Computational Complexity of Dependence Estimation via Generalized Linear Models in Multidimensional Feature Spaces

Vadim Mottl, Alexander Tatarchuk
Computing Center of the Russian Academy of Sciences, Moscow, Russia
vmottl@yandex.ru
aitech@yandex.ru

Valentina Sulimova
Tula State University
Tula, Russia
vsulimova@yandex.ru

Olga Krasotkina
Markov Processes International
New Jersey, USA
o.v.krasotkina@yandex.ru

Alexey Morozov, Ilya Pugach
Moscow Institute of Physics and Technology, Moscow, Russia
ao.morozov@phystech.edu
iliapugach@gmail.com

***Abstract* — Usually, when speaking about dependence estimation in big sets of empirical data, it is adopted to suggest that the set of precedents does not fit in the memory of one computer, and some technology of distributed computing is required. However, even if the entire training set can be placed in one computer, the question remains how much time the training process will take. We keep here to the generalized linear methodology of dependence estimation, which covers, in particular, both regression estimation and pattern recognition. It is assumed that the training information (empirical data set) is a rectangular objects/features table. We consider here two kinds of algorithms of regularized empirical risk minimization, which are mutually opposite in their computational complexity relative to the number of features and the number of training objects, i.e., to the two sizes of the objects/features table. The computational complexity of one of them is linear with respect to the number of objects and polynomial relative to the number of features, whereas the other algorithm is of polynomial complexity in the number of features and linear in that of training objects. Thus, for any combination of the two sizes of the objects/features table, we have an algorithm whose computational complexity is linear relative to the greater of two sizes and polynomial with respect to the smaller of them. This property is especially favorable for the typical situation when the number of available features is much greater than that of training examples.**

Keywords— Dependence estimation, empirical risk, regularization, basic set of real-world objects, training set, object features, computational complexity.

I. Introduction

A. The Generalized Linear Model of a Dependence

Supervised regression analysis and pattern recognition are two most typical cases of dependence estimation from empirical data [1]. In both problems it is required to recover the unknown dependence of a hidden variable $y\in\mathbb{Y}$ associated with any real-world object on the observable vector of its numerical features $\mathbf{x}=(x_1\cdots x_n)^T\in\mathbb{R}^n$, when only a trainng set of real-world objects is available

$$\{(\mathbf{x}_j, y_j),\ j=1,...,N\},\ \mathbf{x}_j=(x_{j,1}\cdots x_{j,n})^T\in\mathbb{R}^n,\ y\in\mathbb{Y}. \quad (1)$$

The only difference between regression and pattern recognition is that the target variable in regression is an arbitrary real number $y\in\mathbb{Y}=\mathbb{R}$, whereas in recognition it is categorical, for instance, takes one of two values $y\in\mathbb{Y}=\{-1,1\}$.

The commonly adopted approach to these problems implies finding a linear decision rule, respectively,

(a) $\hat{y}(\mathbf{x}|\mathbf{a},b)=\mathbf{a}^T\mathbf{x}+b:\mathbb{R}^n\to\mathbb{R}$, or (2)

(b) $\hat{y}(\mathbf{x}|\mathbf{a},b)=\begin{pmatrix}1, & \mathbf{a}^T\mathbf{x}+b>0\\ -1, & \mathbf{a}^T\mathbf{x}+b<0\end{pmatrix}:\mathbb{R}^n\to\{-1,1\}$. (3)

which would be applicable to any new real-world object represented by its feature vector $\mathbf{x}\in\mathbb{R}^n$.

These are two particular cases of John Nelder's Generalized Linear Model of dependencies [2,3], in which the goal variable of any kind $y\in\mathbb{Y}$ is related to the linear regression via the so-called link function:

$$\begin{cases} z(\mathbf{x}|\mathbf{a},b)=\mathbf{a}^T\mathbf{x}+b:\mathbb{R}^n\to\mathbb{R} & -\text{ Generalized Linear Model},\\ q(y,z):\mathbb{Y}\times\mathbb{R}\to\mathbb{R}^+ & -\text{ link function}. \end{cases} \quad (4)$$

In what follows, we shall call the real-valued variable

$$z(\mathbf{x}\,|\,\mathbf{a},b)=\mathbf{a}^T\mathbf{x}+b\in\mathbb{R},\ \ \mathbf{x},\mathbf{a}\in\mathbb{R}^n,\ \ b\in\mathbb{R}, \quad (5)$$

the generalized linear feature of the real-world object represented by the feature vector $\mathbf{x}\in\mathbb{R}^n$. The link function $q(y,z)$, loss function in Vladimir Vapnik's terminology [4], is to be chosen by the observer and is meant to express his/her suggestion on how the Nature would penalize the estimate of the unknown y for an object $\mathbf{x}\in\mathbb{R}^n$ represented by its generalized numerical linear feature $z(\mathbf{x}\,|\,\mathbf{a},b)$.

Since the link function is chosen, the hyperplane parameters $(\mathbf{a},b)$ completely define the decision rule:

$$\hat{y}(\mathbf{x}|\mathbf{a},b)=\arg\min_{y\in\mathbb{Y}} q\big(y,\,z(\mathbf{x}|\mathbf{a},b)\big). \quad (6)$$

Particular dependence estimation problems differ from each other only in the choice of the link function, specifically:

- for regression $y\in\mathbb{R}$, $q(y,z)=(y-z)^2$, $\hat{y}(\mathbf{x}|\mathbf{a},b)=\mathbf{a}^T\mathbf{x}+b$; (7)
- for SVM pattern recognition $y=\pm1$,

$$q(y,z)=\max(0,1-yz),\ \ \hat{y}(\mathbf{x}|\mathbf{a},b)=\begin{cases}1, & \mathbf{a}^T\mathbf{x}+b\geq1,\\ -1, & \mathbf{a}^T\mathbf{x}+b<1;\end{cases} \quad (8)$$

- for logistic regression pattern recognition $y=\pm1$,

$$q(y,z)=\ln\big[1+\exp(-yz)\big],\ \ \hat{y}(\mathbf{x}|\mathbf{a},b)=\begin{cases}1, & \mathbf{a}^T\mathbf{x}+b\geq1,\\ -1, & \mathbf{a}^T\mathbf{x}+b<1.\end{cases} \quad (9)$$

B. The Principle of Empirical Risk Minimization

From the viewpoint of the Generalized Linear Approach to dependence estimation, the quality of the hyperplane parameters $(\mathbf{a},b)$ is the average value of the loss $q(y,\mathbf{a}^T\mathbf{x}+b)$ over all the real-world objects $(\mathbf{x},y)\in\mathbb{R}^n\times\mathbb{Y}$, which is usually called the average risk of error, let it be denoted as $AvR(\mathbf{a},b)$. However, average risk minimization $AvR(\mathbf{a},b)\to\min(\mathbf{a},b)$ is problematic because the properties of the hypothetical universe may be inexhaustibly complex.

Instead, it is commonly adopted to approximately estimate the average risk from the training set (1) as the arithmetic mean of the attainable loss values. This is the famous criterion of Empirical Risk minimization [4], which in our terms has the form

$$EmpR(\mathbf{a},b)=\frac{1}{N}\sum_{j=1}^{N}q(y_j,\,\mathbf{a}^T\mathbf{x}_j+b)\to\min\big(\mathbf{a}\in\mathbb{R}^n,\,b\in\mathbb{R}\big). \quad (10)$$

This optimization problem is convex if the link function $q(y,z)$ is chosen as convex with respect to z, as it is just the case with regression (7) and pattern recognition (8)-(9).

C. Selective ridge regularization

When the practical problem originates from a medical or industrial domain, the available amount of data N is usually limited, whereas the observer tries to take into account as many features n as possible in fear of losing important outward exhibitions of entities. Therefore, the amount of features often far dominates that of training objects $n \gg N$. If so, the problem of empirical risk minimization (10) becomes ill posed – there exist a continuum of models $(\mathbf{a}, b)$ that totally approximate the training data.

In this paper, we apply the selective ridge regularization first proposed in [5,6]

$$J(\mathbf{a}, b | \gamma, \mu) = \gamma \sum_{i=1}^{n} \begin{pmatrix} 2\mu |a_i|, & |a_i| \le \mu \\ \mu^2 + a_i^2, & |a_i| > \mu \end{pmatrix} + \sum_{j=1}^{N} q\left(y_j, \sum_{i=1}^{n} a_i x_{j,i} + b \right) \to \min(\mathbf{a} \in \mathbb{R}^n, b \in \mathbb{R}), \quad (11)$$

where $\mu \ge 0$ is the selectivity parameter. If $\mu = 0$, the regularization function coincides with the usual ridge regularization $\gamma \mathbf{a}^T \mathbf{a} + EmpR(\mathbf{a}, b) \to \min$. When the selectivity parameter grows $\mu > 0$, the penalty $\mu |a_i|$ drives to zero the coefficients at redundant features, which weakly contribute to diminishing of the empirical risk. Further growth of the selectivity parameter $\mu \to \infty$ results finally in complete zeroing of all the coefficients.

D. The aim and structure of the paper

Since the link function $q(y, z)$ (7)-(8) is assumed to be convex with respect to $z \in \mathbb{R}$ for each $y \in \mathbb{Y}$, the criterion of regularized empirical risk minimization $J(\mathbf{a}, b) \to \min$ remains be convex as a whole. The aim of this paper is studying the computational complexity of the problem of selective regularized empirical risk minimization for the opposite ratios of two sizes of the objects/features table $n \le N$ or $n > N$ – the number of features does not exceed that of objects or vice versa, there are more features in the model than objects in the training set.

II. The Straihtforward Formulation of the Regularized Empirical Risk Minimization Problem

The criterion of selctive regularizaed empirical risk minimization (11) is conxex function of $n+1$ variables $(\mathbf{a}, b) = (a_1, ..., a_n, b)$. If considered as a convex problem of general kind, it is imossible to numerically solve it with lower computational complexity than polynomial with respect to the number of features n.

Let us now to investigate the computational complexity relative the training set size.

The specificity of the objective function to be minimized is that the empirical risk $\sum_{j=1}^{N} q(y_j, z_j)$ is sum of convex functions of $n+1$ variables $z_j = \sum_{i=1}^{n} a_i x_{j,i} + b$ being, in their turn, linear functions of $(a_1, ..., a_n, b)$. Let $q(y, z)$ be chosen as twice differentiable functions of $z \in \mathbb{R}$ except a finite number of points:

$$\frac{\partial}{\partial z} q(y, z) = q'(y, z), \ \frac{\partial^2}{\partial z^2} q(y, z) = q''(y, z). \quad (12)$$

Then the gradient and Hessian of the sum at any point $(\hat{\mathbf{a}}, \hat{b})$ are linear combinations of respective vectors and matrices

$$\nabla_{\mathbf{a}} \sum_{j=1}^{N} q(y_j, \hat{z}_j) = \sum_{j=1}^{N} q'(y_j, \hat{z}_j) \mathbf{x}_j, \quad \nabla^2_{\mathbf{aa}} \sum_{j=1}^{N} q(y_j, \hat{z}_j) = \sum_{j=1}^{N} \mathbf{x}_j \mathbf{x}_j^T q''(y_j, \hat{z}_j). \quad (13)$$

This means that computation of the gradient and hessian of the objective function is of linear complexity in the number of training objects.

Thus, minimization of the regularized empirical risk in its original formulation is of polynomial comutational complexity in the number od features and of linear comlexity in the trainng set size.

Such a combination of two kinds of computaional complexity is favorable if the training set contains very many objects N repesented by a small number of features n. However, this kind of data is rather an excepcion than the rule. In practice, the inverse case is much more typical – a moderate training set size N, because it is limited by the difficulty of testing the nature, and a huge dimension of the feature space $n \gg N$, which is bounded only by the imagination of the observer. As a result, the polynomial computatinal compltxity with respect to the dimension of the feature space unacceptable in practice.

As an alternative to the straightforward criterion (11), we consider here the dual formulation of the regularized empirical risk minimizaion problem.

III. The Dual Formulation of the Regularized Empirical Risk Minimization Problem

A. Two Groups of Variables in the Criterion of Regularized Empirical Risk

The parameter of the generalized linear model of the sought-for dependence (4) is the vector argument $\mathbf{a} = (a_1 \cdots a_n)^T \in \mathbb{R}^n$ of the regularized empirical risk to be minimized (11). Let us consider the generalized linear features of all the training objects (5)

$$z_j = \mathbf{a}^T \mathbf{x}_j + b = \sum_{i=1}^{n} a_i x_{j,i} + b, \ j = 1, ..., N, \quad (14)$$

as an additional group of variables $\mathbf{z} = (z_1 \cdots z_N)^T \in \mathbb{R}^N$. Since the variables of these two groups are bound to each other by the system of equalities (14), the idea is to replace the straightforward criterion $J(\mathbf{a}, b)$ (11), $\mathbf{a} \in \mathbb{R}^n$, by some equivalent objective function $W(\boldsymbol{\lambda})$ whose argument would be vector $\boldsymbol{\lambda} = (\lambda_1 \cdots \lambda_N)^T \in \mathbb{R}^N$ of a smaller dimension $N < n$.

The dimension N is just the number of equalities that bind vector $\mathbf{z} = (z_1 \cdots z_N)^T \in \mathbb{R}^N$ to the vector $\mathbf{a} = (a_1 \cdots a_n)^T \in \mathbb{R}^n$. These equalities suggest another formulations of the empirical risk minimization problem (11):

$$\begin{cases} \gamma \sum_{i=1}^{n} \begin{pmatrix} 2\mu |a_i|, & |a_i| \le \mu \\ \mu^2 + a_i^2, & |a_i| > \mu \end{pmatrix} + \sum_{j=1}^{N} q(y_j, z_j) \to \min \\ z_j = \sum_{i=1}^{n} a_i x_{j,i} + b, \ j = 1, ..., N, \end{cases} (a_1, ..., a_n, b, z_1, ..., z_N), \quad (15)$$

We call this formulation disjoint, because the objective function is sum of two partial criteria, which are functions of, respectively, the direction vector $\mathbf{a} \in \mathbb{R}^n$ and the generalized features of the training objects $\mathbf{z} = (z_1 \cdots z_N)^T \in \mathbb{R}^N$ (14), in contrast to the initial unified formulation (11).

B. Properties of the disjoint empirical risk minimization problem

Before solving the disjoint problem (15), we have to introduce an additional notion, let it be called the lower bound of the link function:

$$\varphi(y,\lambda\,|\,\gamma)=-\inf_{z\in\mathbb{R}}\left(\frac{1}{2\gamma}q(y,z)+\lambda z\right),\ \lambda\in\mathbb{R}. \qquad (16)$$

The definition of this function contains the operation "inf", whose result, in the general case, is finite not for all the values of variable $\lambda\in\mathbb{R}$. Since in this paper function $q(y,z)$ is assumed to be convex with respect to $z\in\mathbb{R}$ for each $y\in\mathbb{Y}$, its derivative is nondecreasing function. If, in addition, $q(y,z)$ is chosen as differentiable by z except a finite number of points in $\mathbb{R}$, at which the left-hand or right-hand derivative exists, the range of the respective derivatives is known, let it be denoted as

$$g_{\inf}(y)=\inf_{z\in\mathbb{R}}\frac{\partial q(y,z)}{\partial z}\le\frac{\partial q(y,z)}{\partial z}\le g_{\sup}(y)=\sup_{z\in\mathbb{R}}\frac{\partial q(y,z)}{\partial z}. \qquad (17)$$

It is easy to see that function $\varphi(y,\lambda\,|\,\lambda)$ is convex in $\lambda\in\mathbb{R}$ for any link function $q(y,z)$, $z\in\mathbb{R}$.

C. The Lagrangian of the disjoint problem

The necessary and sufficient minimum condition for the convex equality-constrained objective function (15) is the saddle point of the respective Lagrangian as function of N Lagrange multipliers $(\lambda_1,...,\lambda_N)$ at constraints:

$$\begin{aligned}&L(a_1,...,a_n,b,z_1,...,z_N,\lambda_1,...,\lambda_N)=\frac{1}{2}\sum_{i=1}^{n}\begin{pmatrix}2\mu\,|a_i|,&|a_i|\le\mu\\ \mu^2+a_i^2,&|a_i|>\mu\end{pmatrix}+\\ &\frac{1}{2\gamma}\sum_{j=1}^{N}q(y_j,z_j)-\sum_{j=1}^{N}\lambda_j\left(\sum_{i=1}^{n}a_ix_{j,i}+b-z_j\right)=\\ &L_i(a_i,\lambda_1...,\lambda_N)-\left(\sum_{j=1}^{N}\lambda_j\right)b+\sum_{j=1}^{N}\left(\frac{1}{2\gamma}q(y_j,z_j)+\lambda_jz_j\right)\to\\ &\to\begin{cases}\min(a_1,...,a_n,b,z_1,...,z_N),\\ \partial/\partial\lambda_j=0,\ j=1,...,N,\end{cases}\ \text{where}\end{aligned} \qquad (18)$$

$$L_i(a_i,\lambda_1...,\lambda_N)=\frac{1}{2}\begin{pmatrix}2\mu\,|a_i|-2\left(\sum_{j=1}^{N}\lambda_jx_{j,i}\right)a_i,&|a_i|\le\mu\\ \mu^2+a_i^2-2\left(\sum_{j=1}^{N}\lambda_jx_{j,i}\right)a_i,&|a_i|>\mu\end{pmatrix}. \qquad (19)$$

Assume the Lagrange multipliers be fixed $(\lambda_1,...,\lambda_N)=(\hat\lambda_1,...,\hat\lambda_N)$, and consider the requirement $\to\min(a_1,...,a_n,b,z_1,...,z_N)$ in (18). Minimization by $(a_1,...,a_n)$ results in n single conditions for each variable in accordance with (19)

$$\hat a_i=\begin{cases}0,&\left(\sum_{j=1}^{N}\lambda_jx_{j,i}\right)^2\le\mu^2,\\ \sum_{j=1}^{N}\lambda_jx_{j,i},&\left(\sum_{j=1}^{N}\lambda_jx_{j,i}\right)^2>\mu^2,\end{cases}.\ i=1,...,n, \qquad (20)$$

and minimization by b yields $\sum_{j=1}^{N}\lambda_j=0$. (21)

D. The Low-Dimensional Dual Problem

It can be shown that (19) and (20) jointy give in (18)

$$\begin{aligned}&-\min_{a_1,...,a_n,b,z_1,...,z_N}L(a_1,...,a_n,b,z_1,...,z_N,\lambda_1,...,\lambda_N)=\\ &\frac{1}{2}\sum_{i=1}^{n}\left\{\max\left[0,\ \left(\sum_{j=1}^{N}\lambda_jx_{j,i}\right)^2-\mu^2\right]\right\}-\\ &\sum_{j=1}^{N}\left[\min_{z_j\in\mathbb{R}}\left(\frac{1}{2\gamma}q(y_j,z_j)+\lambda_jz_j\right)\right]\to\left(\frac{\partial}{\partial\lambda_j}=0,\ j=1,...,N\right).\end{aligned}$$

With respect to (16), (17) and (21), this yields the condition

$$\begin{cases}W(\lambda_1,...,\lambda_N\,|\,\gamma,\mu)=\frac{1}{2}\sum_{i=1}^{n}\left\{\max\left[0,\left(\sum_{j=1}^{N}\lambda_jx_{j,i}\right)^2-\mu^2\right]\right\}+\\ \sum_{j=1}^{N}\varphi(y_j,\lambda_j\,|\,\gamma)\to\min(\lambda_1,...,\lambda_N),\\ \sum_{j=1}^{N}\lambda_j=0,\ \ -\frac{1}{2\gamma}g_{\sup}(y_j)\le\lambda_j\le-\frac{1}{2\gamma}g_{\inf}(y_j).\end{cases} \qquad (22)$$

We have obtaned the dual problem for (15). By its mathematical structure, this is the convex programming problem with respect to Lagrange multipliers $(\lambda_1,...,\lambda_N)$ associated with N training objects, thus, its computatonal complexity is polynomial in N. Below in Section IV we consider the iterative Newton algorithm that solves the dual problem in the particular case of SVM pattern recognition.

When the values of Lagrange multipliers $(\hat\lambda_1,...,\hat\lambda_N)$ are found, the independnt formulas (20) provide linear computational complexity of computing the elements of the direction vector $(\hat a_1,...,\hat a_N)$.

It remains only to compute the estimate of the bias b in the generalized linear model (4). It is clear that minimizaton of the Lagrangian (18) by $(z_1,...,z_N)$ results in N single conditions

$$\hat z_j(\hat\lambda_j)=\arg\min_{z\in\mathbb{R}}\left(\frac{1}{2\gamma}q(y_j,z)+\hat\lambda_jz\right), \qquad (23)$$

and equalities (14) give the formula

$$\hat b=\frac{1}{N}\sum_{j=1}^{N}\left(\hat z_j(\hat\lambda_j)-\sum_{i=1}^{n}\hat a_ix_{j,i}\right). \qquad (24)$$

Before looking for an algorithm for the numerical solution of the dual problem, it is convenient to formulate the dual problem (22) in an equivalent form Each of n summands of the first sum in the dual objective function $W(\lambda_1,...,\lambda_N\,|\,\gamma,\mu)$ (22) differs from zero only if $\sum_{j=1}^{N}\lambda_jx_{j,i}>\mu$. Let $\mathbb{I}(\lambda_1,...,\lambda_N)$ stand for the subset of "active" features:

$$\mathbb{I}(\boldsymbol\lambda)=\mathbb{I}(\lambda_1,...,\lambda_N)=\left\{i:\sum_{j=1}^{N}\lambda_jx_{j,i}>\mu\right\}\subseteq\mathbb{I}=\{1,...,n\}. \qquad (25)$$

With respect to this notation, we have in (22)

$$\begin{cases}W(\boldsymbol\lambda\,|\,\gamma,\mu)=\frac{1}{2}\boldsymbol\lambda^T\left(\sum_{i\in\mathbb{I}(\boldsymbol\lambda)}\boldsymbol x_i\boldsymbol x_i^T\right)\boldsymbol\lambda+\sum_{j=1}^{N}\varphi(y_j,\lambda_j\,|\,\gamma)\to\min(\boldsymbol\lambda),\\ \sum_{j=1}^{N}\lambda_j=0,\ \ -\frac{1}{2\gamma}g_{\sup}(y_j)\le\lambda_j\le-\frac{1}{2\gamma}g_{\inf}(y_j),\end{cases} \qquad (26)$$

where $\boldsymbol x_i=(x_{1,i}\cdots x_{N,i})^T\in\mathbb{R}^N$ are vectors of the i th elements in all the training-set feature vectors.

IV. Iterative Algorithms of Solving the Dual Problem

A. The iterative descent algorithm with variable step length

The function $W(\boldsymbol\lambda\,|\,...)$ is differentiable at each point $\boldsymbol\lambda\in\mathbb{R}^N$. Moreover, it is even twice differentiable, and, so, Newton's method is appropriate to find the solution of the dual problem. If $\boldsymbol\lambda^k$ is the current approximation to the solution, then a supposedly better solution $\tilde{\boldsymbol\lambda}^{k+1}$ is defined by the completely differentiable convex programming problem

$$\begin{cases} \tilde{\boldsymbol{\lambda}}^{k+1} = \arg\min \tilde{W}^k(\boldsymbol{\lambda} \mid \gamma, \mu) = \\ \arg\min \left\{ \frac{1}{2} \boldsymbol{\lambda}^T \left(\sum_{i \in \mathbb{I}(\boldsymbol{\lambda}^k)} \boldsymbol{x}_i \boldsymbol{x}_i^T \right) \boldsymbol{\lambda} + \sum_{j=1}^{N} \varphi(y_j, \lambda_j \mid \gamma) \right\}, \\ \sum_{j=1}^{N} \lambda_j = 0, \ -\frac{1}{2\gamma} g_{\sup}(y_j) \le \lambda_j \le -\frac{1}{2\gamma} g_{\inf}(y_j), \end{cases} \quad (27)$$

which differs from (27) only by the fixed summation domain $\mathbb{I}(\boldsymbol{\lambda}^k)$.

To avoid boring mathematical reasoning in the case of the convex link function of general kind $q(y, z)$, we omit here the explanation of how to solve the convex programming problem (27). We will see in Section ??? that, in particular cases of regression (7) and pattern recognition (8)-(9), these will be quadratic programming problems easily solvable by traditional computational means.

It is well seen that the approximation $\boldsymbol{\lambda}^k = (\lambda_1^k, ..., \lambda_N^k)$ to the sought-for solution at step k occurs in (27) only via the subset of active features (25), let it be denoted as

$$\mathbb{I}^k = \mathbb{I}(\boldsymbol{\lambda}^k), \text{ initially, the full feature set } \mathbb{I}^0 = \{1, ..., n\}. \quad (28)$$

Let the supposedly better solution of the dual problem $\tilde{\boldsymbol{\lambda}}^{k+1}$ (27) at step k be found. It may happen that the length of Newton's step is too large, and it should be shortened. To check this necessity, it is enough to compare the values of the dual criterion (26) at points $\boldsymbol{\lambda}^k$ and $\tilde{\boldsymbol{\lambda}}^{k+1}$:

$$\text{If } W(\tilde{\boldsymbol{\lambda}}^{k+1} \mid \gamma, \mu) \le W(\boldsymbol{\lambda}^k \mid \gamma, \mu), \quad \text{the iteration is successful, and } \boldsymbol{\lambda}^{k+1} = \tilde{\boldsymbol{\lambda}}^{k+1}; \quad (29)$$

$$\text{if } W(\tilde{\boldsymbol{\lambda}}^{k+1} \mid \gamma, \mu) > W(\boldsymbol{\lambda}^{kk} \mid \gamma, \mu), \quad \text{the step is to be shortened.} \quad (30)$$

To find the appropriate length of Newton's step, we apply one-dimensional optimization of (27), namely, the golden section algorithm:

$$\begin{aligned} &\tau^{k+1} = \arg\min W\left[(\tau \boldsymbol{\lambda}^k) + (1-\tau) \tilde{\boldsymbol{\lambda}}^{k+1} \mid \gamma, \mu \right], \ 0 \le \tau \le 1, \\ &\boldsymbol{\lambda}^{k+1} = \tau^{k+1} \boldsymbol{\lambda}^k + (1-\tau^{k+1}) \tilde{\boldsymbol{\lambda}}^{k+1}. \end{aligned} \quad (31)$$

Actually, the algorithm iteratively runs over the subsets of regressors $\mathbb{I}^k = \mathbb{I}(\boldsymbol{\lambda}^k) \subset \mathbb{I} = \{1, ..., n\}$ (25) without cycles because $W(\boldsymbol{\lambda}^{k+1} \mid \gamma, \mu) \le W(\boldsymbol{\lambda}^k \mid \gamma, \mu)$ at each step. Thus, the stopping condition

$$\mathbb{I}(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\xi}^{k+1}) = \mathbb{I}(\boldsymbol{\lambda}^k, \boldsymbol{\xi}^k) \quad (32)$$

will be achieved after a finite number of steps.

B. Numerical realization of an iteration for particular cases of the link function

To complete description of the iterative algorithm, it remains only to specify the ways of solving the constrained problem (27) at each step in the particular cases of regression and pattern recognition. The specificity is contained in function $\varphi(y, \lambda \mid \gamma) = -\min_{z \in \mathbb{R}} \left(\frac{1}{2\gamma} q(y, z) + \lambda z \right)$ (16) and constraints $-\frac{1}{2\gamma} g_{\sup}(y_j) \le \lambda_j \le -\frac{1}{2\gamma} g_{\inf}(y_j)$ (17).

- *Regression*

Theorem 1. In the particular case of regression (7), (16) and (17), we have

$$g_{\sup}(y) = \infty, \ g_{\inf}(y) = -\infty, \quad \varphi(y, \lambda \mid \gamma) = \frac{1}{2} \gamma \lambda^2 - y\lambda, \quad (33)$$

the dual problem (27) at the k th step of the iteration process is quadratic

$$\begin{cases} \tilde{\boldsymbol{\lambda}}^{k+1} = \arg\min \tilde{W}^k(\boldsymbol{\lambda} \mid \gamma, \mu) = \arg\min \left\{ \frac{1}{2} \boldsymbol{\lambda}^T \left(\sum_{i \in \mathbb{I}(\boldsymbol{\lambda}^k)} \boldsymbol{x}_i \boldsymbol{x}_i^T \right) \boldsymbol{\lambda} - \right. \\ \left. \mu^2 \sum_{i \in \mathbb{I}(\boldsymbol{\lambda}^k)} 1 + \sum_{j=1}^{N} \left(\frac{1}{2} \gamma \lambda_j^2 - y_j \lambda_j \right) \right\}, \quad \sum_{j=1}^{N} \lambda_j = 0, \end{cases} \quad (34)$$

and its solution is defined by the system of linear equations $(N+1) \times (N+1)$

$$\underbrace{\begin{pmatrix} \sum_{i \in \mathbb{I}^k} \boldsymbol{x}_i \boldsymbol{x}_i^T + \gamma \mathbf{I}_{N \times N} & \mathbf{1}_N \\ \mathbf{1}_N^T & 0 \end{pmatrix}}_{N+1} \begin{pmatrix} \boldsymbol{\lambda} \\ \eta \end{pmatrix} = \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix} \Big\}_{N+1},$$

where $\eta \in \mathbb{R}$ is the idle Lagrange multiplier at the constraint $\mathbf{1}_N^T \boldsymbol{\lambda} = \sum_{j=1}^{N} \lambda_j = 0$. The final bias estimate (24) has the form

$$\hat{b} = \frac{1}{N} \sum_{j=1}^{N} \left(y_j - \sum_{i=1}^{n} \hat{a}_i x_{\alpha_i, j, i} \right) \text{ with respect to (20).} \quad (35)$$

- *Pattern recognition SVM*

Theorem 2. In the particular case of SVM pattern recognition (8), we have

$$\begin{cases} g_{\sup} = 1, \ g_{\inf} = 0, \ y = -1, \\ g_{\sup} = 0, \ g_{\inf} = -1, \ y = 1, \end{cases} \quad 0 \le y\lambda \le \frac{1}{2\gamma}, \ \varphi(y, \lambda) = -y\lambda, \quad (36)$$

the solution to the dual problem at the k th step of the iteration process (27) is that of quadratic programming problem

$$\begin{cases} \tilde{\boldsymbol{\lambda}}^{k+1} = \arg\min \tilde{W}^k(\boldsymbol{\lambda} \mid \gamma, \mu) = \arg\min \left\{ \frac{1}{2} \boldsymbol{\lambda}^T \left(\sum_{i \in \mathbb{I}(\boldsymbol{\lambda}^k)} \boldsymbol{x}_i \boldsymbol{x}_i^T \right) \boldsymbol{\lambda} - \right. \\ \left. \mu^2 \sum_{i \in \mathbb{I}(\boldsymbol{\lambda}^k)} 1 - \sum_{j=1}^{N} y_j \lambda_j \right\}, \quad \sum_{j=1}^{N} \lambda_j = 0, \ 0 \le y_j \lambda_j \le \frac{1}{2\gamma}, \ j = 1, ..., N. \end{cases} \quad (37)$$

The final bias estimate (24) has the form

$$\hat{b} = -\frac{\sum_{i=1}^{n} \left(\hat{a}_i \sum_{j: \, 0 < y_j \hat{\lambda}_j < (1/2\gamma)} y_j \hat{\lambda}_j x_{j,i} \right) + \sum_{j: \, y_j \hat{\lambda}_j = (1/2\gamma)} \hat{\lambda}_j}{\sum_{j: \, 0 < y_j \hat{\lambda}_j < (1/2\gamma)} y_j \hat{\lambda}_j}. \quad (38)$$

This is a standard quadratic programming problem [77] of polynomial computational complexity relative to N.

V. Rough Regularization Path Along the Selectivity Axis

A. Active interval of the selectivity parameter

The selectivity parameter $0 \le \mu < \infty$ is the main hyperparameter of the dependence estimation problem in general (11) and of its dual form in particular (22) or (25)-(26). If $\mu = 0$, the criterions possess no selectivity property at all, and all the estimated components of the direction vector remain active (20).

On the contrary, when the selectivity is large enough $\mu \to \infty$, all the direction vector components become zero. What is the maximal value of selectivity that completely suppresses all the features? We will denote it as μ_0 because it retains 0 active features.

Let us imagine the subset of active features in (26) to be empty:

$$\begin{cases} \sum_{j=1}^{N} \varphi(y_j, \lambda_j \mid \gamma) \to \min(\boldsymbol{\lambda}), \\ \sum_{j=1}^{N} \lambda_j = 0, \quad -\frac{1}{2\gamma} g_{\sup}(y_j) \le \lambda_j \le -\frac{1}{2\gamma} g_{\inf}(y_j). \end{cases} \quad (39)$$

Remember that function $\varphi(y, \lambda \mid \gamma)$ (16) is convex in the respective range.

Let $(\lambda_1^*,...,\lambda_N^*)$ be solutions of the truncated problem (39), and μ_0 be defined as

$$\mu_0 = \max_{i=1,...,n}\left(\sum_{j=1}^{N}\lambda_j^* x_{j,i}\right). \quad (40)$$

Then, if $\mu = \mu_0$, we have $(\hat{\lambda}_1,...,\hat{\lambda}_N) = (\lambda_1^*,...,\lambda_N^*)$ in (22)

$$\begin{cases}(\hat{\lambda}_{1,\mu_0},...,\hat{\lambda}_{N,\mu_0}) = \arg\min W(\lambda_1,...,\lambda_N|\tilde{\mathbf{X}}_\alpha,\mathbf{y},\gamma,\mu_0),\\ \sum_{j=1}^{N}\lambda_j = 0,\ -(1/2\gamma)g_{\sup}(y_j) \le \lambda_j \le -(1/2\gamma)g_{\inf}(y_j),\end{cases} \quad (41)$$

and $\hat{a}_i = 0$ for all $i = 1,...,n$ in (20), i.e. we obtain the trivial empty model.

Thus, the active interval of the selectivity parameter is $0 \le \mu \le \mu_0$.

B. *The idea of the regularization path and its rough implementation*

The number of active features will be growing from 0 to n as μ diminishing from μ_0 to 0. This is just the exact idea of the full regularization path [8,9]. Theoretically, the number of bifurcation points, where the number of active features changes, will not be lesser than n, but in reality it will be much greater than n, because this process is far from being monotonic. As a result, such a procedure would be too time consuming in the case of large number of features.

We consider here a rough implementation of this idea. The experience shows that it is expedient to divide the interval $[10^{-8}\mu_0 \approx 0,\ \mu_0]$ into a number of $m \le n$ subintervals in logarithmic scale $\mu_l = 10^{-8(l/m)}\mu_0$, $l = 0,1,...,m$:

$$\mu_m = 10^{-8}\mu_0 \approx 0 < \mu_{m-1} = 10^{-8((m-1)/m)}\mu_0 < ... < \mu_0 = 10^{-0}\mu_0. \quad (42)$$

The rough regularization path starts with $l = 0$, which corresponds to $\mu = \mu_0$ and the trivial dual problem (41) that yields the empty model $\hat{a}_i = 0$ for all $i = 1,...,n$ (20). Nevertheless, the result of the iteration process $(\hat{\lambda}_{1,\mu_0},...,\hat{\lambda}_{N,\mu_0})$ should be stored.

Each next value of the selectivity parameter $\mu = \mu_l$ will almost coincide with the previous value $\mu = \mu_{l-1}$, and the iteration process (Section IV) started with the previous solution $(\hat{\lambda}_{1,\mu_{l-1}},...,\hat{\lambda}_{N,\mu_{l-1}})$ will converge after only a few iterations, in most cases, after one or two iterations. The number of non-zero components of the direction vector will gradually grow (20).

Finally, at the last step $\mu = \mu_m \approx 0$, we will have the direction vector with almost all active components.

We will see in the next Section that the entire regularization path will take approximately the same computation time as the iteration process for one single value of the selectivity parameter as (28)-(32) in Section IV.A.

VI. Experimental Study of the Computational Complexity of Dependence Estimation with Growing Number of Features

The aim of the study is to experimentally measure the dependence of the processing time of the Newton's iterative algorithm (28)-(32) in Section IV, let it be denoted as T, from the dimension of feature vectors n with emphasized attention to the number of iterations.

A. *SVM pattern recognition – Classification of evoked potentials in Electroencephalograms*

Electroencephalography is a method of testing the electrical activity of the brain by jointly processing several electrical signals registered in parallel at several points on the surface of the skull. It was originally invented and is broadly used as a means to study mechanisms by which human behavior is generated, in particular, for brain diseases diagnosis.

However, in the past decades, electroencephalography has become the basis of many brain-computer interfaces, which decode neural response to different stimuli into commands that, for instance, operate external devices [10].

The experiments we refer to in this paper [11,12] are concerned with another purpose of analyzing responses of a multi-channel electroencephalogram (EEG) to outward stimuli. It is assumed that the person whose EEG is processed is an experienced mammologist able to reliably distinguish between X-ray mammograms of women with breast cancer and those of healthy women. These studies pursue the aim to essentially improve productivity of rare pronounced experts by way of, first, accelerating the screening of mammographic images up to ten pictures per second, and, second, immediately detecting the eventual potentials evoked in the expert's EEG by a target (cancer) image among a crowd of non-target ones before the expert becomes aware of this fact.

In our experiments, we analyzed 66-channel EEG signals registered in parallel at 66 points on the scalp of an expert. Initial EEG signals from each electrode are filtered with a cutoff frequency of 40 Hz, see [11,12] for details.

The diagnostic session of a set of mammograms is organized as follows. The expert is shown a sequence of mammograms at a speed of 10 Images per second, namely, 100 ms per mammogram. The sequence was divided into groups, each of 11 images. There are two kinds of groups called target and non-target ones. A non-target group consists entirely of healthy mammograms, whereas each target group contains exactly one cancer image at a random place surrounded by healthy ones at both sides. Thus, the time duration of each group is 1100 ms. The EEG signal was originally registered at a frequency of 1000 Hz, but we applied 11 times thinning, so, one signal fragment corresponding to one group of mammograms, target or non-target ones, finally consists of 100 samples. Since 66 channels are registered, $n = 6600$ is the entire dimension of the "EEG feature vector" $\mathbf{x}_j = (x_{j,1}\cdots x_{j,n})^T \in \mathbb{R}^n$, which relates to the j-th image group and is built as concatenation of all the 66 channels.

The classification of EEG potentials consists in detection whether the registered EEG signal $\mathbf{x}_j \in \mathbb{R}^n$ is a response to a target image $y_j = 1$ or not target one $y_j = -1$. From the mathematical point of view, this is a two-class pattern recognition problem, which was formulated in [12] as that of selective SVM pattern recognition (8), (15):

$$\begin{cases} F(\mathbf{a},b,z_1,...,z_n|\gamma,\mu) = \gamma\sum_{i=1}^{n}\begin{pmatrix}2\mu|a_i|,\ |a_i|\le\mu\\ \mu^2+a_i^2,\ |a_i|>\mu\end{pmatrix} + \\ \sum_{j=1}^{N} q(y_j,z_j) \to \min\left(\mathbf{a}\in\mathbb{R}^n, b\in\mathbb{R}, \mathbf{z}\in\mathbb{R}^N\right), \\ q(y_j,z_j) = \max\left(0, 1-y_j z_j\right),\ z_j = \mathbf{a}^T\mathbf{x}_j + b,\ \ j = 1,...,N. \end{cases} \quad (43)$$

The raining set is the objects/features table

$$\left(\mathbf{x}_1^n \cdots \mathbf{x}_N^n\right),\ \mathbf{x}_j^n = (x_{j,1}\cdots x_{j,n}) \in \mathbb{R}^n, \quad (44)$$

with the number of features $n = 6600$ and number of objects $N = 196$, so that $n \gg N$. The 6600 EEG features turned out to be tightly correlated – the eigenvalues of the inner product matrix quickly fall:

$\zeta_1 = 16369.8,\ \zeta_2 = 1435.2,\ \zeta_3 = 1076.2,$

$\zeta_{20} = 134.0 < 0.01\zeta_1,\ \zeta_{50} = 32.2 < 0.002\zeta_1.$

B. Chronometry of the learning process on data sets of growing dimensionality with fixed value of the selectivity parameter

Let $i = 1,...,n$ be some natural numeration of features. We transformed the given basic training set (44) into a succession of partial training sets with growing dimensionality of the feature space

$$\left(\mathbf{x}_1^m \cdots \mathbf{x}_N^m\right),\ \mathbf{x}_j^m = (x_{j,1} \cdots x_{j,m}) \in \mathbb{R}^m,\quad m = 1,...,n. \quad (45)$$

Let $0 < \mu < \infty$ be a fixed value of the selectivity parameter that suppresses about three quarters of features, i.e., one fourth of them remains active. For the fixed selectivity, each of these partial training sets defines a succession of convex training criteria

$$J(\mathbf{a}^m, b \mid \gamma, \mu) = \gamma \sum_{i=1}^{m} \begin{pmatrix} 2\mu\,|a_i|, |a_i| \le \mu \\ \mu^2 + a_i^2, |a_i| > \mu \end{pmatrix} + \sum_{j=1}^{N} q\left(y_j, \sum_{i=1}^{m} a_i \tilde{x}_{\alpha,j,i} + b\right) \to \min\left(\mathbf{a}^m \in \mathbb{R}^m, b \in \mathbb{R}\right) \quad (46)$$

as functions of the direction vector of growing dimension $m = 1,...,n$. On the other hand, each of the partial training sets defines the respective succession of dual criteria (22)

$$\begin{cases} W(\lambda_1,...,\lambda_N \mid \gamma, \mu) = \dfrac{1}{2}\sum_{i=1}^{m<n} \left\{ \max\left[0, \left(\sum_{j=1}^{N} \lambda_j x_{j,i}\right)^2 - \mu^2\right]\right\} + \\ \sum_{j=1}^{N} \varphi(y_j, \lambda_j \mid \gamma) \to \min(\lambda_1,...,\lambda_N), \\ \sum_{j=1}^{N} \lambda_j = 0,\quad -\dfrac{1}{2\gamma} g_{\sup}(y_j) \le \lambda_j \le -\dfrac{1}{2\gamma} g_{\inf}(y_j), \end{cases} \quad (47)$$

which are functions of the same fixed number of Lagrange multipliers $\lambda_1,...,\lambda_N$.

First, we applied a standard iterative convex programming procedure available in Matlab to each of the full criteria $J(\mathbf{a}^m, b \mid \mu, \alpha)$ (46), and registered the run time $T_{full}(m)$. Then, we applied the iterative procedure (28)-(32) from Section IV.A to each of dual criteria (47) and registered the run time $T_{dual}(m)$.

The result is shown in Figure 1. As it is seen, the numerical computational complexity of the initial regularized empirical risk minimization problem (23) relative to the number of features remains polynomial and extremely high – the run time T_{fill} swiftly grows as m increases. Numerical solving of this problem in the disjoint formulation (25)-(26) multiply reduces the computational complexity.

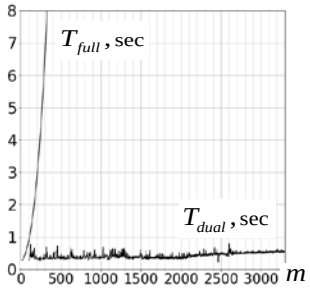

Figure 1. SVM pattern recognition – real-world data.

Chronometry of the learning process with growing number of features $m = 1,...,n$.

VII. Conclusions

We have considered a class of generalized linear models of feature-based dependence estimation from empirical data, which covers, in particular, numerical regression and two-class SVM pattern recognition. Two additional assumptions, which are adequate to the overwhelming majority of practical applications, are that, first, the number of features n far exceeds that of objects in the training set N and, second, the features are tight interdependent, so that the effective dimension of the concentration ellipsoid of feature vectors is essentially smaller than the number of features.

Under some quite lenient assumptions, the traditional formulation of the generalized linear dependence estimation problem results in the convex problem of regularized empirical risk minimization. This problem inevitably has polynomial computational complexity in the number of features, what is in crucial conflict with the assumption on the huge dimension of the feature vectors $n \gg N$.

Therefore, we proposed an alternative disjoint formulation of the generalized linear dependence estimation problem, which allows for its numerical solution in two consecutive stages. First, a convex dual minimization problem of N variables is to be solved, which have the sense of Lagrange multipliers associated with the objects of the training set. Such a problem is of polynomial computational complexity relative to the assumingly modest size of the training set N. After that, it remains only to independently compute the estimates of the coefficients at the features in the generalized linear model of the sought-for dependence. It is clear that this procedure is not only of linear computational complexity in the number of features n, but also easily parallelizable.

References

[1] Vapnik, V. Estimation of Dependences Based on Empirical Data. Springer-Verlag New York, 1982.

[2] Nelder, J, Wedderburn, R. Generalized Linear Models. Journal of the Royal Statistical Society. Series A (General). Vol. 135, Issue 3, 1972, pp. 370-384.

[3] McCullagh, P., Nelder, J. Generalized Linear Models, Second Edition. Chapman and Hall, 1989, 511 p.

[4] Vapnik, V. Statistical Learning Theory. Wiley, 1998.

[5] Tatarchuk, A., Mottl, V., Eliseyev, A., Windridge, D. Selectivity supervision in combining pattern-recognition modalities by feature- and kernel-selective Support Vector Machines. Proceedings of the 19th International Conference on Pattern Recognition ICPR-2008. Vol 1-6, pp. 2336-2339.

[6] Tatarchuk, A., Urlov, E., Mottl, V., Windridge, D. A support kernel machine for supervised selective combining of diverse pattern-recognition modalities. In: Multiple Classifier Systems. Lecture Notes in Computer Science, Vol. 5997. Springer-Verlag, Berlin \ Heidelberg, 2010, pp. 165-174.

[7] Fletcher, R. Practical Methods of Optimization. Wiley, 2000, 450 p.

[8] Park, M. Hastie, T. L1-Regularization path algorithm for generalized linear models. Journal of the Royal Statistical Society, Series B (Statistical Methodology), 2007, Vol. 69, Part 4, pp. 659–677.

[9] Friedman, J., Hastie, T., Tibshirani R. Regularization paths for generalized linear models via coordinate descent. Journ. of Stat. Soft, 2010, Vol. 33, Is. 1, pp. 1-22.

[10] Bucolo, M., Fortuna, L., Frasca, M. Robot control through brain-computer interface for pattern generation. Complex Systems, 2012, Vol. 20, Is. 3, pp. 243-251.

[11] Hope, C., Sterr, A., Langovan, P.E., Geades, N., Windridge, D., Young, K., Wells, K. High Throughput Screening for Mammography using a Human-Computer Interface with Rapid Serial Visual Presentation (RSVP). Proc. of SPIE 8673, Med. Imaging 2013: Image Perception, Observer Performance, and Technology Assessment, 867303, 28 March 2013, 8 p.

[12] Sulimova, V., Bukhonov, S., Krasotkina, O., Mottl, V., Windridge, D. Regularized SVMs for classification of image evoked EEG potentials captured from an observer. Submitted to SIBIRCON-2019.