Approximation of Stochastic Differential Equations by Additive Models Using Splines and Conic Programming

Pakize Taylan[a,b], Gerhard-Wilhelm Weber[b], Erik Kropat[c]

[a]Middle East Technical University, Institute of Applied Mathematics,
06531 Ankara, Turkey
[b]Dicle University, Department of Mathematics,
21280 Diyarbakır, Turkey
[c]University Erlangen-Nuremberg, Department of Mathematics,
91058 Erlangen, Germany

E-Mails: ptaylan@dicle.edu.tr, gweber@metu.edu.tr, kropat@am.uni-erlangen.de

Abstract Stochastic differential equations are widely used to model noise-affected phenomena in nature, technology and economy [12]. As these equations are usually hard to represent by a computer and hard to resolve we express them in simplified manner. We introduce an approximation by discretization and additive models based on splines. Then, we construct a penalized residual sum of squares (PRSS) for this model. We show when the related minimization program can be written as a Tikhonov regularization problem (ridge regression), and we treat it using continuous optimization techniques. In particular, we apply the elegant framework of conic quadratic programming. Convex optimization problems are very well-structured, resembling linear programs and permit the use of interior point methods [14].

Keywords : Stochastic Differential Equations, Regression, Splines, Tikhonov Regularization, Continuous Optimization.

1 Introduction

In the last decade, the development of high-throughput technologies has resulted in an accelerated generation of massive quantities of data. Such large data sets are often characterized by a high variation and are influenced by errors and uncertainty. At the same time, the data need to become well understood and they have to serve as the basis of future prediction with stochastic differential equations. Unfortunately, both the real situation and the practical requests are often hard to balance. In fact, related mathematical modeling faces with nondifferentiability and a high sensitivity with respect to slightest perturbations of the data. In our paper we address approximations of stochastic differential equations and formulate a penalized residual sum of squares problem based on an additive representation of coefficients. We discuss and elaborate the corresponding parameter estimation problem by means of

Tikhonov regularization and conic quadratic programming. Herewith, we offer an alternative view and approach to stochastic differential equations (SDEs).

2 Classical Additive Models

In many applied areas, regression models are of particular importance. Especially linear regression models are widely used, but they often fail in applications because of the nonlinear nature of the system under consideration. To characterize these effects, flexible statistical methods like *non-parametric regression* must be used [5]. However, if the number of independent variables is large in the models, many forms of nonparametric regression do not perform well. It is also difficult to interpret non-parametric regression depending on smoothing spline estimates. To overcome these difficulties, Stone [15] proposed *additive models.* These models estimate an additive approximation of the multivariate regression function. Here, the estimation of the individual terms explains how the dependent variable changes with the corresponding independent variables and we can examine the predictor effect separately in absence of interactions. We refer to [9, 10] for basic elements of the theory of additive models. Let us have N observations on a response (or dependent) variable Y, denoted by $y = (y_1, y_2, \ldots, y_N)^T$ measured at N design vectors $x_i = (x_{i1}, x_{i2}, \ldots, x_{im})^T$. The points x_i may be chosen in advance, or may themselves be measurements of random variables X_m $(j = 1, 2, \ldots, m)$, or both. The additive model is defined by

$$Y = \beta_0 + \sum_{j=1}^{m} f_j(X_j) + \varepsilon, \tag{1}$$

where the errors ε are independent of the factors X_j, $E(\varepsilon) = 0$ and $Var(\varepsilon) = \sigma^2$. Here, the functions f_j are arbitrary unknown, univariate functions. Mostly, they are considered to be splines, i.e., piecewise polynomial functions, because polynomials themselves have a too strong or early asymptotic to ±∞ and by this they are not satisfying for data fitting. We denote the estimates by $\widehat{f_j}$. The standard convention consists in assuming at X_j that $E(f_j(X_j)) = 0$, since otherwise there will be a free constant in each of the functions [9]; all such constants are summarized by the intercept (bias) β_0.

2.1 Estimation Equations for Additive Model

Additive models have a strong motivation as a useful data analytic tool. Friedman and Stuetzle [6] proposed an algorithm for the estimation of the functions in eq. 1 that is known as *backfitting* (or *Gauss-Seidel*) *algorithm.* Here, as an estimator for β_0 we use the mean of the response variable Y, i.e., $\widehat{\beta_0} = E(Y)$. This procedure depends on the partial residual against X_j, given by

$$r_j = Y - \beta_0 - \sum_{k \neq j} f_k(X_k), \tag{2}$$

and it consists of estimating each smooth function by holding all the other ones fixed [13]. Then, $E(r_j|X_j) = f_j(X_j)$ which minimizes $E(Y - \beta_0 - \sum_{j=1}^{m} f_j(X_j))^2$. In a framework of *cycling* from one to the next iteration, this means the following [7, 8]:

```
**initialization:** $\widehat{\beta}_0 = E(Y)$, $\widehat{f}_j^1(\cdot) \equiv 0$, $p = 0$
**cycle** $j = 1, \ldots, m,\ 1, \ldots,$
**iterate:** $p = p + 1$
for $j = 1$ to $m$ do:
    $r_j = Y - \widehat{\beta}_0 - \sum_{k \neq j}^{m} \widehat{f}_k(X_k),$
    $\widehat{f}_j^m(X_j) = E(r_j|X_j)$
**until** $RSS = E(Y - \widehat{\beta}_0 - \sum_{k=1}^{m} \widehat{f}_k(X_k))^2$ fails to decrease.
```

To prove the **convergence** of this procedure, Buja and Hastie [3] used the normal equations. By an arbitrary solution $\widetilde{\boldsymbol{f}}$ of that system a reduction of the problem to the solution of a corresponding homogeneous system can be made. This algorithm has been modified by Taylan and Weber [16]. Here, the l-th iteration in the ***modified*** *backfitting* or *Gauss-Seidel algorithm* includes an additional penalized curvature term.

3 Stochastic Differential Equations

3.1 Definition (Stochastic Differential Equations)

Many phenomena in nature, technology and economy are affected by noise and stochastic fluctuations. In order to describe such random dynamics, stochastic differential equations are widely used. Solutions of these equations are often diffusion processes and, hence, they are connected to the subject of partial differential equations. To solve these problems, we apply an *additive* approximation using spline functions as motivated in Section 2. A *stochastic differential equation*, equipped with an initial value, is given by

$$\begin{cases} \dot{X}(t) = a(X,t) + b(X,t)\delta_t \quad (t \in [0,\infty)), \\ X(0) = x_0, \end{cases} \tag{3}$$

where a is the deterministic part, $b\delta_t$ is the stochastic part, and δ_t denotes a generalized stochastic process [12]. An example of a generalized stochastic process is white noise. For a generalized stochastic processes, derivatives of any order can be defined. Suppose that W_t is a generalized version of a Wiener process which is used to model the motion of stock prices, which instantly responds to the numerous upcoming information. A Wiener process is a time continuous process with the property

$W_t \sim N(0,t)$ $(0 \leq t \leq T)$, usually it is differentiable almost nowhere. To obtain our approximate and then, smoothed model, we treat W_t as if it was differentiable (a first approach which is widespread in literature). Then, white noise δ_t is defined as $\delta_t = \dot{W}_t = dW_t/dt$ and a Wiener process can be obtained by smoothing the white noise. If we replace $\delta_t dt$ by dW_t in eq. 3, an (*Itô*) *stochastic differential equation* can be rewritten as

$$dX_t = a(X_t,t)dt + b(X_t,t)dW_t, \tag{4}$$

where $a(X_t,t)$ and $b(X_t,t)$ are drift and diffusion term, respectively, and X_t is a solution which we try to find based on the experimental data. As we do not know the distribution of X_t, we want to simulate its values. Therefore, we simulate a *discretized* version of the SDE.

3.2 Discretization of SDE

For a discretization of the stochastic differential equation (4) we consider the **Milstein scheme**. Then, an approximate for X_t is given by

$$\begin{aligned} X_{j+1} = X_j &+ a(X_j,t_j)(t_{j+1}-t_j) + b(X_j,t_j)(W_{j+1}-W_j) \\ &+ \frac{1}{2}(b'b)(X_j,t_j)\Big((W_{j+1}-W_j)^2 - (t_{j+1}-t_j)\Big) \end{aligned} \tag{5}$$

where the prime "$'$" denotes the derivative. Referring to the finitely many sample (data) points $(\overline{X}_j, \overline{t}_j)$ $(j = 0,1,\ldots,N)$ we get

$$\dot{\overline{X}}_j = a(\overline{X}_j,\overline{t}_j) + b(\overline{X}_j,\overline{t}_j)\frac{\Delta W_j}{\overline{h}_j} + \frac{1}{2}(b'b)(\overline{X}_j,\overline{t}_j)\left(\frac{(\Delta W_j)^2}{\overline{h}_j} - 1\right), \tag{6}$$

where the vector $\dot{\overline{X}}_j$ represents difference quotients based on the jth experimental data and on step lengths $\overline{h}_j := \overline{t}_{j+1} - \overline{t}_j$ between neighbouring sampling times:

$$\dot{\overline{X}}_j := \begin{cases} \frac{\overline{X}_{j+1}-\overline{X}_j}{\overline{h}_j} & \text{, if } j = 0,1,\ldots,N-1 \\ \frac{\overline{X}_N-\overline{X}_{N-1}}{\overline{h}_N} & \text{, if } j = N. \end{cases}$$

However, as the relations (6) include real data they cannot be expected to hold in an exact sense, but we satisfy them best in the *approximate* sense of least squares of errors. For the sake of convenience, we still write "=" instead of the approximation symbol "≈", and we shall study the least squares estimation in Subsection 3.3.

Since $W_t \sim N(0,t)$, the increments ΔW_j are independent on non-overlapping intervals and moreover, $Var(\Delta \overline{W}_j) = \sqrt{\Delta \overline{t}_j}$ (classical notation; here, $\Delta \overline{t}_j := \overline{h}_j$), hence, the increments having normal distribution can be simulated with the help of standard normal distributed random numbers Z_j. Herewith, we obtain a discrete model for a Wiener process:

$$\Delta \overline{W}_j = Z_j \sqrt{\Delta \overline{t}_j},\ Z_j \sim N(0,1). \tag{7}$$

If we use this value in our discretized equation, we obtain

$$\dot{\overline{X}}_j = a(\overline{X}_j, \overline{t}_j) + b(\overline{X}_j, \overline{t}_j)\frac{Z_j}{\sqrt{\overline{h}_j}} + \frac{1}{2}(b'b)(\overline{X}_j, \overline{t}_j)(Z_j^2 - 1). \tag{8}$$

For simplicity we write eq. 8 as

$$\dot{\overline{X}}_j = \overline{G}_j + \overline{H}_j c_j + \overline{F}_j d_j, \tag{9}$$

where $\overline{G}_j := a(\overline{X}_j, \overline{t}_j)$, $\overline{H}_j := b(\overline{X}_j, \overline{t}_j)$, $\overline{F}_j := (b'b)(\overline{X}_j, \overline{t}_j)$, $c_j := Z_j/\sqrt{\overline{h}_j}$ and $d_j := 1/2(Z_j^2 - 1)$.

3.3 Estimation of Parameters

In this section we turn to an estimation of parameters in the Milstein model. To determine the unknown values $\overline{G}_j$ and $\overline{H}_j$ of eq. 9, we consider the optimization problem

$$\min_{y} \sum_{j=1}^{N} \left\| \dot{\overline{X}}_j - (\overline{G}_j + \overline{H}_j c_j + \overline{F}_j d_j) \right\|_2^2, \tag{10}$$

where the vector y comprises all the parameters in the Milstein model. We know that data coming, e.g., from the stock market, have a high variation. Indeed, for example, investors may temporarily pull financial prices away from their long-term trend level. Over-reactions may occur so that excessive optimism can drive prices unduly high or excessive pessimism may drive prices unduly low, new theoretical and empirical arguments can have an effect on share prices to fall dramatically, even though, to this day, it is impossible to fix a definite cause. Indeed, a thorough search failed to detect any specific or unexpected development that might account for the crash and, many studies have shown a marked tendency for the stock market to trend over time periods of weeks or longer, sometimes the market tends to react irrationally to economic news, even if that news has no real effect on the technical value of securities itself, etc.. For this reason, we must use a parameter estimation method which will *diminish this high variation* and will give a *smoother approximation* to the data. **Splines** are more flexible and they allow us to avoid large oscillation observed for high degree polynomial approximation. We recall that these functions can be described as linear combinations of basis splines and approximate the data $(\dot{\overline{X}}_j, \overline{t}_j)$ smoothly. Therefore, we approximate each function underlying the values

$\overline{G}_j = a(\overline{X}_j, \overline{t}_j)$, $\overline{H}_j c_j = b(\overline{X}_j, \overline{t}_j)c_j$ and $\overline{F}_j d_j = b'b(\overline{X}_j, \overline{t}_j)d_j$ in an *additive* way established on basis splines. This treatment is very useful for the stability of the model in the presence of the many and highly varying data. We use basis splines for each function characterized by a separation of variables (coordinates) in eq. 9. By this we obtain

$$\begin{aligned}
\overline{G}_j &= a(\overline{X}_j, \overline{t}_j) &&= \alpha_0 + \sum_{p=1}^{2} f_p(\overline{U}_{j,p}) = \alpha_0 + \sum_{p=1}^{2} \sum_{l=1}^{d_p^G} \alpha_p^l \, B_p^l(\overline{U}_{j,p}), \\
\overline{H}_j c_j = b(\overline{X}_j, \overline{t}_j)c_j &&&= \beta_0 + \sum_{r=1}^{2} g_r(\overline{U}_{j,r}) = \beta_0 + \sum_{r=1}^{2} \sum_{m=1}^{d_r^H} \beta_r^m \, C_r^m(\overline{U}_{j,r}), \qquad (11) \\
\overline{F}_j d_j = b'b(\overline{X}_j, \overline{t}_j)d_j &&&= \varphi_0 + \sum_{s=1}^{2} h_s(\overline{U}_{j,s}) = \varphi_0 + \sum_{s=1}^{2} \sum_{n=1}^{d_s^F} \varphi_s^n \, D_s^n(\overline{U}_{j,s}),
\end{aligned}$$

where $\overline{U}_j = (\overline{U}_{j,1}, \overline{U}_{j,2}) := (\overline{X}_j, \overline{t}_j)$. Here, if we denote the kth order base spline by $B_{\eta,k}$, a polynomial of degree $k-1$, with knots, say x_η, then a great benefit of using the base splines is provided by the following recursive algorithm [4]:

$$\begin{aligned}
B_{\eta,1}(x) &= \begin{cases} 1 & , \text{ if } x_\eta \leq x < x_{\eta+1} \\ 0 & , \text{ otherwise,} \end{cases} \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (12) \\
B_{\eta,k}(x) &= \frac{x - x_\eta}{x_{\eta+k-1} - x_\eta} B_{\eta,k-1}(x) + \frac{x_{\eta+k} - x}{x_{\eta+k} - x_{\eta+1}} B_{\eta+1,k-1}(x).
\end{aligned}$$

3.4 The Penalized Residual Sum of Squares Problem for SDE

We represent the *penalized residual sum of squares problem* for SDE in the following form:

$$\begin{aligned}
PRSS(\theta, f, g, h) :=& \sum_{j=1}^{N} \Big\{ \dot{\overline{X}}_j - (\overline{G}_j + \overline{H}_j c_j + \overline{F}_j d_j) \Big\}^2 + \sum_{p=1}^{2} \lambda_p \int \left[f_p''(U_p) \right]^2 dU_p \\
& + \sum_{r=1}^{2} \mu_r \int \left[g_r''(U_r) \right]^2 dU_r + \sum_{s=1}^{2} \varphi_s \int \left[h_s''(U_s) \right]^2 dU_s. \qquad (13)
\end{aligned}$$

Here, for convenience, the integral symbol "$\int$" is used as a dummy in the sense of $\int\limits_{[a_\kappa, b_\kappa]}$, where $[a_\kappa, b_\kappa]$ $(\kappa = p, r, s)$ are appropriately large intervals where the integration takes place, respectively. Furthermore, $\lambda_p, \mu_r, \varphi_s \geq 0$ are *smoothing* (or *penalty*) *parameters*, they represent a tradeoff between first and second term. Large values of $\lambda_p, \mu_r, \varphi_s \geq 0$ yield smoother curves, smaller values result in more fluctuation. If we use an additive form based on the basis splines for each function, then PRSS will become

$$\begin{aligned}
&\sum_{j=1}^{N}\Big\{\dot{\overline{X}}_j - (\overline{G}_j + \overline{H}_j c_j + \overline{F}_j d_j)\Big\}^2 \qquad (14)\\
&= \sum_{j=1}^{N}\Big\{\dot{\overline{X}}_j - \Big(\alpha_0 + \sum_{p=1}^{2}\sum_{l=1}^{d_p^G}\alpha_p^l\, B_p^l(\overline{U}_{j,p}) + \beta_0 + \sum_{r=1}^{2}\sum_{m=1}^{d_r^H}\beta_r^m\, C_r^m(\overline{U}_{j,r})\\
&\qquad + \varphi_0 + \sum_{s=1}^{2}\sum_{n=1}^{d_s^F}\varphi_s^n\, D_s^n(\overline{U}_{j,s})\Big)\Big\}^2.
\end{aligned}$$

Now, for simplicity, we introduce the following matrix notation:

$$\begin{aligned}
\overline{G}_j + \overline{H}_j c_j + \overline{F}_j d_j = \alpha_0 &+ \sum_{p=1}^{2}\sum_{l=1}^{d_p^G}\alpha_p^l\, B_p^l(\overline{U}_{j,p}) + \beta_0 + \sum_{r=1}^{2}\sum_{m=1}^{d_r^H}\beta_r^m\, C_r^m(\overline{U}_{j,r})\\
&+ \varphi_0 + \sum_{s=1}^{2}\sum_{n=1}^{d_s^F}\varphi_s^n\, D_s^n(\overline{U}_{j,s}) = \overline{A}_j\theta, \qquad (15)
\end{aligned}$$

where

$$\begin{aligned}
\overline{A}_j &= \big(B_j\ C_j\ D_j\big),\ B_j = \big(1\ B_j^1\ B_j^2\big),\ C_j = \big(1\ C_j^1\ C_j^2\big),\ D_j = \big(1\ D_j^1\ D_j^2\big)\\
B_j^p &= \big(B_p^1(\overline{U}_{j,p}),\ B_p^2(\overline{U}_{j,p}),\ldots,\ B_p^{d_p^G}(\overline{U}_{j,p})\big)\ (p=1,2),\\
C_j^r &= \big(C_r^1(\overline{U}_{j,r}),\ C_r^2(\overline{U}_{j,r}),\ldots,\ C_r^{d_r^H}(\overline{U}_{j,r})\big)\ (r=1,2),\\
D_j^s &= \big(D_s^1(\overline{U}_{j,s}),\ D_s^2(\overline{U}_{j,s}),\ldots,\ D_s^{d_s^F}(\overline{U}_{j,s})\big)\ (s=1,2)
\end{aligned}$$

and

$$\begin{aligned}
\theta &= \big(\alpha^T, \beta^T, \varphi^T\big)^T,\ \alpha = \big(\alpha_0, \alpha_1^T, \alpha_2^T\big)^T,\ \beta = \big(\beta_0, \beta_1^T, \beta_2^T\big)^T,\ \varphi = \big(\varphi_0, \varphi_1^T, \varphi_2^T\big)^T,\\
\alpha_p &= \big(\alpha_p^1, \alpha_p^2, \ldots, \alpha_p^{d_p^G}\big)^T\ (p=1,2),\\
\beta_r &= \big(\beta_r^1, \beta_r^2, \ldots, \beta_r^{d_r^H}\big)^T\ (r=1,2),\\
\varphi_s &= \big(\varphi_s^1, \varphi_s^2, \ldots, \varphi_s^{d_s^F}\big)^T\ (s=1,2).
\end{aligned}$$

Now, the residual sum of squares can be represented as the squared length of the difference between $\dot{\overline{X}}_j$ and $\overline{A}\theta$, where $\overline{A}$ is the matrix which contains the row vectors $\overline{A}_j$, and $\dot{\overline{X}}$ is the vector of difference quotients standing for the change rates of the experimental data:

$$\sum_{j=1}^{N}\{\dot{\overline{X}}_j - \overline{A}_j\theta\}^2 = \big\|\dot{\overline{X}} - \overline{A}\theta\big\|_2^2. \qquad (16)$$

Here, we have used $\overline{A} = \left(\overline{A}_1^T, \overline{A}_2^T, \ldots, \overline{A}_N^T\right)^T$, $\dot{\overline{X}} = \left(\dot{\overline{X}}_1, \dot{\overline{X}}_2, \ldots, \dot{\overline{X}}_N\right)^T$. Indeed, we get a discretized form of each integration term in the following way

$$\int_a^b \left[f_p''(U_p)\right]^2 dU_p \cong \sum_{j=1}^{N-1} \left[f_p''(U_{j,p})\right]^2 (U_{j+1,p} - U_{j,p}) = \sum_{j=1}^{N-1} \left[\sum_{l=1}^{d_p^G} \alpha_p^l B_p^{l''}(U_{j,p}) u_j\right]^2. \quad (17)$$

Using Riemann sums, we can discretize and represent each integration by the squared length of a vector, namely,

$$\begin{aligned} \int_a^b \left[f_p''(U_p)\right]^2 dU_p &\cong \sum_{j=1}^{N-1} \left[B_j^{p''} u_j \alpha_p\right]^2 = \left\|\overline{A}_p^B \alpha_p\right\|_2^2 \ (p = 1, 2), \\ \int_a^b \left[g_r''(U_r)\right]^2 dU_r &\cong \sum_{j=1}^{N-1} \left[C_j^{r''} \nu_j \beta_r\right]^2 = \left\|\overline{A}_r^C \beta_r\right\|_2^2 \ (r = 1, 2), \\ \int_a^b \left[h_s''(U_s)\right]^2 dU_s &\cong \sum_{j=1}^{N-1} \left[D_j^{s''} w_j \varphi_s\right]^2 = \left\|\overline{A}_s^D \varphi_s\right\|_2^2 \ (s = 1, 2). \end{aligned} \quad (18)$$

Here,

$$\begin{aligned} \overline{A}_p^B &:= \left(B_1^{p''T} u_1, B_2^{p''T} u_2, \ldots, B_{N-1}^{p''T} u_{N-1}\right)^T, \ u_j := \sqrt{U_{j+1,p} - U_{j,p}}, \\ \overline{A}_r^C &:= \left(C_1^{r''T} \nu_1, C_2^{r''T} \nu_2, \ldots, C_{N-1}^{r''T} \nu_{N-1}\right)^T, \ \nu_j := \sqrt{U_{j+1,r} - U_{j,r}}, \\ \overline{A}_s^D &:= \left(D_1^{s''T} w_1, D_2^{s''T} w_2, \ldots, D_{N-1}^{s''T} w_{N-1}\right)^T, \ w_j := \sqrt{U_{j+1,s} - U_{j,s}} \end{aligned}$$

for $j = 1, 2, \ldots, N-1$. Using this discretized form in eq. 15, PRSS looks as follows:

$$PRSS(\theta, f, g, h) = \left\|\dot{\overline{X}} - \overline{A}\theta\right\|_2^2 + \sum_{p=1}^{2} \lambda_p \left\|\overline{A}_p^B \alpha_p\right\|_2^2 + \sum_{r=1}^{2} \mu_r \left\|\overline{A}_r^C \beta_r\right\|_2^2 + \sum_{s=1}^{2} \varphi_s \left\|\overline{A}_s^D \varphi_s\right\|_2^2. \quad (19)$$

But, rather than a singleton, there is a finite sequence of the *tradeoff* or *penalty* parameters $\lambda = (\lambda_1, \lambda_2, \mu_1, \mu_2, \varphi_1, \varphi_2)^T$ such that this equation is not yet a *Tikhonov regularization problem* with a single such parameter. For this reason, let us make a uniform penalization by taking the same value $\lambda_p = \mu_r = \varphi_s = \lambda = \delta^2$ for each term. Then, our approximation of PRSS can be rearranged as

$$PRSS(\theta, f, g, h) = \left\|\dot{\overline{X}} - \overline{A}\theta\right\|_2^2 + \delta^2 \left\|\overline{L}\theta\right\|_2^2, \quad (20)$$

with the $6(N-1) \times m$ matrix

$$\overline{L} := \begin{pmatrix} 0 & \overline{A}_1^B & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \overline{A}_2^B & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \overline{A}_1^C & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \overline{A}_2^C & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \overline{A}_1^D & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \overline{A}_2^D \end{pmatrix}.$$

Herewith, based on the basis splines, we have identified the minimization of PRSS for some stochastic differential equation as a *Tikhonov regularization problem*[1] with penalty parameter $\lambda = \delta^2$:

$$\min_{m} \left\|Gm - d\right\|_2^2 + \delta^2 \left\|Lm\right\|_2^2. \tag{21}$$

This regularization method is also known as *ridge regression*; it is very helpful for problems whose solution does not exist, or which is not unique or not stable under perturbations of the data. MATLAB Regularization Toolbox can be used for solution.

4 Alternative Solution for Tikhonov Regularization Problem with Conic Quadratic Programming

4.1 Construction of the Conic Quadratic Programming Problem

We just mentioned that we can solve a Tikhonov regularization problem with MATLAB Regularization Toolbox. In addition, our problem can also be treated by *continuous optimization* techniques which we suppose to become an important complementary technology and alternative to the concept of Tikhonov regularization. In particular, we apply the elegant framework of *conic quadratic programming* (CQP). For this, we reformulate our Tikhonov regularization as the following optimization problem:

$$\min_{\theta} \left\|\overline{A}\theta - \dot{\overline{X}}\right\|_2^2, \tag{22}$$
$$\text{where } \left\|\overline{L}\theta\right\|_2^2 \leq M.$$

This problem requires an appropriate, learning based choice of a bound M. In addition, the objective function in (22) is not linear but quadratic. When the original objective function is moved to the list of constraints, we can formulate an equivalent problem as follows:

$$\begin{aligned} &\min_{t,\theta} t && (23) \\ &\text{subject to } \left\|\overline{A}\theta - \dot{\overline{X}}\right\|_2^2 \leq t^2,\ t \geq 0, \\ &\qquad\qquad\ \ \left\|L\theta\right\|_2^2 \leq M, \end{aligned}$$

or

$$\begin{aligned} &\min_{t,\theta} t && (24) \\ &\text{subject to } \left\|\overline{A}\theta - \dot{\overline{X}}\right\|_2 \leq t, \\ &\qquad\qquad\ \ \left\|L\theta\right\|_2 \leq \sqrt{M}. \end{aligned}$$

Then, if we consider the form of a conic quadratic optimization problem [13]

$$\min_x c^T x, \quad \text{subject to } \left\|D_i x - d_i\right\| \leq p_i^T x - q_i \ \ (i = 1, 2, \ldots, k), \tag{25}$$

we can see that our optimization problem for SDE is a conic quadratic program with

$$\begin{aligned} &c = \left(1\ 0_m^T\right)^T,\ x = \left(t\ \theta^T\right)^T,\ D_1 = \left(0_N, \overline{A}\right),\ d_1 = \dot{\overline{X}},\ p_1 = (1, 0, \ldots, 0)^T,\ q_1 = 0, \\ &D_2 = \left(0_{6(N-1)}, L\right),\ d_2 = 0,\ p_1 = 0_{m+1}^T,\ q_1 = -\sqrt{M}, \\ &m = \sum_{p=1}^{2} d_p^G + \sum_{r=1}^{2} d_r^H + \sum_{s=1}^{2} d_p^F + 3. \end{aligned}$$

In order to state the optimality conditions, we firstly reformulate our problem as

$$\begin{aligned} &\min_{t,\theta} t && (26) \\ &\text{such that } \chi := \begin{pmatrix} 0_N & \overline{A} \\ 1 & 0_m^T \end{pmatrix} \begin{pmatrix} t \\ \theta \end{pmatrix} + \begin{pmatrix} -\dot{\overline{X}} \\ 0 \end{pmatrix}, \\ &\qquad\qquad\ \ \eta := \begin{pmatrix} 0_{6(N-1)} & L \\ 0 & 0_m^T \end{pmatrix} \begin{pmatrix} t \\ \theta \end{pmatrix} + \begin{pmatrix} 0_{6(N-1)} \\ \sqrt{M} \end{pmatrix}. && (27) \end{aligned}$$

Here, χ and η belong to L^{N+1} and $L^{6(N-1)+1}$, where L^{N+1} and $L^{6(N-1)+1}$ are the $N+1$ and $6(N-1)+1$ *dimensional ice-cream* (*or second-order Lorentz*) *cones*, defined by

$$L^\nu := \left\{ x = (x_1, x_2, \ldots, x_\nu)^T \in \mathbb{R}^\nu \,\middle|\, x_\kappa \geq \sqrt{x_1^2 + x_2^2 + \ldots + x_{\nu-1}^2} \right\} \ (\nu \geq 2).$$

Then, we can also write the *dual problem* to the latter problem as

$$\begin{aligned} &\max\ (\dot{\overline{X}}^T, 0)\kappa_1 + (0_{6(N-1)}^T, -\sqrt{M})\kappa_2 \\ &\text{such that } \begin{pmatrix} 0_N^T & 1 \\ \overline{A}^T & 0_m \end{pmatrix} \kappa_1 + \begin{pmatrix} 0_{6(N-1)}^T & 0 \\ L^T & 0_m \end{pmatrix} \kappa_2 = \begin{pmatrix} 1 \\ 0_m \end{pmatrix}, && (28) \\ &\qquad\qquad\ \ \kappa_1 \in L^{N+1},\ \kappa_2 \in L^{6(N-1)+1}. \end{aligned}$$

Moreover, $(t, \theta, \chi, \eta, \kappa_1, \kappa_2)$ is the primal-dual optimal solution if the following constraints are provided in the corresponding ice-cream (second-order Lorentz) cones:

$$\begin{aligned}
&\chi := \begin{pmatrix} 0_N & \overline{A} \\ 1 & 0_m^T \end{pmatrix} \begin{pmatrix} t \\ \theta \end{pmatrix} + \begin{pmatrix} -\dot{\overline{X}} \\ 0 \end{pmatrix}, \\
&\eta := \begin{pmatrix} 0_{6(N-1)} & L \\ 0 & 0_m^T \end{pmatrix} \begin{pmatrix} t \\ \theta \end{pmatrix} + \begin{pmatrix} 0_{6(N-1)} \\ \sqrt{M} \end{pmatrix}, \\
&\begin{pmatrix} 0_N^T & 1 \\ \overline{A}^T & 0_m \end{pmatrix} \kappa_1 + \begin{pmatrix} 0_{6(N-1)}^T & 0 \\ L^T & 0_m \end{pmatrix} \kappa_2 = \begin{pmatrix} 1 \\ 0_m \end{pmatrix}, \\
&\kappa_1^T \chi = 0, \ \kappa_2^T \eta = 0, \\
&\kappa_1 \in L^{N+1}, \ \kappa_2 \in L^{6(N-1)+1}, \\
&\chi \in L^{N+1}, \ \eta \in L^{6(N-1)+1}.
\end{aligned} \tag{29}$$

4.2 On Solution Methods for Conic Quadratic Programming

Interior point methods (IPMs), firstly introduced by *Karmarkar* [11], can be applied for solving "well-structured" convex problems like conic quadratic problems. These methods classically base on the interior points of the feasible set of the optimization problem; this set is assumed to be closed and convex. Then, an *interior penalty function* (barrier) is chosen, well defined (and smooth and strongly convex) in the interior of the feasible set. This function is "blowing up" as a sequence from the interior approaches a boundary point of the feasible set [14].

The *canonical barrier function* for second-order (Lorentz) cones L^ν is defined by $L_\nu(x) := -\ln(x_\nu^2 - x_1^2 - \ldots - x_{\nu-1}^2) = \ln(x^T J_\nu x)$, where $J_\nu = \begin{pmatrix} -I_{\nu-1} & 0 \\ 0 & 1 \end{pmatrix}$. The parameter of this barrier is $\alpha(L_\nu) = 2$. These algorithms have the advantage of employing the structure of the problem, of allowing better complexity bounds and exhibiting a much better practical performance.

5 Conclusion

This paper gave a new contribution to problems related to SDE using regression by an additive model and letting modern methods of inverse problems and continuous optimization, especially, CQP, become accessible and usable. Herewith, a bridge has been offered between statistical learning and data mining on the one hand, and the powerful tools prepared for well-structured convex optimization problems [2] on the other hand. We hope that future research, theoretical and applied achievements on this fruitful interface will be stimulated by our paper. Indeed, applications of our method on real-word data from areas of science, finance and technology may be expected, where our contribution can be utilized.

References

[1] Aster A., Borchers B., Thurber C. (2004) Parameter Estimation and Inverse Problems. Academic Press.

[2] Boyd S., Vandenberghe L. (2004) Convex Optimization. Cambridge University Press.

[3] Buja A., Hastie T., Tibshirani R. (1989) Linear smoothers and additive models. The Ann. Stat. 17, 2, pp. 453-510.

[4] De Boor C. (2001) Practical Guide to Splines. Springer Verlag.

[5] Fox J. (2002) Nonparametric regression, Appendix to an R and S-Plus Compañion to Applied Regression. Sage Publications.

[6] Friedman J.H., Stuetzle W. (1981) Projection pursuit regression. J. Amer. Statist. Assoc. 76, pp. 817-823.

[7] Hastie T., Tibshirani R. (1986) Generalized additive models. Statist. Science 1, 3, pp. 297-310.

[8] Hastie T., Tibshirani, R. (1987) Generalized additive models: some applications. J. Amer. Statist. Assoc. 82, 398, pp. 371-386.

[9] Hastie T., Tibshirani R. (1990) Generalized Additive Models. Chapman and Hall, New York.

[10] Hastie T., Tibshirani R., Friedman J.H. (2001) The Elements of Statistical Learning. Springer Verlag, New York.

[11] Karmarkar N. (1984) A New Polynomial Time Algorithm for Linear Programming. Combinatorica 4, 4, pp. 373-395.

[12] Kloeden P.E., Platen E., Schurz H. (1994) Numerical Solution of SDE Through Computer Experiments, Springer Verlag, New York.

[13] Nemirovski A. (2002) Lectures on modern convex optimization. Israel Institute of Technology, http://iew3.technion.ac.il/Labs/Opt/opt/LN/Final.pdf.

[14] Nesterov Y.E., Nemirovskii A.S. (1993) Interior Point Methods in Convex Programming. SIAM.

[15] Stone C.J. (1985) Additive regression and other nonparametric models. The Annals of Statistics 13, 2, pp. 689-705.

[16] Taylan P., Weber G.-W., Beck A. New approaches to regression by generalized additive models and continuous optimization for modern applications in finance, science and technology. To appear in the special issue of Optimization in honour of Professor Alexander Rubinov.