x_\tau$ where we define the price relative vector $r_t = P_t/P_{t-1}$ (division element-wisely).

Hence, in the sARMA model, it is crucial to accurately forecast the returns, and we compare the model estimated by our cost-sensitive loss with the one using traditional ML estimation. For each approach, we estimate M sARMA models, one for each stock return, and once the predicted returns $\hat{y}_t(j)$'s at t are obtained, $x_t(j)$'s are decided as follows: $x_t(j) = 0$ if $\hat{y}_t(j) < \epsilon$ and $x_t(j) = 1/M_1$ where M_1 is the number of j 's with $\hat{y}_t(j) \geq \epsilon$. We also contrast them with the fairly standard market portfolio strategy which sets $x_t(j)$ to be proportional to the total market volume (i.e., the product of the price and the total number of shares) of the ticker j .

We test the above-mentioned three portfolio strategies on the real-world data, the 30 tickers' daily closing prices from

FIGURE 1: Running relative cumulative wealth for three competing portfolio selection strategies on the DJIA stock return data.

Dow Jones Industrial Average (DJIA) for about 15 months beginning on January 14, 2001, which amounts to about 340 daily records. The dataset is available publicly (<http://www.mysmu.edu.sg/faculty/chhoi/olps/datasets.html>), and the detailed description can be found in [22]. The stock tickers appear to be considerably correlated with one another and include GE, Microsoft, AMEX, GM, COCA-COLA, and Intel.

In the sARMA estimation, we set $p = q = 2$, and the cross predictors c_t are defined to be the returns of the other 29 stocks at day $t - 1$. The parameter λ in our cost-sensitive estimation is empirically chosen. First, the average costs attained, that is, $(1/T) \sum_{t=1}^T C(\hat{y}_t, \bar{y}_t)$, which are further averaged over M different models, are 114.0994 for the ML-estimated sARMA and 0.0030 for the proposed cost-sensitive sARMA. This implies that the proposed estimation method yields a far more accurate prediction performance than the traditional ML method in terms of the proposed cost function.

Next, we depict the running RCW scores for three competing portfolio strategies in Figure 1. As shown, the proposed approach (sARMA-cost) achieves the highest profits consistently for almost all t 's during the time horizon, significantly outperforming the market strategy. The ML-based sARMA estimator performs the worst, which can be explained by its attempt at fitting a model to overall data, not accounting for the asymmetric loss structure for the asset return data, especially regarding the directions of return predictions. In the end, for $t > 250$, the proposed method indeed gives positive return (i.e., $RCW(t) > 1$) whereas the other two methods suffer from substantial budget loss ($RCW(t) < 1$). This again signifies the effectiveness of the cost-sensitive loss minimization in the return prediction.

4. Conclusion

In this paper we have introduced a novel ARMA model identification method that exploits the asymmetric loss structure

for the financial asset return data. The proposed cost function effectively encodes the goodness of matching in directions between true and model predicted asset returns, which is directly related to ultimate profits in the investment. We have provided the subgradient-based optimization using the trust-region approximation, where it has been empirically shown to work well for the portfolio selection problem in a real-world situation.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This study is supported by National Research Foundation of Korea (NRF-2013R1A1A1076101).

References

- [1] G. Box, G. M. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting & Control*, Prentice Hall, Upper Saddle River, NJ, USA, 1994.
- [2] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," *Econometrica*, vol. 50, no. 4, pp. 987–1007, 1982.
- [3] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, no. 3, pp. 307–327, 1986.
- [4] D. B. Nelson, "Conditional heteroskedasticity in asset returns: a new approach," *Econometrica*, vol. 59, no. 2, pp. 347–370, 1991.
- [5] M. B. Priestley, *Non-Linear and Non-Stationary Time Series Analysis*, Academic Press, London, UK, 1988.
- [6] H. Tong, *Non-Linear Time Series: A Dynamical System Approach*, Oxford University Press, Oxford, UK, 1990.
- [7] J. Liu and P. J. Brockwell, "On the general bilinear time-series model," *Journal of Applied Probability*, vol. 25, no. 3, pp. 553–564, 1988.
- [8] H. Tong, *Threshold Models in Nonlinear Time Series Analysis*, vol. 21 of *Lecture Notes in Statistics*, Springer, New York, NY, USA, 1983.
- [9] J. D. Hamilton, "A new approach to the economic analysis of nonstationary time series and the business cycle," *Econometrica*, vol. 57, no. 2, pp. 357–384, 1989.
- [10] C. J. Kim and C. R. Nelson, *State Space Models with Regime Switching*, Academic Press, New York, NY, USA, 1999.
- [11] J. T. Ollkonen, S. Ahtinen, K. Jarvinen, and H. Ollkonen, "Adaptive matrix/vector gradient algorithm for design of IIR filters and ARMA models," *Journal of Signal and Information Processing*, vol. 4, no. 2, pp. 212–217, 2013.
- [12] P. X.-K. Song, R. K. Freeland, A. Biswas, and S. Zhang, "Statistical analysis of discrete-valued time series using categorical ARMA models," *Computational Statistics & Data Analysis*, vol. 57, no. 1, pp. 112–124, 2013.
- [13] S. Laurent, C. Lecourt, and F. C. Palm, "Testing for jumps in conditionally Gaussian ARMA-GARCH models, a robust approach," *Computational Statistics & Data Analysis*, 2014.

- [14] B. Thiesson, D. M. Chickering, D. Heckerman, and C. Meek, "ARMA time-series modeling with graphical models," in *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI '04)*, pp. 552–560, 2004.
- [15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.
- [16] H. Varian, "A Bayesian approach to real estate assessment," in *Studies in Bayesian Econometrics and Statistics in Honor of L.J. Savage*, S. E. Feinberg and A. Zellner, Eds., North-Holland, Amsterdam, The Netherlands, 1974.
- [17] A. Zellner, "Bayesian estimation and prediction using asymmetric loss functions," *Journal of the American Statistical Association*, vol. 81, no. 394, pp. 446–451, 1986.
- [18] C. W. J. Granger, "Prediction with a generalized cost of error function," *Journal of the Operational Research Society*, vol. 20, pp. 199–207, 1969.
- [19] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions*, Springer, Berlin, Germany, 1985.
- [20] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, Mass, USA, 1999.
- [21] D. C. Sorensen, "Newton's method with a model trust region modification," *SIAM Journal on Numerical Analysis*, vol. 19, no. 2, pp. 409–426, 1982.
- [22] B. Li and S. C. Hoi, "On-line portfolio selection: a survey," Tech. Rep., Nanyang Technological University, Singapore, 2012.