**CASE STUDY: Spline Regression(spline_sum, logexp_sum, logistic, crossvalidation)**

***Background***

This case study approximates multidimensional data (with several independent variables) by a sum of splines using PSG function *spline_sum*. Input data:

$\vec{x}$ = matrix of independent factors,
$\vec{y}$= vectors of dependent values,
$\vec{D}$= vector of degrees of splines,
$\vec{K}$ = vector of the number of polynomial pieces in each spline,
$\vec{S}$= vector of smoothing degrees of the splines,
$\vec{V}$ = vector of splines knots ranges,
$\vec{U}$ = vector of upper bounds for knots of splines.

So called input Matrix of Scenarios contains matrix $\vec{x}$and $\vec{y}$ , and an input matrix with parameters specifies $\vec{D}$, $\vec{K}$, $\vec{S}$and optionally$\vec{V}$, $\vec{U}$.

PSG function Maximum Likelihood for Logistic Regression, logexp_sum, is minimized to find variables of splines providing the best approximation of data (see Problem 1). Estimated spline may "overfit" the in-sample data and this may result in poor out-of-sample performance. Cross-validation technique is used to check overfitting (see Problem 2). To prepare data for cross-validation we use PSG Crossvalidation(K,Matrix) matrix operation which splits input Matrix of Scenarios in N pairs of complementary sub-matrices. Overfitting can be reduced by dropping some factors. Selection of factors that should be left in the sum of splines is done by solving optimization problem (see Problem 3). This problem uses additional Boolean variables showing inclusion of factors in the sum of splines: 1 = factor is included in the sum of splines, 0 = not included (see Problem 3). Another way to reduce overfitting is to control values of splines at knot points by setting upper bound (see Problem 4).

***Notations***

*J*=number of points (observations) of vector of factors and corresponding independent variable;

*N* = number of independent factors;

$\vec{x}^{\,j} = (f_j^1, \dots, f_j^n, \dots, f_j^N)$ = vector of independent factors at point *j*, *j*=1,...,*J*;

$y_j$ = point of dependent variable corresponding to the point $x_j$,*j*=1,...,*J*;

$a_{dk}^n$= decision variable = coefficient of degree $d$ in polynomial piece $k$ for factor *n*, $d = 0, \dots, D_n$, $k = 1, \dots K_n$;

$G_{jn}^0(\vec{a}^n) = \sum_{d=0}^{D_n} a_{dk(j,n)}^n \cdot (f_j^n)^d$ = Gain Functions with zero scenario benchmark for factor *n* at point $f_j^n$, *j*=1,...,*J*. The piece number$k(j,n)$ depends on the factor *n* and point number *j*;

$\vec{a} = (\vec{a}^1, \dots, \vec{a}^n, \dots, \vec{a}^N)$ = joint vector of decision variables = vector of coefficients of polynomial pieces;

$G_j^0(\vec{a}) = \sum_{n=1}^N G_{jn}^0(\vec{a}^n) = \sum_{n=1}^N \sum_{d=0}^{D_n} a_{dk(j,n)}^n \cdot (f_j^n)^d$ = sum of Gain Functions with zero scenario benchmark at point $\vec{x}^{\,j}$,*j*=1,..., *J*;

$L_j(\vec{a}) = y_j - G_j^0(\vec{a})$ = Loss Functions at point$\vec{x}^{\,j}$, *j*=1,..., *J*;

$D_n$ = degree of spline of factor *n*, $D_n \geq 0$, integer;

$K_n$ = number of pieces for factor *n*, $K_n > 0$, integer;

$S_n$ = smoothing degree of a spline of factor *n*, $0 \leq S_n \leq D_n$, integer;

$f_n\_range$ = variable for range of variation for individual spline in knots of factor *n;*

$v_n$ = upper bounds for range of variation for individual spline in knots of factor *n;*

$\vec{D} = (D_1, D_2, \dots, D_N)$ = vector of polynomial degrees;

$\vec{K} = (K_1, K_2, \dots, K_N)$ = vector of polynomial piece numbers;

$\vec{S} = (S_1, S_2, \dots, S_N)$ = vector of polynomial smoothness;

$\vec{V} = (v_1, v_2, \dots, v_N)$ = vector of upper bounds for splines knots ranges;

$\mathbf{spline\_sum}(\vec{D}, \vec{K}, \vec{S}, \vec{f}^1, \vec{f}^2, \dots, \vec{f}^N, \vec{y}, \vec{a}) = \{L_1(\vec{a}), L_2(\vec{a}), \dots, L_J(\vec{a})\}$ = PSG function *Spline_sum* generating a set of loss scenarios $L_j(\vec{a})$ using initial data and a smoothing constraints (assuring smoothness according to $\vec{S}$ specification;

$\mathbf{logexp\_sum}(\mathbf{spline\_sum}(\vec{D}, \vec{K}, \vec{S}, \vec{f}^1, \vec{f}^2, \dots, \vec{f}^N, \vec{y}, \vec{a})) = \mathbf{logexp\_sum}(\vec{y}, G_1^0(\vec{a}), G_2^0(\vec{a}), \dots, G_J^0(\vec{a})) =$

$\frac{1}{J}\sum_{j=1}^{J}\left(y_j G_j^0(\vec{a}) - \ln\left(1 + \exp\left(G_j^0(\vec{a})\right)\right)\right) =$

PSG Maximum Likelihood for Logistics Regression function Logarithms Exponents Sum applied to *Spline_sum* function. All $y_j$ should be 0 or 1;

$\mathbf{logistic}(\mathbf{spline\_sum}(\vec{D}, \vec{K}, \vec{S}, \vec{f}^1, \vec{f}^2, \dots, \vec{f}^N, \vec{y}, \vec{a}))$= vector with components:

$\exp\left(G_j^0(\vec{a})\right) / \left(1 + \exp\left(G_j^0(\vec{a})\right)\right)$;

$\mathbf{polinom\_abs}(\eta_0, \vec{\eta}, \vec{z}, \vec{q}, \vec{x}) = \eta_0 + \sum_{i=1}^{I} \eta_i \left| x_i - z_i \right|^{q_i}$ = PSG Polynomial Absolute function;

$\eta_0$, $\vec{\eta} = (\eta_1, \eta_2, \dots, \eta_I)$ = parameters;

$\vec{y} = (y_1, y_2, \dots, y_I)$ = vector of parameters;

$\vec{q} = (q_1, q_2, \dots, q_I)$ = vector of parameters;

$M$ = number of factors included in the sum of splines;

$U$ = upper bound for value of sum of splines at knot points.

**Optimization Problem 1**

*maximizingLogarithms Exponents Sum for building spline*

$$\mathbf{max}_{\vec{a}}\mathbf{logexp\_sum}(\mathbf{spline\_sum}(\vec{D}, \vec{K}, \vec{S}, \vec{f}^1, \vec{f}^2, \dots, \vec{f}^N, \vec{y}, \vec{a})) \qquad \text{(CS1.1)}$$

*calculationof Logarithms Exponents Sum and Logisticon built spline*
**calculate**

$\mathbf{logexp\_sum}(\mathbf{spline\_sum}(\vec{D}, \vec{K}, \vec{S}, \vec{x}, \vec{y}, \vec{a}))$

$\mathbf{logistic}(\mathbf{spline\_sum}(\vec{D}, \vec{K}, \vec{S}, \vec{x}, \vec{y}, \vec{a}))$

**Optimization Problem 2**

*CrossValidation*

*maximizingLogarithms Exponents Sum for building spline*

$$\mathbf{max}_{\vec{a}}\mathbf{logexp\_sum}(\mathbf{spline\_sum}(\vec{D},\vec{K},\vec{S},\vec{f}^{1},\vec{f}^{2},\dots,\vec{f}^{N},\vec{y},\vec{a})) \quad \text{(CS2.1)}$$

*calculationof Logarithms Exponents Sum and Logistic on built spline*

**calculate**

$$\mathbf{logexp\_sum}(\mathbf{spline\_sum}(\vec{D},\vec{K},\vec{S},\vec{x},\vec{y},\vec{a}))$$

$$\mathbf{logistic}(\mathbf{spline\_sum}(\vec{D},\vec{K},\vec{S},\vec{x},\vec{y},\vec{a}))$$

**Optimization Problem 3**

*maximizing Logarithms Exponents Sum for building spline*

$$\mathbf{max}_{\vec{a}}\mathbf{logexp\_sum}(\mathbf{spline\_sum}(\vec{D},\vec{K},\vec{S},\vec{f}^{1},\vec{f}^{2},\dots,\vec{f}^{N},\vec{y},\vec{a})) \quad \text{(CS3.1)}$$

subject to

*constraint on the coefficients of every spline*

$$\mathbf{polynom\_abs}(\vec{f}^{i}) \leq u_i \quad \text{(CS3.2)}$$

*constraint on number of factors:*

$$\sum_{i=1}^{N} u_i \leq M \quad \text{(CS3.3)}$$

*binary variables:*

$u_i$ is binary variable, $i = 1, \dots, N$ (CS3.4)

*calculation of Logarithms Exponents Sum and Logistic on built spline*

**calculate**

$$\mathbf{logexp\_sum}(\mathbf{spline\_sum}(\vec{D},\vec{K},\vec{S},\vec{x},\vec{y},\vec{a}))$$

$$\mathbf{logistic}(\mathbf{spline\_sum}(\vec{D},\vec{K},\vec{S},\vec{x},\vec{y},\vec{a}))$$

**Optimization Problem 4**

*maximizing Logarithms Exponents Sum for building spline*

$$\mathbf{max}_{\vec{a}}\mathbf{logexp\_sum}(\mathbf{spline\_sum}(\vec{D},\vec{K},\vec{S},\vec{V},\vec{f}^{1},\vec{f}^{2},\dots,\vec{f}^{N},\vec{y},\vec{a})) \quad \text{(CS4.1)}$$

subject to

*constraint on the value of sum of splines at knot points*

$$\sum_{i=1}^{N} f_{i}\_range \leq U \quad \text{(CS4.2)}$$

*calculation Logarithms Exponents Sum and Logistic on built spline*

**calculate**

$$\mathbf{logexp\_sum}(\mathbf{spline\_sum}(\vec{D},\vec{K},\vec{S},\vec{V},\vec{x},\vec{y},\vec{a}))$$

$$\mathbf{logistic}(\mathbf{spline\_sum}(\vec{D},\vec{K},\vec{S},\vec{V},\vec{x},\vec{y},\vec{a}))$$