Decisions on Multivariate Time Series: Combining Domain Knowledge with Utility Maximization

Chun-Kit Ngan[1], Alexander Brodsky[2], and Jessica Lin[3]
George Mason University
cngan@gmu.edu[1], brodsky@gmu.edu[2], and jlin2@gmu.edu[3]

Abstract. We propose a general framework that combines the strengths of both domain-knowledge-based and formal-learning-based approaches for maximizing utility over multivariate time series. It includes a mathematical model and a learning algorithm for solving Expert Query Parametric Estimation problems. Using the framework, we conduct a preliminary experiment in the financial domain to demonstrate that our model and algorithm are more effective and produce results that are superior to the two approaches mentioned above.

Keywords. Decision optimization, regression learning, and time series.

Introduction

Automatic identification and detection of events in time series data is an important problem that has gained a lot of interest in the past decade. The timely detection of events can aid in the task of decision making and the determination of action plans. For example, in the financial domain, the identification of certain market conditions can provide investors valuable insight on the best investment opportunities. Existing approaches to identifying and detecting 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. For example, in the financial domain, experts have identified a set of financial indices that can be used to determine a bear market bottom or the "best buy" opportunity. 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. While these parameters may reflect some realities since they are set by the domain experts based on their past experiences, observations, intuitions, and domain knowledge, they are not always accurate. In addition, the parameters are static, but the problem that we deal with is often dynamic in nature.

An alternative is to utilize formal learning methods such as non-linear logit regression models [2, 3, 4]. The logit regression models can be used to predict the occurrence of an event (0 or 1) by learning parametric coefficients of explanatory

variables based on their historical data and a series of mathematical formulations and algorithms, e.g., 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 take advantage of domain knowledge. In addition, formal learning methods are computationally intensive. Fitting a nonlinear model is not a trivial task, especially if the parameter learning process involves multiple explanatory variables (i.e., high dimensionality).

To mitigate the shortcomings of the existing approaches, we propose a general framework 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. With the potentially large data size and multiple variables, classic branch-and-bound approaches have exponential complexity. To this end, we identify multivariate time series parametric estimation problems, in which the objective function is dependent on the time points from which the parameters are learned. We develop a new algorithm that 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. To demonstrate the effectiveness and the efficiency of our algorithm, we compare our method with the domain-knowledge-based approach and the logit regression model. As a proof of concept, we conduct an experiment in the financial domain, but note that our framework can be generalized to solve problems in different domains. We show that our algorithm is more effective and produces results that are superior to the two approaches mentioned above.

The rest of the paper is organized as follows. In Section 1 we formally introduce and define the Expert Query Parametric Estimation (EQPE) problem. We describe our domain-knowledge-inspired learning algorithm in Section 2. Section 3 shows the experimental evaluation on stock market data and concludes our work.

1. Mathematical Model of Expert Query Parametric Estimation Problems

Intuitively, an Expert Query Parametric Estimation (EQPE) problem 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 **T** is represented by the set of natural numbers: **T** = **N**, and we are given a vector of real parameters $(p_1, p_2, \ldots, p_n) \in \boldsymbol{R}^n$.

Definition 1. *Time Series*: A time series S is a function S: T → R, where T is the time domain, and R is the set of real numbers. In our example, the five parametric time series are SPD(t), CG(t), CCD(t), ISM(t), and NLCD(t).

Definition 2. *Parametric Constraint*: A parametric constraint $C(S_1(t), S_2(t), \ldots, S_k(t), p_1, p_2, \ldots, p_n)$ is a symbolic expression in terms of $S_1(t), S_2(t), \ldots, S_k(t), p_1, p_2, \ldots, p_n$, where $S_i(t)$ is a real value of the i^{th} time series S_i at the time t and p_j is the j^{th} decision parameter for $1 \le i \le k$ and $1 \le j \le n$. We assume a constraint C written in a language that has the truth-value interpretation I: $R^k \times R^n \rightarrow$ {True, False}, i.e., $I(C(S_1(t), S_2(t), \ldots, S_k(t), p_1, p_2, \ldots, p_n))$ = True if and only if the constraint C is satisfied at the

time point $t \in$ T and with the parameters $(p_1, p_2, \ldots, p_n) \in R^n$. In this paper, we focus on conjunctions of inequality constraints: $C(S_1(t), S_2(t), \ldots, S_k(t), p_1, p_2, \ldots, p_n) = \wedge_i (S_i(t)\ op\ p_j)$, where $op \in \{<, \leq, =, \geq, >, <>\}$. For example, $SPD(t) < p_1 \wedge CG(t) < p_2 \wedge CCD(t) < p_3 \wedge ISM(t) < p_4 \wedge NLCD(t) > p_5$ are the inequality constraints, where $p_1, p_2, \ldots, p_5$ are the decision parameters.

Definition 3. *Time Utility Function*: A time utility function U is a function U: T → R. In our example, U(t) is the percentage of earning under the assumption that the S&P 500 index (SP) fund is purchased at the time t and is sold at t_S, where SP(t) and SP(t_S) are the buy and sell value of the fund respectively.

Definition 4. *Objective Function*: Given a time utility function U: T → R and a parametric constraint C, an objective function O is a function O: $R^n \rightarrow R$, which maps a vector of parameters on R^n to a real value R, defined as follows: For $(p_1, p_2, \ldots, p_n) \in R^n$, $O(p_1, p_2, \ldots, p_n) \stackrel{def}{=} U(t)$, where U is the utility function, and $t \in$ T is the earliest time point that satisfies C, i.e., ***(1)*** $S_1(t)\ op_1\ p_1 \wedge S_2(t)\ op_2\ p_2 \wedge \ldots \wedge S_n(t)\ op_n\ p_n$ is satisfied, and ***(2)*** there does not exist $0 \leq t' < t$, such that $S_1(t')\ op_1\ p_1 \wedge S_2(t')\ op_2\ p_2 \wedge \ldots \wedge S_n(t')\ op_n\ p_n$ is satisfied.

Definition 5. *Expert Query Parametric Estimation (EQPE) Problem*: An EQPE problem is a tuple $<\dot{\boldsymbol{S}}, \dot{\boldsymbol{P}}, \boldsymbol{C}, \boldsymbol{U}>$, where $\dot{\boldsymbol{S}} = \{S_1, S_2, \ldots, S_k\}$ is a set of k time series, $\dot{\boldsymbol{P}} = \{p_1, p_2, \ldots, p_n\}$ is a set of n real-value parameters, $\boldsymbol{C}$ is a parametric constraint in $\dot{\boldsymbol{S}}$ and $\dot{\boldsymbol{P}}$, and $\boldsymbol{U}$ is a time utility function. Intuitively, a solution to an EQPE problem is an instantiation of values into parameters that maximize the objective.

Definition 6. *Expert Query Parametric Estimation (EQPE) Solution*: A solution to the EQPE problem $<\dot{\boldsymbol{S}}, \dot{\boldsymbol{P}}, \boldsymbol{C}, \boldsymbol{U}>$ is $argmax\ \boldsymbol{O}(p_1, p_2, \ldots, p_n)$, i.e., the estimated values of parameters, $p_1, p_2, \ldots, p_n$, that maximize $\boldsymbol{O}$, where $\boldsymbol{O}$ is the objective function corresponding to $\boldsymbol{U}$. In our example, the solution is argmax $\boldsymbol{O}(p_1, p_2, \ldots, p_5)$.

2. Checkpoint Algorithm for the Expert Query Parametric Estimation Problem

Before explaining the algorithm in detail, we first introduce a new concept, *Dominance*.

Definition 7. *Dominance* $\succ$: Given an EQPE problem $<\dot{\boldsymbol{S}}, \dot{\boldsymbol{P}}, \boldsymbol{C}, \boldsymbol{U}>$ and any two time points $t, t' \in$ 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)*** $\forall(p_1, p_2, \ldots, p_n) \in R^n$, $C(S_1(t), S_2(t), \ldots, S_k(t), p_1, p_2, \ldots, p_n) \rightarrow C(S_1(t'), S_2(t'), \ldots, S_k(t'), p_1, p_2, \ldots, p_n)$. 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 $\mathbb{C}$ - Given the conjunctions of inequality constraints, $S_1(t)\ op_1\ p_1 \wedge S_2(t)\ op_2\ p_2 \wedge \ldots \wedge S_k(t)\ op_k\ p_k$, $t' \succ t$ if and only if $S_1(t')\ op_1\ S_1(t) \wedge S_2(t')\ op_2\ S_2(t) \wedge \ldots \wedge S_k(t')\ op_k\ S_k(t)$.

Conceptually, we can search a particular set of parameters $\{p_1, p_2, \ldots, p_n\}$ which is at the earliest time point t that is not dominated by *any* t' such that the value of the objective function $\boldsymbol{O}$ 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, we propose the Checkpoint algorithm, which uses the KD-tree data structure

and searching algorithm [5, 6, 7] to evaluate whether a time point t is dominated based on the Claim $\mathbb{C}$ for checking the non-dominance. The pseudo code of the algorithm is:

Input: $<\dot{\boldsymbol{S}}, \dot{\boldsymbol{P}}, \boldsymbol{C}, \boldsymbol{U}>$

Output: $\boldsymbol{p}[1\ldots k]$ 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. MaxT is the time point that gives the maximal $\boldsymbol{U}$ so far, denoted by MaxU.

STEP 1: $T_{kd} := <S_1(0), S_2(0), \ldots, S_k(0)>$; MaxT := 0; MaxU := U(0).

STEP 2: Test if t is not dominated using the claim $\mathbb{C}$ and the T_{kd} range query.

STEP 3: If t is not dominated and U(t) > MaxU, then add $<S_1(t), S_2(t), \ldots, S_k(t)>$ to T_{kd}; MaxT := t; MaxU := U(t).

STEP 4: ***FOR*** i := 1 ***To*** k ***DO*** ($\boldsymbol{p}$[i] := S_i(MaxT))

STEP 5: ***RETURN*** $\boldsymbol{p}[1\ldots k]$

Apparently, the first time point is not dominated because there is no time point preceding it. $<S_1(0), S_2(0), \ldots, S_k(0)>$ can be added to T_{kd}. 0 and U(0) can be assigned to MaxT and MaxU respectively.

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 $C(S_1(t), S_2(t), S_3(t), p_1, p_2, p_3) = S_1(t) \geq p_1 \wedge S_2(t) \geq p_2 \wedge S_3(t) \leq p_3$. We also assume that the values of S_1, S_2, S_3, and U at the time point t_1, t_2, and t_3 are shown in Table 1. Using the Checkpoint algorithm step by step, we can search 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 $\boldsymbol{U}$ is maximal.

STEP 1: $T_{kd} := <S_1(t_1), S_2(t_1), S_3(t_1)>$; MaxT := t_1; MaxU := U(t_1).

STEP 2:

1. 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 $\mathbb{C}$.
2. 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 $\mathbb{C}$.

STEP 3: Add $<S_1(t_2), S_2(t_2), S_3(t_2)>$ to T_{kd} because t_2 is *not dominated* and U(t_2) > U(t_1). MaxT := t_2; MaxU := U(t_2).

STEP 4: $\boldsymbol{p}[1] := S_1(MaxT)$; $\boldsymbol{p}[2] := S_2(MaxT)$; $\boldsymbol{p}[3] := S_3(MaxT)$.

STEP 5: Return 25, 15, and 2.

The time complexity for the range search and insertion of a parameter vector in the T_{kd} tree is $O(k\log N)$ respectively. For N parameter vectors, the Checkpoint algorithm correctly computes the EQPE solution with the complexity of $O(kN\log N)$.

Table 1. Values of S_1, S_2, S_3, and U at the time point t_1, t_2, and t_3

Time	$\boldsymbol{S_1}$	$\boldsymbol{S_2}$	$\boldsymbol{S_3}$	$\boldsymbol{U}$
t_1	13	27	3	10
t_2	25	15	2	200
t_3	10	20	5	150

3. Stock Market Experimental Results and Conclusions

The optimal decision parameters and the maximal earning determined by the Checkpoint algorithm for the financial example are shown in Table 2. 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 3 respectively. Note that the Checkpoint algorithm considerably outperforms both the financial expert's criteria and the logit regression model.

To the best of our knowledge, this is the first paper that combines both domain expertise and the learning-based approach to solve EQPE problems. There are still many open research issues, including more expressive query languages to express an EQPE problem and efficient algorithms with low computational complexity to solve it.

Table 2. Optimal Decision Parameters and Maximum Earning (%) from the Learning Data Set[1]

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

Table 3. Investors' Earning of the S&P 500 Index Fund from the Test Data Set[2]

Decision Approach	Best Buy	S&P 500 Index	Earning%
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

References

[1] Stack, J.B., *Technical and Monetary Investment Analysis*. InvesTech Research, Vol 9 Issue 3 & 5, 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] Bentley, J.L., *Multidimensional Binary Search Trees in Database Applications*. IEEE Transactions on Software Engineering, Vol 5 Issue 04, p. 333-340, 1979.

[6] Bentley, J.L., *Multidimensional Binary Search Trees Used for Associative Searching*. Communications of the ACM, Vol 18 Issue 09, p. 509-517, 1975.

[7] Samet, H., *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, 2006.

[8] 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.

[9] 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.

[10] 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.

[1] The learning data set is from 06/01/1997 to 06/30/2005.

[2] 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.