Taylor & Francis
Taylor & Francis Group

Optimization

A Journal of Mathematical Programming and Operations Research

ISSN: 0233-1934 (Print) 1029-4945 (Online) Journal homepage: http://www.tandfonline.com/loi/gopt20

# New approaches to regression by generalized additive models and continuous optimization for modern applications in finance, science and technology

P. Taylan , G.-W. Weber & A. Beck

**To cite this article:** P. Taylan , G.-W. Weber & A. Beck (2007) New approaches to regression by generalized additive models and continuous optimization for modern applications in finance, science and technology, Optimization, 56:5-6, 675-698, DOI: 10.1080/02331930701618740

**To link to this article:** http://dx.doi.org/10.1080/02331930701618740

*Optimization*,
Vol. 56, Nos. 5–6, October–December 2007, 675–698

Taylor & Francis
Taylor & Francis Group

# New approaches to regression by generalized additive models and continuous optimization for modern applications in finance, science and technology¶

P. TAYLAN†‡, G.-W. WEBER*† and A. BECK§

†Institute of Applied Mathematics, Middle East Technical University,
06531 Ankara, Turkey
‡Department of Mathematics, Dicle University, 21280 Diyarbakır, Turkey
§Faculty of Industrial Engineering and Management,
Israel Institute of Technology, Technion, Haifa, Israel

(*Received 14 September 2006; in final form 2 April 2007*)

Generalized additive models belong to modern techniques from statistical learning, and are applicable in many areas of prediction, e.g. in financial mathematics, computational biology, medicine, chemistry and environmental protection. In these models, the expectation of response is linked to the predictors via a link function. These models are fitted through local scoring algorithm using a scatterplot smoother as building blocks proposed by *Hastie and Tibshirani* (1987). In this article, we first give a short introduction and review. Then, we present a mathematical modeling by splines based on a new clustering approach for the $x$, their density, and the variation of output $y$. We contribute to regression with generalized additive models by bounding (penalizing) second-order terms (curvature) of the splines, leading to a more robust approximation. Previously, in [23], we proposed a refining modification and investigation of the *backfitting algorithm*, applied to additive models. Then, because of drawbacks of the modified backfitting algorithm, we solve this problem using *continuous optimization* techniques, which will become an important complementary technology and alternative to the concept of modified backfitting algorithm. In particular, we model and treat the constrained residual sum of squares by the elegant framework of *conic quadratic programming*.

*Keywords:* Regression; Generalized additive model; Statistical learning; Clustering; Separation of variables; Density; Variation; Curvature; Backfitting (Gauss–Seidel) algorithm; Penalty methods; Classification; Continuous optimization; Conic quadratic programming; Financial mathematics

*Mathematics Subject Classifications 2000:* 41A15; 62J99; 65D10; 90C25; 90C90

*Corresponding author. Email: gweber@metu.edu.tr
¶This study was carried out as part of Pakize Taylan's postdoc at METU in the program DOSAP.

*Optimization*
ISSN 0233-1934 print/ISSN 1029-4945 online 
http://www.tandf.co.uk/journals
DOI: 10.1080/02331930701618740

## 1. Introduction

### 1.1. *Learning and models*

In the last decade, learning from data has become very important in every field of science. Modern learning challenges can be found in the fields of computational biology and medicine, and in the financial sector. Estimation and prediction problems frequently arise in learning. For those problems, regression theory is used, mainly based on the idea of least squares or maximum likelihood estimation, but also classification theory is applied.

In statistical learning, we begin with deterministic models and then we turn to the more general case of stochastic models where uncertainties, noise or measurement errors are taken into account. For closer information we refer to the book Hastie *et al*. [12].

In classical models, the approach to explain the recorded data $y$ consists of one unknown function only; the introduction of *additive models* [6] allowed an "approximation" with sum of functions. These functions have separated input variables. Our contribution is the introduction of a new approach that figures out *clusters* of input data points $x$ (or entire data points $(x, y)$), and assigning in the additive *approximation* for each cluster's own function. Thus, each individual function additively contributes to the understanding and learning from the measured data. These functions are defined over domains such as intervals or higher dimensional intervals, and depend on the cluster knots; mostly they are assumed to be splines. We introduce an *index* useful for deciding the spline degrees by *density* and *variation* properties of the corresponding data in $x$ and $y$ components, respectively [23]. In a further step of refinement, aspects of stability and complexity of the problem are implied by keeping the curvatures of the model functions under some chosen bounds. The corresponding constrained least squares problem can be treated for example as a *penalized* unconstrained minimization problem. In this article, for the generalized (penalized) problem, we specify (*modify*) the *backfitting algorithm* which was investigated and applied for additive models. Our new investigation of *generalized additive models* is introduced in a probabilistic framework based on [12] and closer presented in the deterministic case.

This article contributes to both the $m$-dimensional case of input data separated by the model functions and, as our new alternative, to 1- (or higher) dimensional input data partitioned into clusters. Dimensional generalizations of the second interpretation and a combination of both interpretations are possible and indicated. Applicability for data *classification* is noted. We point out advantages and disadvantages of the concept of modified backfitting algorithm.

By all of this, we present and discuss the modified backfitting algorithm related to penalized residual sum of squares. We overcome drawbacks in convergence, which are due to the regular adaption of the penalty parameters by proposing an alternative solution method which uses conic quadratic programming. This class of convex optimization problems arises in different fields and it is well known that efficient polynomial time algorithms (e.g. interior point methods) are available for solving these problems. We treat our problem by theory and methods coming from this new interpretation and as a complement and alternative to our modified backfitting algorithm.

### 1.2. *A motivation of regression*

One of the motivations of this research has been the approximation of financial data points $(y, x)$ coming, e.g. from the stock market, credit rating, economic factors or company properties. For example, to estimate the probability of a default for a *particular* credit during the next year, or of a default for a credit *randomly chosen* from a particular rating class over the next year, we can use the input data of credit rating, economic factors or company properties. The estimation of probability of a default has considerable importance in risk management applications where default risk usually is referred to as credit risk. Sometimes, financial markets may face several events of insolvency and crises. These events have attracted considerable attention of both academics and regulators. For this reason, *Basel II* (Committee on Banking Supervision) propose a revision to the international capital accord that suggests a more prominent role for internal credit risk assessments based on the determination of the probability of default of a borrower or group of borrowers [2].

For the above reason, there are different approaches for estimating the probability of a default. *Regression models* [14] (binary choice) are one of them, but these models must estimate defaults as accurately as possible. Probability of a default of a particular credit during the next year or a default of a credit randomly chosen from a particular rating class over the next year can be estimated by the regression model which we explain in the following. For example, assume that the dependent variable $Y$ (observed data) with $Y=1$ (“default”) or $Y=0$ (“no default”) satisfies [14]

$$Y = F(X) + \varepsilon, \tag{1.1}$$

where $X$ is a vector of independent variable(s) (input) such as credit rating, economic factors, company properties, and the noise term $\varepsilon$ has the expected value 0. Taking the expectation in equation (1.1), we obtain the default probability $P$ as

$$P = E[F(X) + \varepsilon] = F(X). \tag{1.2}$$

Hence, we can obtain an estimate for the default probability of a corporate bond via regression models. Also, this estimation can be done via linear regression. If linear regression is used based on the approach

$$Y = \alpha + \beta^{\mathrm{T}} X + \varepsilon, \tag{1.3}$$

an estimate for the default probability of a corparate bond can be obtained via [14]:

$$P = \alpha + \beta^{\mathrm{T}} X. \tag{1.4}$$

Here, $\alpha$ and $\beta$ are unknown parameters that can be estimated via statistical learning [12], especially linear regression methods or maximum likelihood estimation. In many important cases, these just mean least squares estimation.

For introductory and closer information about these methods from the viewpoints of statistical learning or the stheory of inverse problems, we refer to the books of Hastie *et al.* [12] and Aster *et al.* [3], respectively. A new application in the modeling and prediction of gene-environment networks can be found in [25].

### 1.3. *Additive models*

**1.3.1. Classical additive models.** Regression models, especially linear ones, are very important in many applied areas. However, the traditional linear models often fail in real life, since many effects are generally *nonlinear*. To characterize these effects, flexible statistical methods like *nonparametric regression* must be used [8]. 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 nonparametric regression depending on smoothing spline estimates. To overcome these difficulties, Stone [22] 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. We refer to [10] for basic elements of the theory of additive models.

If we have data consisting of $N$ realizations of random variable $Y$ at $m$ design values, then the additive model takes the from

$$E(Y_i|x_{i1},\ldots,x_{im}) = \beta_0 + \sum_{j=1}^{m} f_j(x_{ij}). \tag{1.5}$$

Here, the functions $f_j$ are mostly considered to be splines, i.e. piecewise polynomial, since, e.g. polynomials themselves are too strong or early asymptotic to $\pm\infty$ and by this they are not satisfying for data fitting. In our first approach to estimate the $f_j$ we use a procedure of successive smoothing on single coordinates, called backfitting algorithm (see subsection 2.5). After a careful discussion of its pros and cons, we do the estimation by conic quadratic programming (see subsection 3.3). We denote estimates by $\hat{f}_j$. By all the $x_{ij}$, we represent input data values; later on, in the backfitting algorithm, these values also serve as the knots of the interpolating (or smoothing) splines which appear there. The estimation of the $f_j$ is first done by an algorithm which performs a stepwise smoothing with respect to suitably chosen spline classes and to the points $x_{ij}$ and difference values between an average $y_i$ and a sum of functions evaluated at the knots $x_{ij}$, rather than with given *a priori* output knots. Materially regarded, the $x_{ij}$ have a twofold interpretation in our article, which we will carefully explain. Indeed, there is the understanding of $x_{ij}$ as the $j$-th component of the $i$-th input variable (classical separation of variable approach), and we offer a new understanding as the $i$-th point of the $j$-th cluster ($I_j$) of input data. This article holds true for *both* of these interpretations. Let us by $y_{ij}$ denote the output values corresponding to the inputs $x_{ij}$. Aggregating over these values with respect to $j$, delivering $y_i := \sum_{j=1}^{m} y_{ij}\,(i = 1, 2, \ldots, N)$, will then represent a summed observation over the $i$-th elements of the $j$-th cluster, e.g. over the mondays, tuesdays, etc. respectively.

The standard convention consists of assuming at $x_{ij}$ that $E(f_j(x_{ij})) = 0$, since otherwise there will be a free constant in each of the functions [13]. Additive models have a strong motivation as a useful data analytic tool. Each function is estimated by an algorithm proposed by [9] and called *backfitting* (or *Gauss–Seidel*) *algorithm*. As the estimator for $\beta_0$, the arithmetic mean (average) of the output data is used: $\hat{\beta}_0 = \text{ave}(y_i|i = 1,\ldots,N) := (1/N)\sum_{i=1}^{N} y_i$. This procedure depends on the partial residual against $x_{ij}$:

$$r_{ij} = y_i - \hat{\beta}_0 - \sum_{k\neq j} \hat{f}_k(x_{ik}) \tag{1.6}$$

and consists of estimating each smooth function by holding all the other ones fixed [11].

To prove its *convergence*, Buja and Hastie [6] used the normal equation (see subsection 2.5.1) for an arbitrary solution $\tilde{\mathbf{f}}$ to reduce the problem to the solution of a corresponding homogeneous system. That is, $\hat{\boldsymbol{P}}\tilde{\mathbf{f}} = \hat{\boldsymbol{Q}}\mathbf{y}$ and it is necessary to find $\mathbf{f}$ such that $\hat{\boldsymbol{P}}(\mathbf{f} - \tilde{\mathbf{f}}) = \mathbf{0}$. For this reason, they used a linear *fixed point equation* of the form $\hat{\mathbf{T}}\mathbf{f} = \mathbf{f}$ and they show that for $\mathbf{y} = \mathbf{0}$, backfitting converges to some solution of $\hat{\boldsymbol{P}}\mathbf{f} = \mathbf{0}$. If the normal equations are nonsingular, this implies convergence to $\mathbf{f} = \mathbf{0}$ [6]. Both the algorithm of *Jacobi* and *Gauss-Seidel* are special cases of the asynchronous algorithm which has been studied by [1]. This algorithm is defined by

$$x_i^{p+1} = \begin{cases} x_i^p & \text{if } i \notin J(p) \\ F_i\left(x_i^{s_1(p)}, \ldots, x_\alpha^{s_\alpha(p)}\right) & \text{if } i \in J(p),\ i = 1, 2, \ldots, \alpha,\ p = 0, 1, \ldots. \end{cases}$$

Here, all vectors $x \in \mathbb{R}^n$ considered are split into the form $x = (x_1, \ldots, x_\alpha)^T \in \mathbb{R}^n$ where $x_i \in \mathbb{R}^{n_i}$, $J = \{J(p)\}_{p \in \mathbb{N}}$ is a subset of the indexes of the components updated at the $p$-th iteration, $S = \{(s_1(p), \ldots, s_\alpha(p))\}_{p \in \mathbb{N}}$ is a sequence of $\mathbb{N}^\alpha$ and $F = (F_1, \ldots, F_\alpha) : \mathbb{R}^n \to \mathbb{R}^n$ is considered as an operator. Convergence and complexity of this algorithm are controlled by $p - s_i(p)$. If we take $s_i(p) = p$ $(p \in \mathbb{N}, i \in \{1, 2, \ldots, \alpha\})$, $J(p) = \{1, 2, \ldots, \alpha\}$ $(p \in \mathbb{N})$, this algorithm describes the Jacobi algorithm. If we take $s_i(p) = p$ $(p \in \mathbb{N}, i \in \{1, 2, \ldots, \alpha\})$, $J(p) = p + 1 (\mathrm{mod}\, \alpha)$ $(p \in \mathbb{N})$, then, our general algorithm describes Gauss–Seidel algorithm [1]. We thus conclude that the wide framework of [1] offers a way to future refinements of our investigation.

**1.3.2. Additive models revisited.** We allow a different and new motivation [23]: additionally to the approach given by a *separation* of the variables $x_j$ done by the functions $f_j$, we perform a *clustering* of the input data of the variable $x$ by a partitioning of the domain into higher dimensional interval $Q_j$ or, in the 1-dimensional case: intervals $I_j$, and an estimation of $f_j$ with reference to the knots lying in $Q_j$ (or $I_j$), respectively. The elements in the $j$-th cluster are called $x_{ij}$, they serve as interpolation knots in the iterations of the modified backfitting algorithm which we are presenting, referring to residual values $r_{ij}$. In any such case, a higher dimensional interval (i.e. product of intervals in $\mathbb{R}$) or interval is taking the place of a dimension or coordinate axis. We mostly refer to one dimension; the higher dimensional case can then be treated by a combination of separation and clustering. The sequence of those clusters can represent any kind of subsequent periods or seasons, any successive time intervals which have some comparable meaning or in some way corresponding to each other. Herewith, the functions $f_i$ are considered more as allocated to sets $I_j$ (or $Q_j$) rather than depending on some special, sometimes arbitrary, elements of those sets (input data) or associated output values. This new interpretation and usage of additive models (or the generalized ones which are introduced next) is a key step of this article.

## 2. Generalized additive models

To extend the additive model to a wide range of distribution families, Hastie and Tibshirani [13] proposed *generalized additive models* (*GAM*) which are among the most practically used modern statistical techniques. Many often-used statistical models

belong to this general class, e.g. additive models for Gaussian data, nonparametric logistic models for binary data, and nonparametric log-linear models for Poisson data.

### 2.1. *Definition of a generalized additive model*

Let us have $m$ covariates $X_1, X_2, \ldots, X_m$, comprised by the $m$-tuple $X=(X_1, \ldots, X_m)^T$, and a response $Y$ to the input $X$ assumed to have exponential family density $h_Y(y, \alpha, \varpi)$ with the mean $\mu = E(Y|X_1, \ldots, X_m)$ linked to the predictors through a link function $G$. Here, $\alpha$ is called the natural parameter and $\varpi$ is the dispersion parameter. Then, in our regression setting, a *generalized additive model* takes the form

$$\eta(X) = G(\mu) = \beta_0 + \sum_{j=1}^{m} f_j(X_j), \tag{2.1}$$

where the functions $f_j$ are unspecified (''nonparametric'') and $\theta=(\beta_0, f_1, \ldots, f_m)^T$ is the unknown parameter to be estimated. The incorporation $\beta_0$ as some average outcome allows us to assume $E(f_j(X_j))=0$ $(j=1, \ldots, m)$. Often, the unknown functions $f_j$ are elements of a finite dimensional space of functions and these functions, depending on the cluster knots, are mostly assumed to be splines approximating the data. The spline orders (or degrees) are suitably chosen depending on the density and variation properties of the corresponding data in the $x$ and $y$ components, respectively. Then, our problem of specifying $\theta$ becomes a finite-dimensional parameter estimation problem.

### 2.2. *Clustering of input data*

**2.2.1. Introduction.** *Clustering* is the process of organizing objects into groups $I_1, I_2, \ldots, I_m$ or, higher dimensionally: $Q_1, Q_2, \ldots, Q_m$, whose elements are similar in some way. A *cluster* is therefore a collection of objects which are ''similar'' between them and are ''dissimilar'' to the objects belonging to other clusters.

In this article, we understand clustering always as being accompanied by a *partitioning* of the (input) space, including space coverage. In other words, it will mean a classification in the absence of different labels or categories. The aim of clustering is to determine the intrinsic grouping in a set of unlabeled data. Therefore, we decide about clustering methods which depend on a criterion. This criterion must be supplied by the user in such a way that the result of the clustering will suit his needs [18]. Clustering algorithms can be applied in many fields like marketing, biology, libraries, book ordering, insurance, city-planning or earthquake studies. For further information we refer to [4].

**2.2.2. Clustering for generalized additive models.** Financial markets have different kinds of trading activities. These activities work with considerably long horizons, ranging from days and weeks to months and years. For this reason, we may have any kind of data. The three parts of figure 1 show some important cases of input data distribution and clustering in the way of [23]: the *equidistant case* (cf. 1(a)) where all points can be put into one cluster (or interval) $I_1$, the *equidistant case with regular breaks* (weekends, holidays, etc. cf. 1(b) where the regularly neighboring points and the

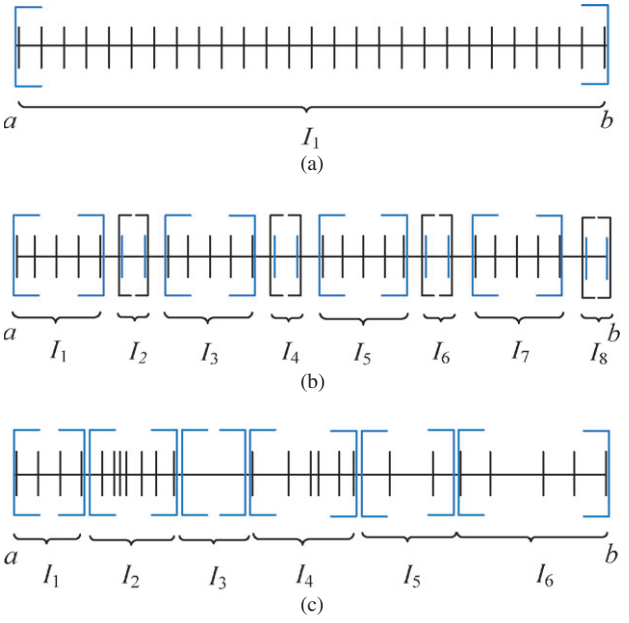


Figure 1. Three important cases of *input data distribution* and its *clustering*: (a) equidistance, (b) equidistance with breaks, and (c) general case.

free days could be put in separate cluster intervals $I_j$, and the *general case* (cf. 1(c)) where there are many interval $I_j$ of different interval lengths and densities. Furthermore, we can also include properties of the output data $y$ into this clustering.

Now, we take into account the data variation (for a first impression cf. figure 2).

Without loss of generality, we may assume that the number $N_j$ of input data points $x_{ij}$ in each cluster $I_j$ is the same, say $N_j \equiv N\,(j = 1, 2, \ldots, m)$. Otherwise, there will be no approximation needed at data points missing and the residuals of our approximation will be 0 there. Furthermore, given the output data $y_{ij}$, we denote the aggregated value over all the $i$-th output values of the clusters by

$$y_i := \sum_{j=1}^{m} y_{ij}\,(i = 1, 2, \ldots, N).$$

In figure 1(b), this data summation may refer to all the days $i$ from Monday to Friday. Herewith, the cluster can also have a chronological meaning. By definition, up to the division by $m$, the values $y_i$ are averages of the output values $y_{ij}$.

### 2.3. *Splines*

Let $x_{1j}, x_{2j}, \ldots, x_{Nj}$ be $N$ distinct knots of $[a, b]$, where $a \le x_{1j} < x_{2j} < \ldots < x_{Nj} \le b$. The function $f_k(x)$ on the interval $[a, b]$ (or in $\mathbb{R}$) is a spline of some degree $k$ relative to the knots $x_{ij}$ if

(1) $f_{k|[x_{ij}, x_{i+1j}]} \in IP_k$ (polynomial of degree $\le k$; $i = 1, \ldots, N-1$),
(2) $f_k \in C^{k-1}[a, b]$.

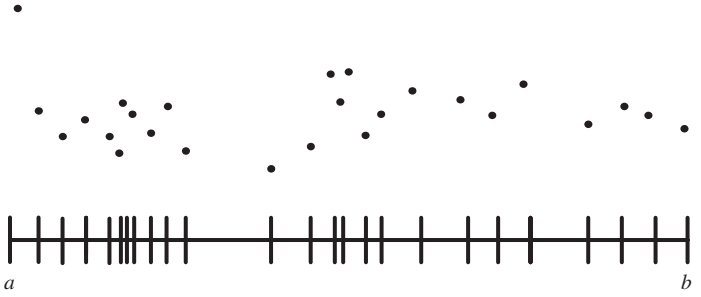


Figure 2. Example of data (scatterplot); here, we refer to figure 1, case (c).

To characterize a spline of degree $k$, $f_{k,i} := f_{k|[x_{ij}, x_{i+1j}]}$ can be represented by

$$f_{k,i}(x) = \sum_{l=0}^{k} g_{li}\left(x - x_{ij}\right)^l \left(x \in \left[x_{ij}, x_{i+1j}\right]\right).$$

There are $(k+1)(N-1)$ coefficients $g_{li}$ to be determined. Furthermore, it has to hold $f^{(l)}_{k,i-1,}(x_{ij}) = f^{(l)}_{k,i}(x_{ij})\,(i = 1, \ldots, N-2;\ l = 0, \ldots, k-1)$. Then, there are $k(N-2)$ conditions, and the remaining degrees of freedom are $(k+1)(N-1) - k(N-2) = k + N - 1$ [20].

It is necessary to select the order of the spline, the number of knots and their placement. We shall subsequently follow the latter approach; there, we define a special *index* for the selection of the spline degrees and, herewith, their orders. For basic information about higher and 1-dimensional splines, we refer to [7].

### 2.4. *Density, variation and index*

In [23], we defined a special *index* for the selection of the spline degrees based on variation and density for corresponding $j$-th interval $I_j$, herewith, of their orders (see also figure 2). This index is defined as $\mathrm{Ind}_j := D_j \cdot V_j$ or, more generally, $\mathrm{Ind}_j := d_j(D_j) \cdot v_j(V_j)$, where $d_j$, $v_j$ are some positive, strongly monotonically increasing functions selected by the modeller, then, $D_j$, $V_j$ are density and variation of the input data $x_{ij}$ in the $j$-th interval $I_j$, respectively. These are defined by $D_j :=$ (numbers of point $x_{ij}$ in $I_j$)/(length of $I_j$) and $V_j := \sum_{i=1}^{N-1} |y_{i+1j} - y_{ij}|$. This definitions can be directly generalized to the higher dimensional interval rather than intervals $I_j$, by referring to the higher dimensional volumes. Since in our algorithm we do the spline interpolation with respect to the residuals $r_{i,j}$, we can, in each iteration separately, refer to the variation

$$V_j := \sum_{i=1}^{N-1} \left|r_{i+1j} - r_{ij}\right|.$$

We determine the degree of the splines $f_j$ with the help of the numbers $\mathrm{Ind}_j$. If the number $\mathrm{Ind}_j$ is big, we choose a high degree of the spline. In this case, the spline may have a more complex structure and many coefficients have to be determined, i.e. we may have many system equations or a high-dimensional vector of unknowns. The solution can then become more difficult; furthermore, a high degree of the splines $f_1$, $f_2, \ldots, f_m$ causes a high curvatures or oscillations, i.e. there is a high "energy" implied.

This means a higher (co)variance or instability under data perturbations. As the extremal case of high curvature we consider nonsmoothness, meaning an instantaneous movement at a point which does not obey to any tangent.

The previous words introduced a model-free element into our explanations. Indeed, the concrete determining of the spline degree can be done adaptively by the implementer who writes the code. From a close mathematical perspective, we propose to introduce discrete *threshold*s $\gamma_\nu$ and to assign to all the intervals of indices $\mathrm{Ind} \in [\gamma_\nu, \gamma_{\nu+1})$ the same specific spline degrees. This determination and allocation has to base on the above reflections and data (or residuals) given.

For the above reasons, we impose some control on the oscillation. To make the oscillation smaller, the curvature of each spline must be bounded by the penalty parameter. We introduce a *penalty parameter* into the criterion of minimizing RSS, called *penalized sum of squares PRSS* now [12]:

$$PRSS(\beta_0, f_1, \ldots, f_m) := \sum_{i=1}^{N} \left\{ y_i - \beta_0 - \sum_{j=1}^{m} f_j(x_{ij}) \right\}^2 + \sum_{j=1}^{m} \varphi_j \int_a^b \left[ f_j''(t_j) \right]^2 \mathrm{d}t_j. \tag{2.2}$$

While the first term measures "goodness of data fitting", the second term means "penalties" and is defined by the functions' curvatures. Here, the interval $[a, b]$ is the union of all the intervals $I_j$. In the case of separation of variables, the interval bounds may also depend on $j$, i.e. they are intervals $[a_j, b_j]$. We recall that one basic idea of the additive models just consists a model with variables separated, and remind that our research is also applicable to that interpretation.

In (2.2), $\varphi_j \geq 0$ are tuning or *smoothing* parameters and represent a trade-off between the first and the second term. Large values of $\varphi_j$ yield smoother curves, smaller values result in more fluctuation. It can be shown that the minimizer of *PRSS* is an additive spline model [11]. In [23], we constructed a new solution method for *PRSS*. For this reason, there we introduced

$$F(\beta_0, f) := \sum_{i=1}^{N} \left\{ y_i - \beta_0 - \sum_{j=1}^{m} f_j(x_{ij}) \right\}^2 \quad \text{and} \quad g_j(f) := \int [f_j'']^2 \mathrm{d}t_j - M_j$$

with $M_j > 0$ being some prescribed upper bounds for the corresponding integral curvature term. Intending to keep the curvature integrals as small as possible, this bound can be interpreted as an "(error) tolerance" and it can be selected by the practitioner. Herewith, the combined standard form of our regression problem subject to the constrained curvature condition looks as follows:

$$\begin{aligned} &\text{Minimize} \quad F(\beta_0, f) \\ &\text{subject to} \quad g_j(f) \leq 0 \quad (j = 1, 2, \ldots, m). \end{aligned} \tag{2.3}$$

Now, *PRSS* can be represented with the following *Lagrange function*:

$$L((\beta_0, f), \varphi) := \sum_{i=1}^{N} \left\{ y_i - \beta_0 - \sum_{j=1}^{m} f_j(x_{ij}) \right\}^2 + \sum_{j=1}^{m} \varphi_j \left( \int \left[ f_j''(t_j) \right]^2 \mathrm{d}t_j - M_j \right), \tag{2.4}$$

where $\varphi := (\varphi_1, \ldots, \varphi_m)^T$. Here, $\varphi_j$ are *penalty parameters* [5]. In the light of our optimization problem, they can now be seen as *Lagrange multipliers* associated with the

constraints $g_j \leq 0$. For the *Lagrangian dual problem* we refer to [23]. Any solution or iteratively approximate solution of this optimization problem serves to determine the smoothing parameters $\varphi_j$ and, in particular, the functions $f_j$ will be found, likewise their bounded curvatures $\int [f_j''(t_j)]^2 \mathrm{d}t_j$. In section 3.3, we will construct another continuous optimization problem which is an alternative to our backfitting algorithm concept that implies penalization. Now, we go on with refining and discussing the backfitting concept for the additive model.

## 2.5. *Modified backfitting algorithm for additive model*

**2.5.1. Additive model revisited.** For the additive model given in subsection 1.3, we modified the backfitting algorithm used before for fitting additive model (cf. subsection 1.3) [23]. For this reason, we used the following theoretical setting in terms of conditional expectation [6], where for $j = 1, 2, \ldots, m$:

$$f_j(X_j) = P_j\left(Y - \beta_0 - \sum_{k \neq j} f_k(X_k)\right) := E\left(Y - \beta_0 - \sum_{k \neq j} f_k(X_k) | X_j\right). \tag{2.5}$$

To find $f_j(X_j)$ in additive model, we added the term $-\sum_{k=1}^{m} \varphi_k \int [f_j''(t_k)]^2 \mathrm{d}t_k$ to equation (2.5) and used the fact of $\sum_{k \neq j}^{m} \varphi_k \int [\hat{f}_k''(t_k)]^2 \mathrm{d}t_k = c_j$, then, we updated (2.5) as

$$f_j(X_j) + \varphi_j \int \left[f_j''(t_j)\right]^2 \mathrm{d}t_j \leftarrow E\left(Y - \beta_0 - \sum_{k \neq j} \left(f_k(X_k) + \varphi_k \int \left[f_k''(t_k)\right]^2 \mathrm{d}t_k\right) \middle| X_j\right), \tag{2.6}$$

where on both sides the integration is over the interval $[a, b]$ and defines constants. Here, the functions $\hat{f}_j$ are unknown and will be determined in the course of iteration.

If we denote $Z_k(X_k) := f_k(X_k) + \varphi_k \int [f_k''(t_k)]^2 \mathrm{d}t_k$ (the same for $j$), we get the update formula

$$Z_j(X_j) \leftarrow E\left(Y - \beta_0 - \sum_{k \neq j} Z_k(X_k) \middle| X_j\right). \tag{2.7}$$

We use theoretical setting of the conditional expectation for random variables $(Y, X)$ (for the formula without intercept $\beta_0$, we refer to [6]).

$$\begin{pmatrix} I & P_1 & \cdot & \cdot & P_1 \\ P_2 & I & \cdot & \cdot & P_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ P_m & P_m & \cdot & \cdot & I \end{pmatrix} \begin{pmatrix} Z_1(X_1) \\ Z_2(X_2) \\ \cdot \\ \cdot \\ Z_m(X_m) \end{pmatrix} = \begin{pmatrix} P_1(Y - \beta_0 \mathbf{e}) \\ P_2(Y - \beta_0 \mathbf{e}) \\ \cdot \\ \cdot \\ P_m(Y - \beta_0 \mathbf{e}) \end{pmatrix}, \tag{2.8}$$

where $\mathbf{e}$ is the $N$-vector or entries 1; or, in short, $\boldsymbol{PZ} = \boldsymbol{Q}(\boldsymbol{Y} - \beta_0)$. Here, $\boldsymbol{P}$ and $\boldsymbol{Q}$ represent the matrix and vector of the included operators, respectively. If we want to apply the normal equations to any given discrete experimental data, we must change the variables $(\boldsymbol{Y}, \boldsymbol{X})$ in (2.8) by their realizations $(y_i, \mathbf{x}_i)$, $\mathbf{x}_i = (x_{i1}, \ldots, x_{im})^T$, and the conditional expectations $P_j = E(\cdot | X_j)$ by smoothers $S_j$ on $x_j$. Then, we shortly get

$$\hat{\boldsymbol{P}}\boldsymbol{z} = \hat{\boldsymbol{Q}}\left(\boldsymbol{y} - \hat{\beta}_0\right) =: \hat{\boldsymbol{Q}}\mathbf{y}_1, \tag{2.9}$$

where $\boldsymbol{y} - \hat{\beta}_0 =: \mathbf{y}_1$ and $S_j = (h_{jl}(x_i))_{\substack{i=1,\ldots,N \\ l=1,\ldots,N}}$ and are smoothing matrices of type $N \times N$, $\mathbf{z}_j$ are $N$-vectors representing the spline function $\hat{f}_j + \varphi_j \int [\hat{f}_j''(t_j)]^2 \mathrm{d}t_j$ in a canonical form (1.12); i.e. $\sum_{l=1}^{N} \theta_{jl} h_{jl}(X)$ (with the number of unknowns equal to the number of conditions). In this notation, without loss of generality, we already changed from lower spline degrees $d_j$ to a maximal one $d$, and to the order $N$.

Furthermore, (2.9) is an $(Nm \times Nm)$-system of *normal equations*. The solutions to (2.9) satisfy $\boldsymbol{z_j} \in \Re(S_j)$, where $\Re(S_j)$ is the range of the linear mapping $S_j$, since we update by $\boldsymbol{z_j} \leftarrow S_j(\boldsymbol{y} - \hat{\beta}_0 \mathbf{e} - \sum_{k \neq j} z_k)$. In case we want to emphasize $\hat{\beta}_0$ among the unknowns, i.e. $(\hat{\beta}_0^T, z_1^T, \ldots, z_m^T)^T$, then we can write a new equation which can be represented equivalently to (2.9) [23].

There is a variety of efficient methods for solving the system (2.9), which depend on both the number and type of smoother used [19].

In the following, we shall focus on *additive models* but will point out the essence of what the *generalized* additive models will request in a remark.

**2.5.2. Modified backfitting algorithm.** Gauss–Seidel method, applied to blocks consisting of vectorial component $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m$, exploits the special structure of (2.9). It coincides with the backfitting algorithm. If in the algorithm we write $\hat{z}_j = \hat{f}_j + \varphi_j \int [\hat{f}_j''(t_j)]^2 \mathrm{d}t_j$ (in fact, the functions $\hat{f}_j$ are unknown), then the $l$-th iteration in the backfitting or Gauss–Seidel includes the additional penalized curvature term. When we do not forget the step-wise update of the penalty parameter $\varphi_j$ and not mention it explicitly, then the framework of the procedure looks as follows:

(1) *initialize* $\hat{\beta}_0 = (1/N) \sum_{i=1}^{N} y_i, \quad \hat{f}_j \equiv 0 \Rightarrow \hat{z}_j \equiv 0 \quad \forall j$
(2) *cycle* $j = 1, 2, \ldots, m, 1, 2, \ldots, m, \ldots$.

$$\hat{z}_j \leftarrow S_j \left[ \left\{ y_i - \hat{\beta}_0 - \sum_{k \neq j} \hat{z}_k(x_{ik}) \right\}_{i=1}^{N} \right].$$

This iteration is done until the individual functions do not change: here, in each iterate, $\hat{z}_j$ is with the spline referring to the knots $x_{ij}$ found by the values $y_i - \hat{\beta}_0 - \Sigma_{k \neq j} \hat{z}_k(x_{ik}) (i = 1, 2, \ldots, N)$, i.e. by the other $\hat{z}_k$ and, finally, by the functions $\hat{f}_k$ and the penalty (smoothing) parameter $\varphi_k$. Actually, since by definition it holds that $\hat{z}_j = \hat{f}_j + \varphi_j \int \hat{f}_j''(t_j)^2 \mathrm{d}t_j$, throughout the algorithm we must have a *book keeping* about both $\hat{f}_j$ and the curvature effect $\varphi_j \int [\hat{f}_j''(t_j)]^2 \mathrm{d}t_j$ controlled by the penalty parameter $\varphi_j$ which we can update from step to step [23]. This book keeping is guaranteed since $\hat{f}_j$ and the curvature $\int [\hat{f}_j''(t_j)]^2 \mathrm{d}t_j$ can be determined via $\hat{z}_j$ and, herewith,

$$\hat{f}_j := \hat{z}_j - \varphi_j \int \left[ \hat{f}_j''(t_j) \right]^2 \mathrm{d}t_j.$$

**2.5.3. Discussion about modified backfitting algorithm.** Provided we regard our optimization problem on (2.2) (cf. also (2.5)) as fixed with respect to $\varphi_j$, then we can carry over the *convergence theory* about additive models (see section 1.3) to the present modified backfitting for additive model, replacing the functions $\hat{f}_j$ by $\hat{z}_j$. However, at

least approximately, we have to guarantee feasibility also, i.e. $\int[\hat{f}_j''(t_j)]^2 \mathrm{d}t_j \le M_j$ $j=1,\ldots,m$. If $\int[\hat{f}_j''(t_j)]^2 \mathrm{d}t_j \le M_j$, then we preserve the value of $\varphi_j$ for $l \leftarrow l+1$; otherwise, we increase $\varphi_j$. But this update changes the values of $\hat{z}_j$ and, herewith, the convergence behavior of the algorithm. Moreover, the modified backfitting algorithm bases on both terms in the objective function to be approximated by 0; too large an increase of $\varphi_j$ can shift too far away from 0 the corresponding penalized curvature value in the second term.

The iteration stops if the functions $f_j$ become stationary, i.e. not changing very much and if we request it, if $\sum_{i=1}^{N}\{y_i - \beta_0 - \sum_{j=1}^{m} f_j(x_{ij})\}^2$ becomes sufficiently small, i.e. lying under some error threshold $\varepsilon$, and, in particular, $\int[\hat{f}_j''(t_j)]^2 \mathrm{d}t_j \le M_j$ $(j=1,2,\ldots,m)$.

**2.5.4. A remark on fitting generalized additive models.** The algorithm described so far fits just additive models and it provides an estimation of the functions $\hat{f}_j$. In contrast, any algorithm for *generalized* additive models is a little more complicated. These models are extensions of generalized linear models [25], obtained by replacing form $\eta(X) = G(\mu) = \beta_0 + \Sigma_{j=1}^{m} X_j\beta_j$ with the additive form $\eta(X) = G(\mu) = \beta_0 + \Sigma_{j=1}^{m} f_j(X_j)$. For computing the maximum likelihood estimates in a generalized linear model, one can use the iteratively reweighted least-squares procedure [13]. For a generalized linear model, the maximum likelihood estimate of $\beta$ is defined by the score equations

$$\sum_{i=1}^{N} x_{ij}\left(\frac{\partial\mu_i}{\partial\eta_i}\right) C_i^{-1}(y_i - \mu_i) = 0 \quad (j = 0, 1, \ldots, m),$$

where $C_i$ is the variance matrix for $Y_i$, $(\partial\mu_i/\partial\eta_i)x_{ij} = (\partial\mu_i/\partial\beta_j)\,(i=1,\ldots,N;$ $j=0,\ldots,m)$ and we assume that in above equation $x_{i0}=1$. The *Fisher scoring procedure* is the standard method for solving these equations. It involves a Newton–Raphson algorithm. An equivalent procedure convenient for generalized additive models is called *dependent variable regression* and it is a form of the iteratively reweighted least-squares procedure. Actually, the algorithm which is used to estimate generalized additive models consisting of a combination of backfitting and local scoring algorithms, therefore, estimating generalized additive models that consist of two loops. Inside each step of the local scoring algorithm (outer loop), there is a weighted backfitting algorithm (inner loop) which estimates the functions $f_j$ until convergence is achieved. Then, based on the estimates from this weighted backfitting algorithm, a new set of weights is calculated and the next iteration of the scoring algorithm starts. If we have a vector of coefficient, $\beta^0$, vector for linear predictor $\eta^0 = (\eta_1^0, \ldots, \eta_N^0)^T$and $\mu^0 = (\mu_1^0, \ldots, \mu_N^0)^T$, the framework of the *local scoring algorithm* procedure looks as follows [13]:

I. Initialization:

$$\beta_0 = G\left(\sum_{i=1}^{N} y_i/n\right); \quad f_j^0 = 0\ (j=1,\ldots,k),\ (k=0)$$

II. Iterate: $m \leftarrow m+1$

Form the adjusted dependent variable:

$$s_i = \eta_i^0 + \left(y_i - \mu_i^0\right)\left(\frac{\partial\eta_i}{\partial\mu_i}\right)_0 \quad \text{with } \eta_i^0 = \beta_0^0 + \sum_{j=1}^{m} f_j^0\left(x_{ij}\right) \quad \text{and} \quad \mu_i^0 = G^{-1}\left(\eta_i^0\right).$$

Form the weights:

$$w_i^{-1} = \left(\frac{\partial \eta_i}{\partial \mu_i}\right)_0^2 (C_i^0).$$

Fit an additive model to $S_i$, to obtain estimated functions $f_j^1$, the additive predictor $\eta_i^1$, and the expectation $\mu_i^1$.

Then, compute the convergence criterion with respect to two neighboring iterations

$$\Delta(\eta^1, \eta^0) = \frac{\sum_{j=1}^m \left\| f_j^1 - f_j^0 \right\|_2}{\sum_{j=1}^m \left\| f_j^0 \right\|_2}.$$

III. Repeat step 2 replacing $\eta^0$ by $\eta^1$ until $\Delta(\eta^1, \eta^0)$ is below some small threshold.

Here, $\|f\|_2 := \|(f(x_{ij}), \ldots, f(x_{Nj}))^T\|_2$ is the length of the vector evaluations of $f$ at the $N$ sample points.

Further refining improvements and refinements of the generalized additive model and the corresponding modified backfitting algorithm are possible (cf [23]). However, because of our discussion around the need of an adaptive choice of the penalty parameters while having to guarantee convergence, there is a need for more developed and elegant methods of *continuous optimization theory*. These have to become an important complementary technology and alternative to the concept of backfitting algorithm. In particular, *conic quadratic programming* will be introduced and studied in our next section.

## 3. On conic programming and its application in statistical learning with spline regression

### 3.1. *Introduction: convex and conic programming*

Convex programming deals with problems consisting of minimizing a convex function over a convex set. Such problems arise frequently in many different application fields and have many important properties, like strong duality theory and the fact that any local minimum is a global minimum. These programs are not only computationally tractable, but they also have theoretically efficient solution methods. Convex programming consists of several important specially structured classes of problems such as semidefinite programming, second-order cone programming, and geometric programming. Let us give some information about convex programming by benefiting from [15,16].

Geometrically, a convex program has the form:

$$\min_x c^T x, \quad \text{where} \quad x \in X;$$

where, $c \in \mathbb{R}^n$ and $X \subseteq \mathbb{R}^n$ is a convex set. *Linear programming* (*LP*), in which the objective and all constraint functions $f_i (i = 0,1, \ldots, m)$ are linear, is the simplest case of a convex program:

$$\min_{u \in \mathbb{R}^n} f_0(u), \quad \text{where} \quad f_i(u) \leq 0 \ (i = 1, 2, \ldots, m). \tag{3.1}$$

Such a problem can be written in the canonical form

$$\min_x \ c^T x, \quad \text{where} \quad Ax - b \in K := \mathbb{R}^n_+. \tag{3.2}$$

If, however, the objective or constraints are nonlinear, then we must take into account the nonlinearity in the corresponding function $f_i$ in (3.1). It is easily seen [15] that a convex program (3.1) can be represented in the conic form similar to (3.2):

$$\min_x c^T x, \quad \text{where } Ax - b \in K, \tag{3.3}$$

here, $K \subseteq \mathbb{R}^N$ is a cone (closed, pointed, convex and with a nonempty interior), and $\mathbb{R}^n \to \mathbb{R}^N$, defined by $x \mapsto Ax$, is a linear embedding.

Generally, convex programs depend on three generic cones $K$ (in the second case referring to the Euclidean or $\ell_2$ norm):

$$\begin{aligned} &\textit{nonnegative orthant}: && \mathbb{R}^n_+ = \{x \in \mathbb{R}^n | x \geq 0\}, \\ &\textit{direct products of Lorentz cone}: && \mathbf{L}^n = \left\{(x,t) \in \mathbb{R}^n \times \mathbb{R} | \|x\|_2 \leq t\right\}, \\ &\textit{semidefinite cone}: && S^n_+ = \left\{X \in S^n : X \succeq 0\right\}; \end{aligned}$$

they will get introduced in more detail below. The optimization problems based on these three cones can be solved by primal-dual interior point methods. These methods are very effective methods for *linear*, *conic quadratic* and *semidefinite* programming – all are examples of conic problems.

In the following sections, we shall pay attention to the class of *conic quadratic* problems. Then, motivated by our problems from statistical learning, which we apply in financial mathematics and computational biology, we introduce and investigate a very important modern class of conic quadratic programming problems.

We are about to consider the conic quadratic program. For the cone underlying these problems, it can be described explicitly as the dual cone. Because in many cases, "duality" is very important for understanding of original models and converting it into equivalent forms better suited for numerical processing, etc.

### 3.2. *Conic quadratic programming*

The $n$-dimensional *ice-cream* (:=*second-order*, or *Lorentz*) cone $\mathbf{L}^n$ is defined by:

$$\mathbf{L}^n = \left\{ x = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n | x_n \geq \sqrt{x_1^2 + \cdots + x_{n-1}^2} \right\} \quad (n \geq 2).$$

A *conic quadratic problem is a conic* problem,

$$\min_x \ c^T x, \quad \text{where} \quad Ax - b \in \mathbf{K}, \tag{3.4}$$

for which the cone $\mathbf{K}$ is a direct product of several "*ice-cream cones*":

$$\begin{aligned} \mathbf{K} &= \mathbf{L}^{n_1} \times \mathbf{L}^{n_2} \times \cdots \times \mathbf{L}^{n_k} \\ &= \left\{ \left(y[1]^T, \ldots, y[k]^T\right)^T \middle| y[i] \in \mathbf{L}^{n_i} \quad (i = 1, 2, \ldots, k) \right\}. \end{aligned} \tag{3.5}$$

From (3.5) we can see that a conic quadratic program is an optimization problem with a linear objective function and finitely many "*ice-cream constraints*"

$$A_i x - b_i \in \mathbf{L}^{n_i} \quad (i = 1, 2, \ldots, k),$$

where

$$[A, b] = \left[[A_1, b_1]^T, \ldots, [A_k, b_k]^T\right]^T$$

is the partition of the data matrix $[A, b]$ corresponding to the partition of $y$ in (3.5). Thus, our conic quadratic program can be written as

$$\min_x c^T x, \quad \text{where} \quad A_i x - b_i \in \mathbf{L}^{n_i} \ (i = 1, 2, \ldots, k) \tag{3.6}$$

Sometimes, the relation $A_i x - b_i \in \mathbf{L}^{n_i}$ is also written in the form of a vector inequality, namely, $Ax_i - b \succeq_{\mathbf{L}^{n_i}} 0$ or $Ax_i \succeq_{\mathbf{L}^{n_i}} b$. This means a partial ordering. More generally, this kind of notation and partial order can be used in any finite-dimensional Euclidean space $\mathbf{E}$, where a good vector inequality "$\succeq$" is completely identified by the set $\mathbf{K}$ of "$\succeq$"-nonnegative vectors: $\mathbf{K} = \{a \in \mathbf{E} | a \succeq 0\}$, where $a \succeq b \Leftrightarrow a - b \succeq 0$ ($\Leftrightarrow a - b \in \mathbf{K}$). But the set $\mathbf{K}$ cannot be arbitrary. It must be a pointed convex cone. We note that every *pointed* convex cone $\mathbf{K}$ in $\mathbf{E}$ induces a partial ordering on $\mathbf{E}$, given by "$\succeq_K$", where $a \succeq_{\mathbf{K}} b \Leftrightarrow a - b \succeq_{\mathbf{K}} 0 \Leftrightarrow a - b \in \mathbf{K}$ [15].

Partitioning the data matrix $[A_i, b_i]$ by

$$[A_i, b_i] = \begin{bmatrix} D_i & d_i \\ p_i^T & q_i \end{bmatrix},$$

with $D_i$ being of the type $(n_i - 1) \times (\dim x)$, the problem can be written as

$$\min_x c^T x, \quad \text{where } \|D_i x - d_i\|_2 \leq p_i^T x - q_i \ (i = 1, 2, \ldots, k). \tag{3.7}$$

Here, $\|\cdot\|_2$ is the Euclidean norm. This is a most explicit form of the conic problem and the one which we will use. In this form, $D_i$ are matrices of the same row dimension as $x$. Furthermore, the lengths of the column vectors $d_i$ are the column dimensions of the matrices $D_i$, and $p_i$ are column vectors of the same dimension as $x$; finally, $q_i$ are reals. It can immediately be seen that (3.5) is indeed a cone, in fact a self-dual one: $\mathbf{K}_* = \mathbf{K}$ [15].

Consequently, the problem dual to (3.4) is

$$\max_\lambda \ b^T \lambda, \quad \text{where } A^T \lambda = c, \ \lambda \in \mathbf{K}. \tag{3.8}$$

If we write $\lambda$ as $\lambda := (\lambda_1^T, \lambda_2^T, \ldots, \lambda_k^T)^T$ with $m_i$-dimensional blocks $\lambda_i$, then the dual problem can be stated as follows:

$$\max_{\lambda_1, \ldots, \lambda_n} \quad \sum_{i=1}^k b_i^T \lambda_i, \quad \text{where } \sum_{i=1}^k A_i^T \lambda_i = c \text{ and } \lambda_i \in \mathrm{L}^{n_i} \ (i = 1, 2, \ldots, k). \tag{3.9}$$

If it is taken $\lambda_i = (\kappa_i^T, \nu_i)^T$ with a scalar component $\nu_i$, and using the meaning of "$\geq_{\mathbf{L}^n} 0$", it can be shown that following form is the problem dual to (3.7):

$$\max_{(\mu_i),(\nu_i)} \sum_{i=1}^{k} \left[\kappa_i^T d_i + \nu_i q_i\right], \quad \text{where } \sum_{i=1}^{k} \left[D_i^T \kappa_i + \nu_i p_i\right] = c, \|\kappa_i\|_2 \leq \nu_i \ (i = 1, 2, \ldots, k). \tag{3.10}$$

The design variables in (3.10) are column vectors $\kappa_i$, having the same dimensions as the vectors $d_i$, and reals $\nu_i$ ($i = 1, 2, \ldots, k$). The programs (3.7) and (3.10) are standard forms of a conic quadratic problem and of its dual.

Sometimes, optimization problems arising in applications are not in their standard forms; it is very important to always identify the original formulation by a standard optimization problem [15,17]. Generally, optimization problems are given in the form

$$\min_x f(x), \quad \text{where } x \in X. \tag{3.11}$$

Here, $f$ is a "loss function" and the set $X$ consists of admissible design vectors and is typically given by

$$X = \bigcap_{i=1}^{n} X_i, \tag{3.12}$$

where every $X_i$ is the set of vectors admissible for a particular design restriction which is, in many cases, represented by

$$X_i = \left\{x \in \mathbb{R}^n \,|\, g_j(x) \leq 0\right\}, \tag{3.13}$$

where $g_j(x)$ is $j$-th *constraint function*. Here, the objective $f$ in (3.11) is always assumed to be linear, otherwise the original objective function can be moved to the list of constraints, and the equivalent problem is written in the following form:

$$\min_{t,x} \ t, \quad \text{where } (t, x) \in \hat{X},$$
$$\text{with } \hat{X} := \left\{(x, t) | x \in X, \ t \geq f(x)\right\}.$$

This representation is helpful, e.g. when $f(x)$ is given in terms of the (nonsquared) Euclidean norm. In case where $f(x)$ is a sum of squares, i.e. a squared Euclidean norm, then we prefer to write $t^2 \geq f(x)$, $t \geq 0$, which is in accordance with the definition of the Lorentz cone. In the following, we will use any of both conventions about indeed *equivalent* reformulations just as being helpful.

Thus, we may assume that the original problem looks in this way:

$$\min_x \ c^T x, \quad \text{where } x \in X := \bigcap_{i=1}^{n} X_i.$$

In order to determine that $X$ has a standard form, one needs a kind of dictionary which contains different forms of the same structure. Such a dictionary is built for the conic quadratic programs. Thus, it can be understood when a given set $X$ can be represented by *conic quadratic inequalities* $\|Dx - d\|_2 \leq p^T x - q$. Shortly, it is *CQr*

(*conic quadratic representable*), if there exists a system of finitely many vector inequalities of the form

$$A_j\begin{pmatrix} x \\ u \end{pmatrix} - b_j \geq_{\mathbf{L}^{m_j}} 0, \tag{3.14}$$

in the variables $x \in \mathbb{R}^n$ and additional variables $u$ such that $X$ is the projection of the solution set of (3.14) onto the $x$-space. This means: $x \in X$ if and only if one can extend $x$ to a solution $(x; u)$ of the system

$$x \in X \Leftrightarrow \exists u : A_j\begin{pmatrix} x \\ u \end{pmatrix} - b_j \geq_{\mathbf{L}^{m_j}} 0 \ (j = 1, 2, \ldots, N).$$

Every such system (3.14) is called a *conic quadratic representation* or, in short, a *CQR*, of the set $X$.

### 3.3. *Application of conic quadratic programming to regression theory with splines*

Let us show how optimization over cones can be applied for a problem class from data mining and statistical learning which is motivated by real-world applications in, e.g. the financial sector or computational biology. In section 2, we formulated the optimization problem as follows,

$$\begin{aligned} &\min F(\beta_0, f), \\ &\quad \text{where } g_j(f) \leq 0 \ (j = 1, 2, \ldots, m). \end{aligned} \tag{3.15}$$

Here, we have the objective function $F(\beta_0, f) := \sum_{i=1}^{N}\{y_i - \beta_0 - \sum_{j=1}^{m} f_j(x_{ij})\}^2$ of least-squares and the constraint functions (in simplified notation) $g_j(f) := \int [f_j''(t_j)]^2 \mathrm{d}t_j - M_j$. We can equivalently write our optimization problem in the following form:

$$\begin{aligned} &\min_{t,\beta_0,f} \ t, \\ &\text{where } \sum_{i=1}^{N}\left\{y_i - \beta_0 - \sum_{j=1}^{m} f_j(x_{ij})\right\}^2 \leq t^2, \quad t \geq 0, \\ &\qquad \int \left[f_j''(t_j)\right]^2 \mathrm{d}t_j \leq M_j \quad (j = 1, 2, \ldots, m). \end{aligned} \tag{3.16}$$

Here, *equivalence* refers to the positions of the optimal solutions in the sense of the pair of variables $(\beta_0, f)$. As mentioned previously, the functions $f_j$ are elements in a corresponding spline spaces, i.e., linear combinations of the parametrical form:

$$f_j(x) = \sum_{l=1}^{d_j} \theta_l^j h_l^j(x), \tag{3.17}$$

where $h_l^j : \mathbb{R} \to \mathbb{R}$ is the $l$-th transformation (base spline) of $x$ $(l = 1,2, \ldots, d_j)$ $(\theta_l^j)$ is the $(l,j)$-th entry of the family $\theta = (\theta_l^j)_{l=1,\ldots,d_j;j=1,\ldots,m}$ and for the sake of simplicity, by introducing additional terms with coefficients 0, we may assume that $h_l^j \equiv h_l$, $d_j \equiv d$ $(j = 1, 2, \ldots, m)$ such that the family becomes a matrix. We recall that our splines will refer to the corresponding knots $x_{ij}$ in the sense of input data where the approximation (regression) bases on, whereas in the course of backfitting algorithm a real interpolation

is stepwise performed there with respect to residual values. From now on, when representing the function dependence of the objective function, we may write $\theta$ instead of $f$. Instead of $\int[f_j''(t_j)]^2 \mathrm{d}t_j$ we will use an approximative discreted form, e.g. by evaluating the base splines $f_j''(\cdot)$ at the knots $x_{ij}$. To be more precise: either, we integrate between the end points $a < b$, uniformly for all $j$; in this case, we would add some further knots $x_{ij} \in [a, b]$ in addition to our cluster points $x_{ij}$ which are located in the interior of $I_j := [a_j, b_j]$. Or we cut off $f_j''(\cdot)$ outside of $I_j$, but add the points $a_j$ and $b_j$ to our cluster points $x_{ij}$ from $I_j$. Now, we get the following approximative evaluation:

$$\begin{aligned}\int \left[f_j''(t_j)\right]^2 \mathrm{d}t_j &\cong \sum_{i=1}^{N-1} \left[f_j''(x_{ij})\right]^2 (x_{i+1j} - x_{ij}) \\ &= \underbrace{\left(f_j''(x_{1j})\omega_1, \ldots, (x_{N-1j})\omega_{N-1}\right)}_{:=V_j^T(\beta_0,\theta)} \underbrace{\left(f_j''(x_{1j})\omega_1, \ldots, (x_{N-1j})\omega_{N-1}\right)^T}_{:=V_j(\beta_0,\theta)},\end{aligned}$$

where $\omega_i := \sqrt{x_{i+1j} - x_{ij}}$ $(i = 1, 2, \ldots, N-1)$. Let us abbreviate:

$$\begin{aligned}V(\theta) &:= \left(V_1^T(\theta), \ldots, V_N^T(\theta)\right)^T \text{ and } W(\beta_0, \theta) \\ &:= \left(y_1 - \beta_0 - \sum_{j=1}^{m} f_j(x_{1j}), \ldots, y_N - \beta_0 - \sum_{j=1}^{m} f_j(x_{Nj})\right)^T.\end{aligned}$$

Then, our optimization problem becomes

$$\begin{aligned}&\min_{t,\beta_0,\theta} \ t, \\ &\text{where} \quad \|W(\beta_0,\theta)\|_2^2 \le t^2, \\ &\qquad\quad \|V_j(\beta_0,\theta)\|_2^2 \le M_j \quad (j = 1, 2, \ldots, m), \\ &\qquad\quad 0 \le t,\end{aligned} \tag{3.18}$$

where $\|W\|_2^2 := W^T W$ and $\|V\|_2^2 := V^T V$ denote Euclidean norm squared. In fact, for the ease of exposition, we use a notation with "squares" in order to suppress the occurrence of square roots firstly.

Let us now explicitly insert the parametrical form (3.17) of the functions $f_j$ into this optimization problem. Then, our optimization problem looks as follows:

$$\begin{aligned}&\min_{t,\beta_0,f} \ t, \\ &\text{where} \quad \sum_{i=1}^{N} \left\{ y_i - \beta_0 - \sum_{j=1}^{m} \sum_{l=1}^{d_j} \theta_l^j h_l^j(x_{ij}) \right\}^2 \le t^2, \\ &\qquad\quad \sum_{i=1}^{N-1} \left\{ \sum_{l=1}^{d_j} \theta_l^j \omega_i h_l^{j''}(x_{ij}) \right\}^2 \le M_j \quad (j = 1, 2, \ldots, m), \\ &\qquad\quad 0 \le t.\end{aligned} \tag{3.19}$$

For all $i = 1, 2, \ldots, N-1$ we can write

$$\sum_{j=1}^{m}\sum_{l=1}^{d}\theta_l^j h_l\left(x_{ij}\right) = \theta_1^1 h_1(x_{i1}) + \cdots + \theta_d^1 h_d(x_{i1}) + \cdots + \theta_1^m h_1(x_{im}) + \cdots + \theta_d^m h_d(x_{im})$$
$$= \left(h_1(x_{i1}), \ldots, h_d(x_{i1})\right)\left(\theta_1^1, \ldots, \theta_d^1\right)^T + \cdots + \left(h_1(x_{im}), \ldots, h_d(x_{im})\right)$$
$$\times \left(\theta_1^m, \ldots, \theta_d^m\right)^T,$$

or

$$\sum_{j=1}^{m}\sum_{l=1}^{d}\theta_l^j h_l\left(x_{ij}\right) = H_i^1\theta^1 + \cdots + H_i^m\theta^m = \left(H_i^1, \ldots, H_i^m\right)\left(\theta^{1T}, \ldots, \theta^{mT}\right)^T = H_i\theta,$$

where $\theta^j := (\theta_1^j, \ldots, \theta_d^j)^T, \theta = (\theta^{1T}, \ldots, \theta^{mT})^T$, indices $H_i^j := (h_1(x_{ij}), \ldots, h_d(x_{ij}))$ $(j = 1, 2, \ldots, m)$ and $H_i := (H_i^1, \ldots, H_i^m)$ $(i = 1, 2, \ldots, N)$. Furthermore, we get

$$\int\left[f_j''(t_j)\right]^2 \mathrm{d}t_j \cong \sum_{i=1}^{N-1}\left[f_j''(x_{ij})\right]^2\left(x_{i+1j} - x_{ij}\right)$$
$$\cong \sum_{i=1}^{N-1}\left[\sum_{l=1}^{d}\theta_l^j\omega_i h_l''\left(x_{ij}\right)\right]^2$$
$$= \sum_{i=1}^{N-1}\left[H_i^{j''}\omega_i\theta^j\right]^2,$$

where we use the notation $H_i^{j''} := (h_1^{j''}(x_{ij}), \ldots, h_d^{j''}(x_{ij}))$ $(i = 1, 2, \ldots, N-1;\ j = 1, 2, \ldots, m)$.

If we assume that $\beta_0$ is fixed via the estimation $\hat{\beta}_0 := \mathrm{ave}(y_i | i = 1, 2, \ldots, N)$ by the arithmetic mean of the values $y_i$, then our optimization problem takes the following brief form:

$$\begin{aligned} &\min_{t,\theta}\ t, \\ &\text{where} \quad \|W(\theta)\|_2^2 \le t^2, \\ &\qquad\quad \left\|V_j(\theta)\right\|_2^2 \le M_j \quad (j = 1, 2, \ldots, m), \\ &\qquad\quad 0 \le t. \end{aligned} \tag{3.20}$$

Altogether, we obtain:

$$\begin{aligned} \|W(\theta)\|_2^2 &= \sum_{i=1}^{N}\left\{y_i - \hat{\beta}_0 - \sum_{j=1}^{m}\sum_{l=1}^{d_j}\theta_l^j h_l^j\left(x_{ij}\right)\right\}^2 = \sum_{i=1}^{N}\left\{y_i - \hat{\beta}_0 - H_i\theta\right\}^2, \\ \left\|V_j(\theta)\right\|_2^2 &= \sum_{i=1}^{N-1}\left[H_i^{j''}\omega_i\theta^j\right]^2 \end{aligned} \tag{3.21}$$

Then,

$$\begin{aligned} \left\|W(\theta)\right\|_2^2 &= \left\|\boldsymbol{H}\theta - u\right\|_2^2, \\ \left\|V_j(\theta)\right\|_2^2 &= \left\|\boldsymbol{H}_j\theta^j - 0\right\|_2^2, \end{aligned} \tag{3.22}$$

where $u_i = y_i - \hat{\beta}_0$ $(i = 1, 2, \ldots, N)$, $u = (u_1, \ldots, u_N)^T$ and $\boldsymbol{H} = (H_1^T, \ldots, H_N^T)^T$ with

$$\boldsymbol{H}_j = \left( H_1^{j''T} \omega_1, \ldots, H_{N-1}^{j''T} \omega_{N-1} \right)^T.$$

Then, the parametric form (3.19) looks as follows:

$$\begin{aligned} &\min_{t,\theta} \ t, \\ &\text{where} \quad \|\boldsymbol{H}\theta - u\|_2 \le t, \\ &\qquad \left\| \boldsymbol{H}_j \theta^j - 0 \right\|_2 \le \sqrt{M_j} \quad (j = 1, 2, \ldots, m), \end{aligned} \tag{3.23}$$

where $\boldsymbol{H}$ is an $N \times md$ matrix while $\boldsymbol{H}_j$ is an $(N-1) \times d$ matrix.

This optimization problem is a *conic quadratic* problem of the form (3.7) with

$$c = \begin{pmatrix} 1 & 0_{md}^T \end{pmatrix}^T, \ x = \begin{pmatrix} t & \theta^T \end{pmatrix}^T, \ D_1 = (\boldsymbol{H}, 0), \ d_1 = u, \ p_1 = (0, \ldots, 0, 1)^T, \ q_1 = 0$$

and, furthermore,

$$D_i = \left( 0, \ldots, 0, \boldsymbol{H}_{i-1}, 0, \ldots, 0, 0 \right), \ d_i = 0, \ p_i = 0^T \text{ and } q_i = -\sqrt{M_{i-1}} \quad \text{for } i = 2, 3, \ldots, m+1.$$

If we assume that $\beta_0$ is variable in (3.21), then,

$$\begin{aligned} \left\| W(\beta_0, \theta) \right\|_2^2 &= \sum_{i=1}^{N} \left\{ y_i - 1\beta_0 - H_i\theta \right\}^2 \\ &= \sum_{i=1}^{N} \left\{ y_i - (1, H_i) \begin{pmatrix} \beta_0 \\ \theta \end{pmatrix} \right\}^2 \\ &= \sum_{i=1}^{N} \left\{ y_i - R_i \tau \right\}^2 = \left\| y - \boldsymbol{R}\tau \right\|_2^2, \end{aligned} \tag{3.24}$$

where $R_i = (1, H_i, 0)$ $(i = 1, 2, \ldots, N)$, $\tau = (\beta_0, \theta^T, t)^T$ and $\boldsymbol{R} = (R_1^T, \ldots, R_N^T)^T$.

Because of the above equation, the optimization problem (3.18) has the following form:

$$\begin{aligned} &\min_{t,\beta_0,\theta} \ t, \\ &\text{where} \quad \left\| \boldsymbol{R}\tau - y \right\| \le t, \\ &\qquad \left\| \overline{\boldsymbol{H}}_j \theta^j - 0 \right\| \le \sqrt{M_j} \quad (j = 1, 2, \ldots, m), \end{aligned} \tag{3.25}$$

where $y = (y_1, \ldots, y_N)^T$. Here, $R_i$ and $\tau$ are $1 \times (md+2)$ and $(md+2) \times 1$ vectors, respectively, and $\boldsymbol{R}$ and $\overline{\boldsymbol{H}}_j$ are $N \times (md+2)$ and $(N-1) \times d$ matrices. This is of *conic quadratic* form again.

Let us consider (3.21) and $\beta_0$ be a variable in problem (3.21), then,

$$\left\| W(\beta_0, \theta) \right\|_2^2 = \sum_{i=1}^{N} \left\{ y_i - \beta_0 - H_i\theta \right\}^2 = \left\| y - \beta_0 \boldsymbol{e}_N - H\theta \right\|_2^2,$$

where $\mathbf{e}_N$ is the $N$-dimensional vector of all ones and $y = (y_1, y_2, \ldots, y_N)^T$. Because of the above equation, the optimization problem (3.18) will have the following form:

$$\begin{aligned} &\min_{t,\beta_0,\theta} \ t, \\ &\text{where} \quad \left\| H\theta + \beta_0 \boldsymbol{e}_N - y \right\|_2 \leq t, \\ &\qquad \left\| \boldsymbol{H}_j \theta^j - 0_{N-1} \right\|_2 \leq \sqrt{M_j} \quad (j = 1, 2, \ldots, m). \end{aligned} \tag{3.26}$$

In order to write the optimality condition for this problem, we will first reformulate (3.26) as follows:

$$\begin{aligned} &\min_{t,\beta_0,\theta} \ t, \\ &\text{such that} \quad v = \begin{pmatrix} 0_N & \boldsymbol{e}_N & H \\ 1 & 0 & 0_{md}^T \end{pmatrix} \begin{pmatrix} t \\ \beta_0 \\ \theta \end{pmatrix} + \begin{pmatrix} -y \\ 0 \end{pmatrix}, \\ &\qquad z_j = \begin{pmatrix} 0_{N-1} & 0_{N-1} & D_j \\ 0 & 0 & 0_{md}^T \end{pmatrix} \begin{pmatrix} t \\ \beta_0 \\ \theta \end{pmatrix} + \begin{pmatrix} 0_{N-1} \\ \sqrt{M_j} \end{pmatrix} \quad (j = 1, 2, \ldots, m), \\ &\qquad v \in L^{N+1}, \ z_j \in L^N \ (j = 1, 2, \ldots, m). \end{aligned} \tag{3.27}$$

The *dual problem* to the latter problem according to (3.9) is given by

$$\begin{aligned} &\max \quad (y^T, 0) x_0 + \sum_{j=1}^{m} \left( 0_{N-1}^T, \ -\sqrt{M_j} \right) x_j \\ &\text{such that} \quad \begin{pmatrix} 0_N^T & 1 \\ \boldsymbol{e}_N^T & 0 \\ H^T & 0_{md} \end{pmatrix} x_0 + \sum_{j=1}^{m} \begin{pmatrix} 0_{N-1}^T & 0 \\ 0_{N-1}^T & 0 \\ D_j^T & 0_{md} \end{pmatrix} x_j = \begin{pmatrix} 1 \\ 0_{md+1} \end{pmatrix}, \\ &\qquad x_0 \in L^{N+1}, x_j \in L^N \ (j = 1, 2, \ldots, m). \end{aligned} \tag{3.28}$$

Moreover, $(t, \beta_0, \theta, v, z_1, \ldots, z_m, x_0, x_1, \ldots, x_m)$ is a *primal-dual optimal solution* if and only if

$$\begin{aligned} &v = \begin{pmatrix} 0_N & 1_N & H \\ 1 & 0 & 0_{md}^T \end{pmatrix} \begin{pmatrix} t \\ \beta_0 \\ \theta \end{pmatrix} + \begin{pmatrix} -y \\ 0 \end{pmatrix}, \\ &z_j = \begin{pmatrix} 0_{N-1} & 0_{N-1} & D_j \\ 0 & 0 & 0_{md}^T \end{pmatrix} \begin{pmatrix} t \\ \beta_0 \\ \theta \end{pmatrix} + \begin{pmatrix} 0_{N-1} \\ \sqrt{M_j} \end{pmatrix} \quad (j = 1, 2, \ldots, m), \\ &\begin{pmatrix} 0_N^T & 1 \\ 1_N^T & 0 \\ H^T & 0_{md} \end{pmatrix} x_0 + \sum_{j=1}^{m} \begin{pmatrix} 0_{N-1}^T & 0 \\ 0_{N-1}^T & 0 \\ D_j^T & 0_{md} \end{pmatrix} x_j = \begin{pmatrix} 1 \\ 0_{md+1} \end{pmatrix}, \\ &x_0^T v = 0, \ x_j^T z_j = 0 \quad (j = 1, 2, \ldots, m), \\ &x_0 \in L^{N+1}, \ x_j \in L^N \quad (j = 1, 2, \ldots, m), \\ &v \in L^{N+1}, \ z_j \in L^N \quad (j = 1, 2, \ldots, m). \end{aligned} \tag{3.29}$$

**3.3.1. Solution methods for conic quadratic programming.** For solving convex optimization problems like semidefinite programming, geometric programming and, in particular, conic quadratic problems, classical *polynomial time algorithms* can be applied. But these algorithms have some disadvantage since they use local information on the objective function and the constraints. For this reason, to solve "well-structured" convex problems like conic quadratic problems, there are *interior point* algorithms [17,21] which were firstly introduced by *Karmarkar* (1984). These algorithms have the advantage of employing the structure of the problem, of allowing better complexity bounds and exhibiting a much better practical performance. Since in this present article we represented our spline regression problem as a conic quadratic problem, we became enabled for future research to exploit its special structure in this analytical and numerical way.

**3.3.2. Complexity of conic quadratic programming.** If we consider the following conic quadratic optimization program,

$$\min_x \ c^T x, \quad \text{where } \|D_i x - d_i\|_2 \le p_i^T x - q_i \quad (i = 1, 2, \ldots, k), \ \|x\|_2 \le t,$$

where the matrices $D_i$ are of the type $n_i \times n$, $p_i, x \in \mathbb{R}^n$ and $d_i \in \mathbb{R}^{n_i}$. Let us represent the data of (3.7) in the way of [15] by defining

$$\text{Data}((3.7)) := [k; n; n_1, \ldots, n_k; c; D_1, d_1, p_1, q_1; \ldots, D_k, d_k, p_k, q_k; t] \text{ and}$$

$$\text{Size}((3.7)) := \dim \text{Data}((3.7)) = \left(k + \sum_{i=1}^{k} n_i\right)(n+1) + k + n + 3.$$

The arithmetic complexity of $\varepsilon$-solution is given by

$$\text{Compl}((3.7), \varepsilon) := O(1)(k+1)^{1/2} n \left(n^2 + k + \sum_{i=1}^{k} n_i^2\right) \text{ Digits } ((3.7), \varepsilon),$$

where

$$\text{Digits}((3.7), \varepsilon) := \ln\left(\frac{(\text{Size}((3.7)) + \|\text{Data}((3.7))\|_1 \varepsilon^2)}{\varepsilon}\right)$$

is defined as the number of accuracy digits in an $\varepsilon$-solution to (3.7), referring to the sum (or $\ell_1$) norm.

We can specify the complexity related to our problem. However, again we must consider whether $\beta_0$ is a variable or not. If we assume that $\beta_0$ is *fixed*, in this case we consider (3.23), then

$$\text{Data}((3.23)) = [m+1; md+1; N; c; D_1, d_1, p_1, q_1; \ldots, D_{m+1}, d_{m+1}, p_{m+1}, q_{m+1}],$$
$$\text{Size}((3.23)) = \dim(\text{Data}((3.23))) = 5 + N(2md+1) - m$$

and

$$\text{Compl}(3.23, \varepsilon) = O(1)(m+2)^{1/2}(md+1)((md+1)^2 + (m+1) + N^2) \text{ Digits } (3.23, \varepsilon),$$

where

$$\mathrm{Digits}((3.23),\varepsilon) := \ln\left(\frac{(\mathrm{Size}((3.23)) + \|\mathrm{Data}((3.23))\|_1\varepsilon)}{\varepsilon}\right).$$

If we assume that $\beta_0$ is a *variable*, we consider (3.24), then

$$\mathrm{Data}((3.25)) = [m+1; md+2; N; c; D_1, d_1, p_1, q_1; \ldots, D_{m+1}, d_{m+1}, p_{m+1}, q_{m+1}],$$
$$\mathrm{Size}((3.25)) = \dim(\mathrm{Data}((3.25))) = 6 + N(2md+3) - m \quad \text{and}$$
$$\mathrm{Compl}((3.25),\varepsilon) = O(1)(m+2)^{1/2}(md+2)((md+2)^2 + (m+1) + N^2)\mathrm{Digits}((3.25),\varepsilon).$$

## 4. Concluding remarks

This article gives a contribution to the discrete approximation, or regression, of data in the one and in the multivariate cases. *Additive* and *generalized additive models* have been investigated, input data grouped by clustering, its density measured, data variation quantified, spline classes selected by indices, and their curvatures bounded with the help of penalization. The backfitting algorithm which is applicable for data classification has become modified accordingly and investigated. However, there are difficulties to use the modified backfitting algorithm, such as possible divergence. For this reason, we introduced developed methods of *continuous optimization* given by *conic quadratic programming* for which polynomial time *interior point methods* are applicable. By this investigation we hope to serve for future applications in finance, biology, medicine and many other areas of economy, science, technology, to welfare and development.

## References


[1] Addou, A. and Benahmed, A., 2005, Parallel synchronous algorithms for nonlinear fixed point problems. *International Journal of Mathematical Sciences*, **19**, 3175–3183.
[2] Alessio, A.S., 2004, Predicting and pricing the probability of Default, August 4. Available online at: www.personal.anderson.ucla.edu/alessio.saretto/default.pdf (accessed 2 September 2007).
[3] Aster, A., Borchers, B. and Thurber, C., 2004, *Parameter Estimation and Inverse Problems* (Amsterdam: Elsevier).
[4] Bock, H.H., Sokolowski, A. and Jajuga, K., 2002, *Classification, Clustering, and Data Analysis: Recent Advances and Applications* (Berlin: Springer Verlag).
[5] Boyd, S. and Vandenberghe, L., 2004, *Convex Optimization* (Cambridge: Cambridge University Press).
[6] Buja, A., Hastie, T. and Tibshirani, R., 1989, Linear smoothers and additive models. *The Annals of Statistics*., **17**(2), 453–510.
[7] De Boor, C., 2001, *Practical Guide to Splines* (Berlin: Springer Verlag).
[8] Fox, J., 2002, *Nonparametric regression, Appendix to an R and S-Plus Companion to Applied Regression* (London: Sage Publications).
[9] Friedman, J.H. and Stuetzle, W., 1981, Projection pursuit regression. *Journal of the American Statistical Association*, **76**, 817–823.
[10] Hastie, T. and Tibshirani, R., 1986, Generalized additive models. *Statistical Science*, **1**(3), 297–310.
[11] Hastie, T. and Tibshirani, R., 1987, Generalized additive models: some applications. *Journal of the American Statistical Association*, **82**(398), 371–386.
[12] Hastie, T., Tibshirani, R. and Friedman, J.H., 2001, *The Element of Statistical Learning* (New York: Springer Verlag).
[13] Hastie, T.J. and Tibshirani, R.J., 1990, *Generalized Additive Models* (New York: Chapman and Hall).
[14] Korn, R., Baydar, E., 2006, Workshop on Credit Rating in View of Basel II, Fraunhofer Institute for Industrial Mathematics.

[15] Nemirovski, A., 2002, Lectures on modern convex optimization, Israel Institute Technology, http://iew3.technion.ac.il/Labs/Opt/opt/LN/Final.pdf (accessed 2 September 2007).
[16] Nemirovski, A., 2005, Modern Convex Optimization, Lecture notes, Israel Institute of Technology.
[17] Nesterov, Y.E. and Nemirovski, A.S., 1993, *Interior Point Methods in Convex Programming* (Philadelphia: SIAM).
[18] Politicnico di Milano, Dipartimento de Elettronica e Informazione, A Tutorial on Clustering Algorithms. Available online at: http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/ (accessed 2 September 2007).
[19] Pringle, R.M. and Rayner, A.A., 1971, *Generalized Inverse Matrices with Applications to Statistics* (New York: Hafner Publishing).
[20] Quarteroni, A., Sacco, R. and Saleri, F., 1991, *Numerical Mathematics*, Vol. 37 (Berlin: Springer).
[21] Renegar, J., 2000, *Mathematical View of Interior Point Methods in Convex Programming* (Philadelphia: SIAM).
[22] Stone, C.J., 1985, Additive regression and other nonparametric models. *The Annals of Statistics*, **13**(2), 689–705.
[23] Taylan, P., and Weber, G.-W., 2007, New approaches to regression in financial mathematics by generalized additive models. *Journal of Computational Technologies*, **12**(2), 3–22.
[24] Weber, G.-W., Tezel. A., Taylan, P., Soyler, A. and Çetin, M., On Dynamics and optimization of gene-environment networks, Institute of Applied Mathematics, METU, 2006 (submitted to the special issue of *Optimization* in honour of the 60th birthday of Prof. Dr. H.Th. Jongen).
[25] Wood, S.N., 2006, *Generalized Additive Models*, An Introduction with R (New York: Chapman and Hall/CRC, Taylor and Francis Group).