

## Multi-Event Decision Making over Multivariate Time Series

**Chun-Kit Ngan, Alexander Brodsky<sup>1</sup>, and Jessica Lin<sup>2</sup>**

Department of Computer Science, George Mason University  
4400 University Drive MSN 4A5, Fairfax, Virginia 22030-4422, United States  
cngan@gmu.edu, brodsky@gmu.edu, jessica@cs.gmu.edu  
<http://cs.gmu.edu/~brodsky/><sup>1</sup>, <http://cs.gmu.edu/~jessica/><sup>2</sup>

### ABSTRACT

We propose a multidimensional time-point model and algorithm to solve Multi-Event Expert Query Parametric Estimation (ME-EQPE) problems over multivariate time series. Our proposed model and algorithm combine the strengths of both domain-knowledge-based and formal-learning-based approaches to learn optimal decision parameters for maximizing utility over multivariate time series. More specifically, our approach solves the decision optimization problems to maximize the utility from multiple decision time points, as well as maintaining an optimality of the learned multiple sets of decision parameters in their respective events during the computations. We show that our approach guarantees a remarkable forecasting result by using the learned multiple sets of decision parameters.

**Keywords:** Statistical Learning, Prediction, Multivariate Time-Series, Decision-Support

## 1. INTRODUCTION

Making decisions over multivariate time series is an important topic which has gained significant interest in the past decade, as two or more time series are often observed simultaneously in many fields. In business and economics, financial analysts and researchers monitor daily stock prices, weekly interest rates, and monthly price indices to analyze different states of stock markets. In medical studies, physicians and scientists measure patients' diastolic and systolic blood pressure over time and electrocardiogram tracings to evaluate the patients' health of respiratory systems. In social sciences, sociologists and demographers study annual birth rates, mortality rates, accident rates, and various crime rates to dig out hidden social problems within a community. The purpose of these measures over multivariate time series is to assist the specialists in understanding the same problem from different perspectives within particular domains. The events identified and detected over multivariate time series can lead the professionals to make better decisions and take timely, appropriate actions. Those events may include index bottoms and tops in financial markets, irregular readings on blood pressure and pulse anomalies on electrocardiogram, as well as low birth but high death rates in a population region.

Currently, existing approaches to identifying and detecting those interesting events can be roughly divided into two categories: domain-knowledge-based and formal-learning-based. The former relies solely on domain expert knowledge. Based on their knowledge and experiences, domain experts determine the conditions that trigger the events of interest. Consider one particular example of the timely event detection of certain conditions in the stock market, e.g., the bear market bottom, that can provide investors a valuable insight into the best investment opportunity. Such identification and detection can aid in the task of decision-making and the determination of action plans. To assist users in making better decisions and determinations, domain experts have identified a set of financial indices that can be used to determine a bear market bottom. The indices include the S&P 500 percentage decline (SPD), Coppock Guide (CG), Consumer Confidence point drop (CCD), ISM Manufacturing Survey (ISM), and Negative Leadership Composite "Distribution" (NLCD). If these indices satisfy the pre-defined, parameterized conditions, e.g.,  $SPD < -20\%$ ,  $CG < 0$ , etc., [1], it signals that the best period for the investors to buy the stocks is approaching. Often these parameters may reflect some realities since they are set by the domain experts based on their past experiences, observations, intuitions, and domain knowledge. However, they are not always accurate. In addition, the parameters are static, but the problem that we deal with is often dynamic in nature. The market is constantly impacted by many unknown and uncontrollable factors from the business surroundings. Thus, this approach lacks a formal mathematical computation that dynamically learns the parameters to meet the needs of the changing environment.

An alternative approach is to utilize formal learning methods such as non-linear logit regression models. [2, 3, 4] The logit regression models are used to predict the occurrence of an event (0 or 1) by learning parametric coefficients of the logistic distribution function of the explanatory variables. This is done based on the historical data by applying nonlinear regression models and Maximum Likelihood Estimation (MLE). The main challenge concerning using formal learning methods to support decision-making is that they do not always produce satisfactory results, as they do not consider incorporating domain knowledge into their formal learning approaches. Without domain experts' knowledge, formal learning methods become computationally intensive and time consuming. The whole model building is an iterative and interactive process, including model formulation, parameter estimation, and model evaluation. Despite enormous improvements in computer software in recent years,

fitting such nonlinear quantitative decision model is not a trival task, especially if the parameter learning process involves multiple explanatory variables, i.e., high dimensionality. Moreover, working with high-dimensional data creates difficult challenges, a phenomenon known as the “curse of dimensionality.” Specifically, the amount of observations required in order to obtain good estimates increases exponentially with the increase of dimensionality. In addition, many learning algorithms do not scale well on high dimensional data due to the high computational cost. The parameter computations by formal-learning-based approaches, e.g., logit regression model, are complicated and costly, and they lack the consideration of integrating experts’ domain knowledge into the learning process – a step that could potentially reduce the dimensionality. Clearly, both approaches, domain-knowledge-based and formal-learning-based, do not take advantage of each other to learn the optimal decision parameters, which are then used to monitor the events and make better recommendations.

To mitigate the shortcomings of the existing approaches, we have proposed a mathematical model, Expert Query Parametric Estimation (EQPE), and algorithm, Checkpoint [5, 6], that combines the strengths of both domain-knowledge-based and formal-learning-based approaches. More specifically, we take the template of conditions identified by domain experts—such template consists of inequalities of values in the time sequences—and “parameterize” it, e.g.,  $SPD < p_1$ . Our goal is to efficiently find parameters that maximize the objective function, e.g., earnings in our financial example, in which the objective function is dependent on the time points from which the parameters are learned. The Checkpoint algorithm guarantees a true optimal time point and has the complexity of  $O(kN \log N)$ , where  $N$  is the size of the learning data set, and  $k$  is the number of parametric time series. As a proof of concept, we have conducted an experiment in the financial domain and shown that our algorithm is more effective and produces results that are superior to the two approaches mentioned above.

However, the EQPE model and the Checkpoint algorithm are only able to learn one set of decision parameters for one particular event at a single time point, whereas there are many real-world scenarios that the parameter learning is at multiple time points in sequence. For instance, consider our financial example again, in which the investors would like to decide on both when the S&P 500 fund is purchased at  $t_{purchased}$  and when the fund is sold at  $t_{sold}$  rather than either  $t_{purchased}$  or  $t_{sold}$  only. Using the EQPE model and the Checkpoint algorithm, the investors would not be able to obtain the optimal decision parameters simultaneously for the two interrelated decision time points and then to gain the maximal earning of the S&P 500 index fund.

To address the shortcomings of our one-dimensional-time-point model and algorithm, in this paper, we propose an extended model, Multi-Event Expert Query Parametric Estimation (ME-EQPE), and an algorithm, Multidimensional  $M$ -Checkpoint, to solve the problems. The new extended model and algorithm not only maintain the advantages of one-dimensional-time-point model, EQPE, and algorithm, Checkpoint, but also learn decision parameters at multiple, inter-related time points optimally for mutli-events. More specifically, the contributions of this paper are as follows:

- Combines the strengths of both domain-knowledge-based and formal-learning-based approaches to solve the decision optimization problems that involve multiple decision time points to maximize the utility.
- Maintains an optimality of the learned multiple sets of decision parameters in their respective events during the computations.
- Guarantees a satisfactory forecasting result by using the learned multiple sets of decision parameters.

The rest of the paper is organized as follows. In Section 2, we review the mathematical model, EQPE, the computational algorithm, Checkpoint, and the experimental evaluation on our financial example, i.e., the bear market bottom. Section 3 describes our extended model, ME-EQPE, and algorithm,  $M$ -Checkpoint, that simultaneously learn multiple sets of decision parameters at their respective time points in sequence. Using the ME-EQPE model and the  $M$ -Checkpoint algorithm, we illustrate the remarkably experimental results on the stock market data, i.e., the parameter learning at multiple time points for the “Besy Buy” and “Best Sell” investment, in Section 4. Note that our new extension is applicable to solve the problems in different domains and scenarios. Section 5 contains the conclusions and future work.

## 2. LEARNING DECISION PARAMETERS AT A SINGLE TIME POINT

In this section, we first explain the methodologies used in the parameter learning from a single time point. More specifically, we review the mathematical formulations of the Expert Query Parametric Estimation (EQPE) problem and solution. [5, 6] We then illustrate the computational algorithm, Checkpoint, used for the parameter learning.

### 2.1 Expert Query Parametric Estimation (EQPE) Model

The goal of an EQPE is to find optimal values of decision parameters that maximize an objective function over historical, multivariate time series. For an EQPE problem being constructed, we need to define a set of mathematical notations and a model for it. We assume that the time domain  $\mathbf{T}$  is represented by a set of natural numbers:  $\mathbf{T} = \mathbf{N}$ , and that we are also given a vector of  $n$  real-valued parameter variables  $(p_1, p_2, \dots, p_n)$ .

**Definition 1. Time Series:** A time series  $S$  is a function  $S: \mathbf{T} \rightarrow \mathbf{R}$ , where  $\mathbf{T}$  is the time domain, and  $\mathbf{R}$  is the set of real numbers.

**Definition 2. Parametric Monitoring Constraint:** A parametric monitoring constraint  $C(S_1(t), S_2(t), \dots, S_k(t), p_1, p_2, \dots, p_n)$  is a symbolic expression in terms of  $S_1(t), S_2(t), \dots, S_k(t), p_1, p_2, \dots, p_n$ , where  $S_1(t), S_2(t), \dots, S_k(t)$  are time series,  $t \in \mathbf{T}$  is a time point, and  $(p_1, p_2, \dots, p_n)$  is a vector of parameters.

We assume a constraint  $C$  written in a language that has the truth-value interpretation  $I: \mathbf{R}^k \times \mathbf{R}^n \rightarrow \{\text{True}, \text{False}\}$ , i.e.,  $I(C(S_1(t), S_2(t), \dots, S_k(t), p_1, p_2, \dots, p_n)) = \text{True}$  if and only if the constraint  $C$  is satisfied at the time point  $t \in \mathbf{T}$  and with the parameters  $(p_1, p_2, \dots, p_n) \in \mathbf{R}^n$ . In this paper, we focus on conjunctions of inequality constraints:  $C(S_1(t), S_2(t), \dots, S_k(t), p_1, p_2, \dots, p_n) = \bigwedge_i (S_i(t) \text{ op } p_i)$ , where  $\text{op} \in \{\leq, =, \geq\}$ .

**Definition 3. Time Utility Function:** A time utility function  $U$  is a function  $U: \mathbf{T} \rightarrow \mathbf{R}$ .

**Definition 4. Objective Function:** Given a time utility function  $U: \mathbf{T} \rightarrow \mathbf{R}$  and a parametric constraint  $C$ , an objective function  $O$  is a function  $O: \mathbf{R}^n \rightarrow \mathbf{R}$ , which maps a vector of  $n$  parameters in  $\mathbf{R}^n$  to a real value  $R$ , defined as follows. For  $(p_1, p_2, \dots, p_n) \in \mathbf{R}^n$ ,  $O(p_1, p_2, \dots, p_n) \stackrel{\text{def}}{=} U(t)$ , where  $U$  is the utility function, and  $t \in \mathbf{T}$  is the earliest time point that satisfies  $C$ , i.e.,

- (1)  $S_1(t) \text{ op}_1 p_1 \wedge S_2(t) \text{ op}_2 p_2 \wedge \dots \wedge S_n(t) \text{ op}_n p_n$  is satisfied, and
- (2) There does not exist  $0 \leq t' < t$ , such that  $S_1(t') \text{ op}_1 p_1 \wedge S_2(t') \text{ op}_2 p_2 \wedge \dots \wedge S_n(t') \text{ op}_n p_n$  is satisfied.

**Definition 5.** *Expert Query Parametric Estimation (EQPE) Problem:* An EQPE problem is a tuple  $\langle \hat{S}, \hat{P}, C, U \rangle$ , where  $\hat{S} = \{S_1, S_2, \dots, S_k\}$  is a set of  $k$  time series,  $\hat{P} = \{p_1, p_2, \dots, p_n\}$  is a set of  $n$  real-value parameter variables,  $C$  is a parametric constraint in  $\hat{S}$  and  $\hat{P}$ , and  $U$  is a time utility function.

Intuitively, a solution to an EQPE problem is an instantiation of values into the vector  $\hat{P}$  of  $n$  real-value parameters that maximizes the objective  $O$ .

**Definition 6.** *Expert Query Parametric Estimation (EQPE) Solution:* A solution to the EQPE problem  $\langle \hat{S}, \hat{P}, C, U \rangle$  is  $\text{argmax } O(p_1, p_2, \dots, p_n)$ , i.e., the (estimated) values of parameters,  $p_1, p_2, \dots, p_n$ , that maximize  $O$ , where  $O$  is the objective function corresponding to  $U$ .

The base time series in our financial example, i.e., detecting the bearish market bottom, are shown in Table 1. We suppose that the first starting date in any time-series data set is  $t = 0$ .

**Table 1: Base Time-Series Data**

<i>Base Time Series S</i>	<i>Abbreviation</i>
S&P 500	SP500(t)
Coppock Guide	CG(t)
Consumer Confidence	CC(t)
ISM Manufacturing Survey	ISM(t)
Negative Leadership Composite	NLC(t)

Note that some base time series are the direct inputs, whereas some are used to derive another set of time series. For instance, the derived time series in our case study are shown in Table 2.

**Table 2: Derived Time-Series Data**

<i>Derived Time Series S</i>	<i>Abbreviation</i>
Percentage decline in SP(t) at the time point t	SPD(t)
Points drop in CC(t) at the time point t	CCD(t)
Number of consecutive days in Bear Market “DISTRIBUTIOIN” of NLC(t) at and before the time point t	NLCD(t)
Time Utility Earning at the time point t, i.e., the index fund is bought at t and sold at $t_s$ , where $t_s$ is the last day of the learning data set	Earning(t)

The decision parameters used in the case study are defined in Table 3.

**Table 3: Decision Parameters**

<i>Parameter</i>	<i>Interpretation</i>
$p_1$	Test if SPD(t) is less than $p_1$ at $t$ .
$p_2$	Test if CG(t) is less than $p_2$ at $t$ .
$p_3$	Test if CCD(t) is less than $p_3$ at $t$ .
$p_4$	Test if ISM(t) is less than $p_4$ at $t$ .
$p_5$	Test if NLCD(t) is greater than $p_5$ at $t$ .

Let us consider the following constraint  $C$  as an illustration:

$$C(\text{SPD}(t), \text{CG}(t), \text{CCD}(t), \text{ISM}(t), \text{NLCD}(t), p_1, p_2, p_3, p_4, p_5) \\ = \text{SPD}(t) < p_1 \wedge \text{CG}(t) < p_2 \wedge \text{CCD}(t) < p_3 \wedge \text{ISM}(t) < p_4 \wedge \text{NLCD}(t) > p_5$$

It means that the parametric monitoring constraint  $\mathbf{C}$  is satisfied, i.e., its interpretation is **True**, if the above inequalities with the decision parameters are satisfied at the time point  $t$ . The interpretation also indicates that the monitoring event occurs. We assume that the investor buys the S&P 500 index fund at the decision variable time  $t$  and sell it at the given  $t_s$ , which is the last day of the given training data set. The earning function  $SP(t_s)/SP(t) - 1 \in \mathbf{R}$  is the utility, which is maximized by choosing the optimal value  $t \in \mathbf{T}$ , where  $SP(t_s)$  and  $SP(t)$  are the sell and buy value of the S&P 500 index fund at the time  $t_s$  and  $t$  respectively. The EQPE problem and solution for our financial example can be constructed by putting the considered time series, parameters, constraints, and functions to the definitions shown in Table 4.

**Table 4: EQPE Problem and Solution Formulation for the S&P 500 Index Fund**

<i>Problem and Solution</i>
<p><i>Problem:</i>  <math>\langle \dot{\mathbf{S}}, \dot{\mathbf{P}}, \mathbf{C}, U \rangle</math>, where  <math>\dot{\mathbf{S}} = \{SPD, CG, CCD, ISM, NLCD\}</math>  <math>\dot{\mathbf{P}} = \{p_1, p_2, p_3, p_4, p_5\}</math>  <math>\mathbf{C} = SPD(t) &lt; p_1 \wedge CG(t) &lt; p_2 \wedge CCD(t) &lt; p_3 \wedge ISM(t) &lt; p_4 \wedge NLCD(t) &gt; p_5</math>  <math>U = SP(t_s)/SP(t) - 1</math></p> <p><i>Solution:</i>  <math>\operatorname{argmax} \mathbf{O}(p_1, p_2, p_3, p_4, p_5) \stackrel{\text{def}}{=} U(t)</math></p>

The values of the optimal decision parameters can be determined by using the learning algorithm, Checkpoint. [5, 6] Before explaining the Checkpoint algorithm in detail, we first review the concept of *Dominance*.

**Definition 7. Dominance**  $\succ$ : Given an EQPE problem  $\langle \dot{\mathbf{S}}, \dot{\mathbf{P}}, \mathbf{C}, U \rangle$  and any two time points  $t, t' \in \mathbf{T}$ , we say that  $t'$  dominates  $t$ , denoted by  $t' \succ t$ , if the following conditions are satisfied:

- (1)  $0 \leq t' < t$ , and
- (2)  $\forall (p_1, p_2, \dots, p_n) \in \mathbf{R}^n, C(S_1(t), S_2(t), \dots, S_k(t), p_1, p_2, \dots, p_n) \rightarrow C(S_1(t'), S_2(t'), \dots, S_k(t'), p_1, p_2, \dots, p_n)$ .

Intuitively,  $t'$  dominates  $t$  if for any selection of parametric values, the query constraint satisfaction at  $t$  implies the satisfaction at  $t'$ . Clearly, the dominated time points should be discarded when the optimal time point is being determined. We formally claim that:

**Claim 1** - Given the conjunctions of inequality constraints,  $S_1(t) \text{ op}_1 p_1 \wedge S_2(t) \text{ op}_2 p_2 \wedge \dots \wedge S_k(t) \text{ op}_k p_k$  and the two time points  $t', t$  such that  $0 \leq t' < t, t' \succ t$  if and only if  $S_1(t') \text{ op}_1 p_1 \wedge S_2(t') \text{ op}_2 p_2 \wedge \dots \wedge S_k(t') \text{ op}_k p_k$ .

For example, suppose there are three time series  $S_1, S_2, S_3$  and three decision parameters  $p_1, p_2, p_3$ . And the constraints are  $C(S_1(t), S_2(t), S_3(t), p_1, p_2, p_3) = S_1(t) \geq p_1 \wedge S_2(t) \geq p_2 \wedge S_3(t) \leq p_3$ . Also assume the values for  $S_1, S_2$ , and  $S_3$  at the time point  $t_1, t_2$ , and  $t_3$  respectively in Table 5.

**Table 5: Values of  $S_1, S_2, S_3$ , and  $U$  at the time point  $t_1, t_2$ , and  $t_3$**

<i>Time</i>	$S_1$	$S_2$	$S_3$	$U$
$t_1$	13	27	3	10

$t_2$	25	15	2	200
$t_3$	10	20	5	150

In this case, the time point  $t_3$  is dominated because there is a time point  $t_1$  that make the inequality,  $S_1(t_1) \geq S_1(t_3) \wedge S_2(t_1) \geq S_2(t_3) \wedge S_3(t_1) \leq S_3(t_3)$ , equal to true. On the contrary, for all  $t' < t$ , if  $S_1(t') \neg\text{op}_1 S_1(t) \vee S_2(t') \neg\text{op}_2 S_2(t) \vee \dots \vee S_n(t') \neg\text{op}_n S_n(t)$  is satisfied,  $t$  is not dominated by  $t'$  denoted by  $t' \not\prec t$ . Let us consider the same example above. Because  $S_1(t_1) < S_1(t_2) \vee S_3(t_1) > S_3(t_2)$ ,  $t_2$  is not dominated.

## 2.2 Checkpoint Algorithm and Experimental Evaluation

Conceptually, we can search a particular set of parameters  $\{p_1, p_2, \dots, p_n\}$  which is at the earliest time point  $t$  that is not dominated by *any*  $t'$  such that the value of the objective function  $O$  is maximal among all the instantiations of values into parameters. However, the problem of this approach is that for every single parameter set at  $t$  in a learning data set, the parameter set at  $t$  has to be examined with all the previous sets of parameters at  $t'$  for checking the non-dominance before the optimal solution can be found. In fact, due to the quadratic nature, the conceptual approach is time consuming and expensive particularly if the size of the learning data set is significantly large. Instead, the Checkpoint algorithm [5, 6] uses the KD-tree data structure and searching algorithm [7, 8, 9] to evaluate whether a time point  $t$  is dominated based on the **Claim 1** for checking the non-dominance. The pseudo code of the algorithm is:

**Input:**  $\langle \dot{S}, \dot{P}, C, U \rangle$

**Output:**  $p[1 \dots k]$  is an array of the optimal parameters that maximize the objective.

**Data Structures:**

1.  $N$  is the size of the learning data set.
2.  $T_{kd}$  is a KD tree that stores the parameter vectors that are not dominated so far.
3.  $\text{MaxT}$  is the time point that gives the maximal  $U$  so far, denoted by  $\text{MaxU}$ .

**Processing:**

**STEP 1:**  $T_{kd} := \langle S_1(0), S_2(0), \dots, S_k(0) \rangle$ ;  $\text{MaxT} := 0$ ;  $\text{MaxU} := U(0)$ ;

**STEP 2:** **FOR**  $t := 1$  **TO**  $N - 1$  **DO** {

Non-Dominance Test: Query the  $T_{kd}$  to find if there exists a point  $(p_1, p_2, \dots, p_k)$  in the  $T_{kd}$ , which is in the range  $[S_1(t), \infty) \times [S_2(t), \infty) \times \dots \times [S_k(t), \infty)$ .

IF (NOT AND  $t$  is not dominated AND  $U(t) > \text{MaxU}$ ) THEN

Add  $\langle S_1(t), S_2(t), \dots, S_k(t) \rangle$  to  $T_{kd}$ ;

$\text{MaxT} := t$ ;

$\text{MaxU} := U(t)$ ;

}

**STEP 3:** **FOR**  $i := 1$  **TO**  $k$  **DO** {

$p[i] := S_i(\text{MaxT})$ ;

}

**STEP 4:** **RETURN**  $p[1 \dots k]$ ;

Clearly, the first time point is not dominated because there is no time point preceding it. Therefore,  $\langle S_1(0), S_2(0), \dots, S_k(0) \rangle$  can be added to  $T_{kd}$ .  $0$  and  $U(0)$  can be assigned to  $\text{MaxT}$  and  $\text{MaxU}$  respectively. Using the Checkpoint algorithm step by step for the problem shown

in Table 5, we can search through a particular set of parameters  $\{p_1, p_2, p_3\}$  which is at the earliest time point  $t$  that is not dominated by *any*  $t'$  such that the value of the utility function  $U$  is maximal. In **STEP 1**, the  $\langle S_1(t_1), S_2(t_1), S_3(t_1) \rangle$  is added to the  $T_{kd}$  since it is the first time point. Then  $t_1$  and  $U(t_1)$  are assigned to MaxT and MaxU respectively. In **STEP 2**,  $t_2$  is *not dominated* because  $S_1(t_1) < S_1(t_2) \wedge S_2(t_1) > S_2(t_2) \wedge S_3(t_1) > S_3(t_2)$  does not satisfy the Claim 1. However,  $t_3$  is *dominated* because  $S_1(t_1) > S_1(t_3) \wedge S_2(t_1) > S_2(t_3) \wedge S_3(t_1) < S_3(t_3)$  does satisfy the Claim 1.  $\langle S_1(t_2), S_2(t_2), S_3(t_2) \rangle$  is added to the  $T_{kd}$  because  $t_2$  is *not dominated* and  $U(t_2) > U(t_1)$ . Thus  $t_2$  and  $U(t_2)$  are assigned to MaxT and MaxU respectively. In **STEP 3**,  $p[1] := S_1(\text{MaxT})$ ,  $p[2] := S_2(\text{MaxT})$ , and  $p[3] := S_3(\text{MaxT})$  in the for-loop statement. In **STEP 4**, the algorithm returns 25, 15, and 2.

**Theorem 1:** For  $N$  parameter vectors in the data set, the Checkpoint algorithm correctly computes an EQPE solution, i.e.,  $\text{argmax } \mathbf{O}(p_1, p_2, p_3, p_4, p_5)$ , where  $\mathbf{O}$  is the objective function of the EQPE problem, with the complexity  $O(kN \log N)$ .

Using the Checkpoint algorithm, we can obtain the optimal decision parameters and the maximal earning from the training data set for the financial problem shown in Table 6. The time complexity of the MLE for the logit regression model is  $O(k^2N)$ , where  $k$  is the number of decision parameters, and  $N$  is the size of the learning data set. For the Checkpoint algorithm, the complexity is  $O(kN \log N)$ . Using the decision parameters from the financial expert (i.e., -20%, 0, -30, 45, 180 days), the logit regression model, and the Checkpoint algorithm, the ‘‘Best Buy’’ opportunities in stock and their earnings are shown in Table 7. Note that the Checkpoint algorithm considerably outperforms both the financial expert’s criteria and the logit regression model.

**Table 6: Optimal Parameters and Maximum Earning (%) from the Learning Data Set<sup>1</sup>**

$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$\mathbf{O}(p_1, p_2, p_3, p_4, p_5)$
-29.02	-20.01	-26.61	49	70	53.37

**Table 7: Investors’ Earning of the S&P 500 Index Fund from the Test Data Set<sup>2</sup>**

<i>Decision Approach</i>	<i>Best Buy</i>	<i>S&amp;P 500 Index</i>	<i>Earning%</i>
Financial Expert’s Criteria	10/09/08	909.92	1.03
Logit Regression Model	11/26/08	887.68	3.56
Checkpoint Algorithm with Financial Expert’s Template	03/10/09	719.6	27.8

### 3. LEARNING DECISION PARAMETERS AT MULTIPLE TIME POINTS

For solving the formulations which the utility is determined from the multiple time points, a Multi-Event Expert Query Parametric Estimation (ME-EQPE) model is proposed. The model assists users finding optimal values of decision parameter *vectors* that maximize an objective function and satisfy their corresponding parametric constraints over historical, multivariate time series. In this section, we describe our extension based on the EQPE model and Checkpoint algorithm presented in Section 2.1 and 2.2 respectively.

#### 3.1 Multi-Event Expert Query Parametric Estimation (ME-EQPE) Model

Now we consider a sequence of  $m$  events that we would like to detect. Each event  $i$  has a

<sup>1</sup> The learning data set is from 06/01/1997 to 06/30/2005.

<sup>2</sup> The test data set is from 07/01/2005 to 06/30/2009 that is the sell date of the fund with the value of 919.32.



vector of real parameters  $\vec{P}_i$  that will be learned from the training data set of its multivariate-parametric-time-series vector  $\vec{S}_i(t)$ . After the learning process, the vector of the learned parameters  $\vec{P}_i$  is used to detect the occurrence of the event  $i$ .

Before explaining the new learning algorithm, we first extend a set of mathematical notations and a model for the ME-EQPE problem. We still assume that the time domain  $\mathbf{T}$  is represented by the set of natural numbers:  $\mathbf{T} = \mathbf{N}$ . We are also given a set of time sequences  $(\vec{S}_1(t), \vec{S}_2(t), \dots, \vec{S}_m(t))$ , and the learning algorithm generates a set of  $m$  vectors of real parameters  $(\vec{P}_1, \vec{P}_2, \dots, \vec{P}_m)$ , where  $\vec{S}_i(t) = (S_1(t), S_2(t), \dots, S_{k_i}(t)) \in R^{k_i}$  and  $\vec{P}_i = (p_1, p_2, \dots, p_{n_i}) \in R^{n_i}$  for  $1 \leq i \leq m$ ,  $k_i, n_i \in \mathbf{Z}^+$ ,  $k_i$  is the number of parametric time series in  $\vec{S}_i(t)$  for the event  $i$ ,  $n_i$  is the number of parameters in  $\vec{P}_i$  for the event  $i$ , and  $m$  is the number of time points that are corresponding to their events  $i$  being monitored.

**Definition 8. Parametric Monitoring Constraint:** A parametric monitoring constraint  $C_i(\vec{S}_i(t), \vec{P}_i)$  is a symbolic expression in terms of  $\vec{S}_i(t)$  and  $\vec{P}_i$ , where  $\vec{S}_i(t)$  is the  $i^{\text{th}}$  time sequence vector at the time  $t$ , and  $\vec{P}_i$  is the  $i^{\text{th}}$  decision parameter vector for  $1 \leq i \leq m$ .

We assume a constraint  $C_i$  written in a language that has the truth-value interpretation  $I: R^{k_i} \times R^{n_i} \rightarrow \{\text{True}, \text{False}\}$ , i.e.,  $I(C_i(\vec{S}_i(t), \vec{P}_i)) = \text{True}$  if and only if the constraint  $C_i$  is satisfied at the time point  $t \in \mathbf{T}$  with the vectors, time sequence  $\vec{S}_i(t) = (S_1(t), S_2(t), \dots, S_{k_i}(t)) \in R^{k_i}$  and parameter  $\vec{P}_i = (p_1, p_2, \dots, p_{n_i}) \in R^{n_i}$ . Again we focus on a combinational conjunction of inequality constraints:  $C_i(\vec{S}_i(t), \vec{P}_i) = S_1(t) \text{ op } p_1 \wedge S_2(t) \text{ op } p_2 \wedge \dots \wedge S_{k_i}(t) \text{ op } p_{n_i}$ , where  $\text{op} \in \{\leq, =, \geq\}$ .

**Definition 9. Time Utility Function:** A time utility function  $U$  is a function  $U: \mathbf{T}^m \rightarrow \mathbf{R}$ , where  $m$  is the number of time points.

**Definition 10. Objective Function:** Given a time utility function  $U: \mathbf{T}^m \rightarrow \mathbf{R}$  and  $m$  parametric constraints  $(C_1, C_2, \dots, C_m)$ , an objective function  $O$  is a function  $O: R^{n_1} \times R^{n_2}, \dots, R^{n_m} \rightarrow \mathbf{R}$ , which maps a set of vectors of parameters on  $R^{n_1} \times R^{n_2}, \dots, R^{n_m}$  to a real value  $R$ , defined as follows: For a set of  $(\vec{P}_1, \vec{P}_2, \dots, \vec{P}_m)$ ,  $O(\vec{P}_1, \vec{P}_2, \dots, \vec{P}_m) \stackrel{\text{def}}{=} U(t_1, t_2, \dots, t_m)$ , where  $U$  is the utility function, and  $(t_1, t_2, \dots, t_m)$  are the earliest time points that satisfies their corresponding parametric constraints, i.e.,

- (1)  $0 \leq t_1 < t_2 < \dots < t_m$
- (2)  $C_1(\vec{S}_1(t_1), \vec{P}_1), C_2(\vec{S}_2(t_2), \vec{P}_2), \dots, C_m(\vec{S}_m(t_m), \vec{P}_m)$  are satisfied, and
- (3) There does not exist  $(0 \leq t'_1 < t'_2 < \dots < t'_m)$  such that
  - (a)  $t'_1 \leq t_1, t'_2 \leq t_2, \dots, t'_m \leq t_m$
  - (b) for some  $i = 1, 2, \dots, m$ ,  $t'_i < t_i$ , and
  - (c)  $C_1(\vec{S}_1(t'_1), \vec{P}_1), C_2(\vec{S}_2(t'_2), \vec{P}_2), \dots, C_m(\vec{S}_m(t'_m), \vec{P}_m)$  are satisfied.

**Definition 11. Multi-Event Expert Query Parametric Estimation (ME-EQPE) Problem:** A ME-EQPE problem is a tuple  $\langle \dot{\mathbf{S}}, \dot{\mathbf{P}}, \mathbf{C}, \mathbf{U} \rangle$ , where  $\dot{\mathbf{S}} = \{\vec{S}_1(t), \vec{S}_2(t), \dots, \vec{S}_m(t)\}$  is a set of  $m$  vectors of time sequences,  $\dot{\mathbf{P}} = \{\vec{P}_1, \vec{P}_2, \dots, \vec{P}_m\}$  is a set of  $m$  decision parameter vectors,  $\mathbf{C} = (C_1, C_2, \dots, C_m)$  is a set of  $m$  parametric constraints in  $\dot{\mathbf{S}}$  and  $\dot{\mathbf{P}}$ , and  $\mathbf{U}$  is a time utility function.

Intuitively, a solution to a ME-EQPE problem is an instantiation of values into all the parameter vectors,  $\vec{P}_1, \vec{P}_2, \dots, \vec{P}_m$ , that maximize the objective.

**Definition 12.** *Multi-Event Expert Query Parametric Estimation (ME-EQPE) Solution:* A solution to the ME-EQPE problem  $\langle \dot{S}, \dot{P}, C, U \rangle$  is  $\text{argmax } \mathbf{O}(\vec{P}_1, \vec{P}_2, \dots, \vec{P}_m)$ , i.e., the estimated values of all the parameter vectors,  $\vec{P}_1, \vec{P}_2, \dots, \vec{P}_m$ , that maximize  $\mathbf{O}$ , where  $\mathbf{O}$  is the objective function corresponding to  $U$ .

**Definition 13.** *Time Length ( $T_L$ ):*  $T_L$  is the length of the time duration in terms of the number of business days that the events among them may occur.

The base time series, the derived time series, and the decision parameters in our financial example, i.e., the “Best Buy” and “Best Sell” opportunities in the investment, are shown in Table 8, Table 9, and Table 10 respectively.

**Table 8: Base Time-Series Data**

<i>Base Time Series S</i>	<i>Abbreviation</i>
S&P 500	SP500(t)
Coppock Guide	CG(t)
Consumer Confidence	CC(t)
ISM Manufacturing Survey	ISM(t)
Negative Leadership Composite “Distribution”	NLC(t)
Negative Leadership Composite “Selling Vacuum”	NLCSV(t)

**Table 9: Derived Time-Series Data**

<i>Derived Time Series S</i>	<i>Abbreviation</i>
Percentage decline in SP(t) at the time point t	SPD(t)
Percentage increase in SP(t) at the time point t	SPI(t)
Points drop in CC(t) at the time point t	CCD(t)
Points increase in CC(t) at the time point t	CCI(t)
Number of consecutive days in Bear Market “DISTRIBUTIOIN” of NLC(t) at and before the time point t	NLCD(t)
Time Utility Earning at the time points $t_1$ and $t_2$ , i.e., the index fund is bought at $t_1$ and sold at $t_2$ of the learning data set	Earning( $t_1, t_2$ )

**Table 10: Decision Parameters**

<i>Parameter</i>	<i>Interpretation</i>
$p_1$	Test if SPD(t) is less than $p_1$ at $t_1$ .
$p_2$	Test if CG(t) is less than $p_2$ at $t_1$ .
$p_3$	Test if CCD(t) is less than $p_3$ at $t_1$ .
$p_4$	Test if ISM(t) is less than $p_4$ at $t_1$ .
$p_5$	Test if NLCD(t) is greater than $p_5$ at $t_1$ .
$q_1$	Test if SPI(t) is greater than or equal to $q_1$ at $t_2$ .
$q_2$	Test if CG(t) is greater than or equal $q_2$ at $t_2$ .
$q_3$	Test if CCI(t) is greater than or equal $q_3$ at $t_2$ .
$q_4$	Test if ISM(t) is greater than or equal $q_4$ at $t_2$ .
$q_5$	Test if NLCSV(t) is greater than $q_5$ at $t_2$ .

The constraints  $C_1$  and  $C_2$  are illustrated as follows:

$$C_1(\text{SPD}(t_1), \text{CG}(t_1), \text{CCD}(t_1), \text{ISM}(t_1), \text{NLCD}(t_1), p_1, p_2, p_3, p_4, p_5)$$

$$= \text{SPD}(t_1) < p_1 \wedge \text{CG}(t_1) < p_2 \wedge \text{CCD}(t_1) < p_3 \wedge \text{ISM}(t_1) < p_4 \wedge \text{NLCD}(t_1) > p_5$$

$$\begin{aligned} & C_2(\text{SPI}(t_2), \text{CG}(t_2), \text{CCI}(t_2), \text{ISM}(t_2), \text{NLCSV}(t_2), q_1, q_2, q_3, q_4, q_5) \\ & = \text{SPI}(t_2) \geq q_1 \wedge \text{CG}(t_2) \geq q_2 \wedge \text{CCI}(t_2) \geq q_3 \wedge \text{ISM}(t_2) \geq q_4 \wedge \text{NLCSV}(t_2) > q_5 \end{aligned}$$

The earning function  $U(t_1, t_2) = \text{SP}(t_2) / \text{SP}(t_1) - 1 \in \mathbf{R}$  is the utility, which is maximized by choosing the optimal value  $t_1$  and  $t_2 \in \mathbf{T}$ , where  $\text{SP}(t_2)$  and  $\text{SP}(t_1)$  are the sell and buy value of the S&P 500 index fund at the time  $t_2$  and  $t_1$  respectively. The ME-EQPE problem and solution for our financial example can be constructed by putting the considered time sequence vectors, parameter vectors, constraints, and functions to the definitions shown in Table 11.

**Table 11: ME-EQPE Problem and Solution Formulation for the S&P 500 Index Fund**

<i>Problem and Solution</i>
<p><i>Problem:</i>  <math>\langle \vec{S}, \vec{P}, C, U \rangle</math>, where  <math>\vec{S} = \{\vec{S}_1(t_1), \vec{S}_2(t_2)\}</math>, where <math>\vec{S}_1(t_1) = (\text{SPD}(t_1), \text{CG}(t_1), \text{CCD}(t_1), \text{ISM}(t_1), \text{NLCD}(t_1))</math> and  <math>\vec{S}_2(t_2) = (\text{SPI}(t_2), \text{CG}(t_2), \text{CCI}(t_2), \text{ISM}(t_2), \text{NLCSV}(t_2))</math>  <math>\vec{P} = \{\vec{P}_1, \vec{P}_2\}</math>, where <math>\vec{P}_1 = (p_1, p_2, p_3, p_4, p_5)</math> and <math>\vec{P}_2 = (q_1, q_2, q_3, q_4, q_5)</math>  <math>C = \{C_1, C_2\}</math>, where <math>C_1 = \text{SPD}(t_1) &lt; p_1 \wedge \text{CG}(t_1) &lt; p_2 \wedge \text{CCD}(t_1) &lt; p_3 \wedge \text{ISM}(t_1) &lt; p_4 \wedge \text{NLCD}(t_1) &gt; p_5</math> and  <math>C_2 = \text{SPI}(t_2) \geq q_1 \wedge \text{CG}(t_2) \geq q_2 \wedge \text{CCI}(t_2) \geq q_3 \wedge \text{ISM}(t_2) \geq q_4 \wedge \text{NLCSV}(t_2) &gt; q_5</math>  <math>U = \text{SP}(t_2) / \text{SP}(t_1) - 1</math></p> <p><i>Solution:</i>  <math>\text{argmax } O(\vec{P}_1, \vec{P}_2) \stackrel{\text{def}}{=} U(t_1, t_2)</math></p>

The values of the optimal decision parameters can be determined by using the learning algorithm, *M*-Checkpoint.

### 3.2 Multidimensional *M*-Checkpoint Algorithms for the ME-EQPE Problem

The extended version, *M*-Checkpoint algorithm, which keeps using the KD-tree data structure and searching techniques to evaluate whether or not the time point  $t_1$  of the first event is dominated. The pseudo code of the *M*-Checkpoint algorithm is:

**Input:**  $\langle \vec{S}, \vec{P}, C, U \rangle$

**Output:**  $p[i][j]$  is a two-dimensional array of the optimal parameter vectors that maximize the objective, where  $1 \leq i \leq m$ ,  $1 \leq j \leq n_i$ , and  $m, n_i \in \mathbf{Z}^+$ .

**Data Structures:**

1.  $N$  is the size of the learning data set.
2.  $T_{kd}$  is a KD tree that stores the parameter vectors that are not dominated so far for the 1<sup>st</sup> event.
3.  $\text{isDominated}[t_i]$  is a boolean array to signal that a time point  $t_i$  of the 1<sup>st</sup> event is dominated, i.e.,  $\text{isDominated}[t_i] := \text{true}$ , by at least one previous time point, or else  $\text{isDominated}[t_i] := \text{false}$ .
4.  $\text{MaxT}[i]$  is the array of time points that gives the maximal  $U$  so far, denoted by  $\text{MaxU}$ , where  $1 \leq i \leq m$  and  $\text{MaxT}[1] < \text{MaxT}[2] < \dots < \text{MaxT}[m]$ .

**Initialization:** *FOR*  $i := 1$  *TO*  $m$  *DO* {  
      $t_i := \text{MaxT}[i] := 0$ ;  
 }

$\text{MaxU} := U(0, 0, \dots, 0) := 0$ ;

$T_L := 365$ ; // Assume that the event occurrences among them are within 365 business days.

**Pre-processing:**

**STEP 1:**  $T_{kd} := \vec{S}_1(0)$ ;  $\text{isDominated}[0] = \text{false}$ ;

**STEP 2:** *FOR*  $t_l := 1$  *TO*  $N - 1$  *DO* {  
 Non-Dominance Test: Query the  $T_{kd}$  to find if there exists a point  $(p_1, p_2, \dots, p_k)$  in the  $T_{kd}$ , which is in the range  $\prod_{j=1}^k [S_{j_1}(t_1), \infty)$ .  
 IF (NOT AND  $t_l$  is not dominated) THEN  
     Add  $\vec{S}_1(t_1)$  to  $T_{kd}$ ;  
      $\text{isDominated}[t_l] = \text{false}$ ;  
 ELSE  
      $\text{isDominated}[t_l] = \text{true}$ ;  
 ENDIF  
}

**Processing:**

**STEP 1:** *FOR* ( $t_m := N - 1$ ;  $t_m \geq 0$ ;  $t_m := t_m - 1$ ) *DO* {  
     *FOR* ( $t_{m-1} := t_m - 1$ ;  $t_{m-1} \geq 0$  AND  $t_{m-1} \geq t_m - T_L$  AND  $t_{m-1} \neq t_m$ ;  $t_{m-1} := t_{m-1} - 1$ ) *DO* {  
         *FOR* ( $t_{m-2} := t_{m-1} - 1$ ;  $t_{m-2} \geq 0$  AND  $t_{m-2} \geq t_{m-1} - T_L$  AND  $t_{m-2} \neq t_{m-1}$ ;  $t_{m-2} := t_{m-2} - 1$ ) *DO* {  
             :  
             :  
             :  
             *FOR* ( $t_1 := t_2 - 1$ ;  $t_1 \geq 0$  AND  $t_1 \geq t_2 - T_L$  AND  $t_1 \neq t_2$ ;  $t_1 := t_1 - 1$ ) *DO* {  
                 IF ((NOT  $\text{isDominated}[t_1]$ ) AND  $U(t_1, t_2, \dots, t_m) > \text{MaxU}$ ) THEN  
                      $\text{MaxU} := U(t_1, t_2, \dots, t_m)$ ;  
                      $\text{MaxT}[1] := t_1, \text{MaxT}[2] := t_2, \dots, \text{MaxT}[m] := t_m$ ;  
                 ENDIF  
             }  
         }  
     }  
 }

**STEP 2:** *FOR*  $i := 1$  *TO*  $m$  *DO* {  
     *FOR*  $j := 1$  *TO*  $n_i$  *DO* {  
          $p[i][j] := S_{j_i}(\text{MaxT}[i])$ ;  
     }  
 }

**STEP 3:** *RETURN*  $p[i][j]$ , where  $1 \leq i \leq m$  and  $1 \leq j \leq n_i$ ;

**Theorem 2:** For  $m$  sets of  $N$  parameter vectors in the data set, the  $M$ -Checkpoint algorithm correctly computes a ME-EQPE solution, i.e.,  $\text{argmax } \mathbf{O}(\vec{P}_1, \vec{P}_2, \dots, \vec{P}_m)$ , where  $\mathbf{O}$  is the objective function of the ME-EQPE problem, with the complexity  $O(N^m)$ .

#### 4. STOCK MARKET EXPERIMENTAL EVALUATION

The time complexity of the MLE for the logit regression model is  $O(mk^2N)$ , where  $m$  is the number of time points,  $k$  is the number of decision parameters, and  $N$  is the size of the learning data set. For the  $M$ -Checkpoint algorithm, the complexity is  $O(N^m)$ . Although the time complexity of the MLE is more efficient than that of the  $M$ -Checkpoint algorithm, the  $M$ -Checkpoint algorithm generates the vectors of optimal learned parameters that maximize the earning from the training data set for this financial problem shown in Table 12. Using the  $M$ -Checkpoint algorithm, we can use the optimal decision parameters for detecting both investment events, the “Best Buy” and “Best Sell” opportunities on the S&P 500 Index Fund. Using the logit regression model and the  $M$ -Checkpoint algorithm, the “Best Buy” and “Best Sell” opportunities in this investment and their earnings are shown in Table 13. Note that the  $M$ -Checkpoint algorithm considerably outperforms the logit regression model.

**Table 12: Optimal Parameters and Maximum Earning (%) from the Learning Data Set<sup>3</sup>**

$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$O(\vec{P}_1, \vec{P}_2)$ = 55.70%
-15.34	19.70	-16.89	48.70	79	
$q_1$	$q_2$	$q_3$	$q_4$	$q_5$	
0.09	16.24	-4.60	54.9	0	

**Table 13: Investors’ Earning of the S&P 500 Index Fund from the Test Data Set<sup>4</sup>**

Decision Approach	Best Buy	S&P 500 Index	Best Sell	S&P 500 Index	Earning%
Logit Regression Model	10/14/08	998.01	09/03/09	1003.24	0.52%
$M$ -Checkpoint Algorithm with Financial Expert’s Template	10/31/08	968.75	09/07/10	1091.84	12.71%

\* Note that the financial experts do not provide the decision parameters that can be used to determine the “Best Buy” and “Best Sell” opportunity in the sequence of occurrence.

#### 5. CONCLUSIONS AND FUTURE WORK

To the best of our knowledge, this is the first paper to propose a multidimensional time-point model and algorithm to solve the ME-EQPE problems over multivariate time series. The new extensions, ME-EQPE model and  $M$ -Checkpoint algorithm, provide the contributions that 1) combine the strengths of both domain-knowledge-based and formal-learning-based approaches to learn the optimal decision parameters; 2) solve the decision optimization problems to maximize the utility from multiple decision time points; 3) maintain an optimality of the learned multiple sets of decision parameters in their respective events during the computations; and 4) guarantee a remarkably forecasting result by using the learned multiple sets of decision parameters. However, there are still many open research questions, for example, what other methodologies, such as heuristic and decomposition, would be able to solve the same problems at a low time complexity, and whether those approaches could present a reasonably forecasting results.

<sup>3</sup> The learning data set is from 06/01/1997 to 01/31/2004.

<sup>4</sup> The test data set is from 02/01/2004 to 03/31/2011.

## 6. REFERENCES

1. Stack, J.B., "Technical and Monetary Investment Analysis", Vol 9 Issue 3 & 5. InvesTech Research, 2009.
2. Dougherty, C., "Introduction to Econometrics (Third Edition)", Oxford University Press, 2007.
3. Hansen, B.E., "Econometrics", University of Wisconsin, 2010. <http://www.ssc.wisc.edu/~bhansen/econometrics/Econometrics.pdf>.
4. Heij, D., De Boer, P., Franses, P.H., Kloek, T., and Van Dijk, H.K., "Econometric Methods with Applications in Business and Economics", Oxford University Press, 2004.
5. Ngan, C.K., Brodsky, A., and Lin, J., "Decisions on Multivariate Time Series: Combining Domain Knowledge with Utility Maximization", The 15th IFIP WG8.3 International Conference on Decision Support Systems, 2010.
6. Ngan, C.K., Brodsky, A., and Lin, J., "A Service Framework for Learning, Querying, and Monitoring Multivariate Time Series", The 13th International Conference on Enterprise Information Systems, 2011.
7. Bentley, J.L., "Multidimensional Binary Search Trees Used for Associative Searching", Communications of the ACM, Vol 18 Issue 09, p. 509-517, 1975.
8. Bentley, J.L., "Multidimensional Binary Search Trees in Database Applications", Vol 5 Issue 04, p. 333-340. IEEE Transactions on Software Engineering, 1979.
9. Samet, H., "Foundations of Multidimensional and Metric Data Structures", Morgan Kaufmann, 2006.
10. Bellman, R., "Adaptive Control Processes: A Guided Tour", Princeton, University Press, 1961.
11. Brodsky, A., Henshaw, S.M., and Whittle, J., "CARD: A Decision-Guidance Framework and Application for Recommending Composite Alternatives", 2nd ACM International Conference on Recommender Systems, 2008.
12. Brodsky, A. and Wang, X.S., "Decision-Guidance Management Systems (DGMS): Seamless Integration of Data Acquisition, Learning, Prediction, and Optimization", Proceedings of the 41st Hawaii International Conference on System Sciences, 2008.
13. Brodsky, A., Bhot, M.M., Chandrashekar, M., Egge, N.E., and Wang, X.S., "A Decisions Query Language (DQL): High-Level Abstraction for Mathematical Programming over Databases", Proceedings of the 35th SIGMOD International Conference on Management of Data, 2009.
14. Dumas, M., O'Sullivan, J., Heravizadeh, M., Edmond, D., and Hofstede, A., "Towards a Semantic Framework for Service Description", Proceedings of the IFIP TC2/WG2.6 Ninth Working Conference on Database Semantics: Semantic Issues in E-Commerce Systems, 2001.

15. Erl, T., "Service-Oriented Architecture (SOA): Concepts, Technology, and Design", Prentice Hall, 2005.
16. Erradi, A., Anand, S., and Kulkarni, N., "SOAF: An Architectural Framework for Service Definition and Realization", IEEE International Conference on Services Computing (SCC'06), 2006.
17. Harrington, J., "Relational Database Design and Implementation (Third Edition)", Morgan Kaufmann, 2009.
18. Holyfield, S., "Non-technical Guide to Technical Frameworks", JISC CETIS, 2005. <http://www.elearning.ac.uk/features/nontechguide1>.
19. Josuttis, N., "SOA in Practice: The Art of Distributed System Design", O'Reilly Media, 2007.
20. Nicholls, P., "Enterprise Architectures and the International e-Framework", e-framework Organization, 2009. [http://www.e-framework.org/Portals/9/docs/EAPaper\\_2009-07.pdf](http://www.e-framework.org/Portals/9/docs/EAPaper_2009-07.pdf)
21. Olivier, B., Roberts, T., and Blinco, K., "The e-Framework for Education and Research: An Overview", e-framework Organization, 2005. <http://www.e-framework.org/Portals/9/Resources/eframeworkrV1.pdf>.
22. Ort, Ed., "Service-Oriented Architecture and Web Services: Concepts, Technologies, and Tools", Sun Developer Network Technical Articles and Tips. <http://java.sun.com/developer/technicalArticles/WebServices/soa2/>
23. Papazoglou, M., and Heuvel, W., "Service Oriented Architectures: Approaches, Technologies, and Research Issues", The VLDB Journal, June, 2005.
24. Quartel, D., Steen, M., Pokraev S., and Sinderen, M., "COSMO: A Conceptual Framework for Service Modelling and Refinement", Volume 9, Numbers 2-3, 225-244, July. Journal of Information Systems Frontiers, 2007.
25. Ralph H. Sprague, Jr., "A Framework for the Development of Decision Support Systems", Volume 4, Number 4, 1-26, December. MIS Quarterly, 1980.
26. Stephen, B., et al., "Database Design: Know It All", Morgan Kaufmann, 2008.
27. Wilson, S., Blinco, K., and Rehak, D., "Service-Oriented Frameworks: Modelling the Infrastructure for the Next Generation of e-Learning Systems", JISC CETIS, 2004. [http://www.jisc.ac.uk/uploaded\\_documents/AlttilabServiceOrientedFrameworks.pdf](http://www.jisc.ac.uk/uploaded_documents/AlttilabServiceOrientedFrameworks.pdf).
28. Zhang, T., Ying, S., and Cao, S., and Jia, S., "A Modeling Framework for Service-Oriented Architecture", Proceedings of the Sixth International Conference on Quality Software (QSIC'06), 2006.