Hindawi Publishing Corporation
Journal of Optimization
Volume 2016, Article ID 2659012, 7 pages
http://dx.doi.org/10.1155/2016/2659012

Research Article

Evidence Maximization Technique for Training of Elastic Nets

Igor Dubnov,[1] Alexander Merkov,[2] Vladimir Arlazarov,[1,2] and Ilia Nikolaev[3]

[1] *Moscow Institute of Physics and Technology, Moscow 141700, Russia*
[2] *Institute for Systems Analysis, Russian Academy of Sciences, Prospekt 60-Let Octyabria 9, Moscow 117312, Russia*
[3] *MV Lomonosov Moscow State University, Leninskie Gory 1, Moscow 119991, Russia*

Correspondence should be addressed to Igor Dubnov; dubnov@phystech.edu

Received 15 February 2016; Revised 10 May 2016; Accepted 15 May 2016

Academic Editor: Manlio Gaudioso

This paper presents a technique of evidence maximization for automatic tuning of regularization parameters of elastic nets, which allows tuning many parameters simultaneously. This technique was applied to handwritten digit recognition. Experiments showed its ability to train either models with high accuracy of recognition or highly sparse models with reasonable accuracy.

1. Introduction

One of the important aspects of machine learning is to choose an appropriate subset of the (possibly huge) set of all virtually available features, such as the trained model which depends only on this subset of features. A good choice (*feature selection*, [1]) can both speed up the training and improve the quality of its result. It depends not only on the particular problem, but also on the data available for training.

Feature selection can either precede the learning itself (e.g., entropy-based or correlation analysis) or be a built-in part of the learning process (e.g., learning with l_1-regularization, such as LASSO regression and l_1-SVM) [2]. This paper deals with the latter case only.

It is known that learning with l_1-regularization can produce rather sparse models which depend on rather few features, but learning with l_2-regularization usually produces more accurate models. In [3] some mixed regularization called "*elastic net*" was proposed. Let $F(x, w)$ be a model parameterized by w, predicting response y by feature vector x, and let $J(F(x, w), y)$ be the cost of prediction $F(x, w)$ provided the true response is y. Then training of such a model with elastic net regularization on the set of samples $\{(x_i, y_i), i = 1, \ldots, N\}$ using loss minimization (a.k.a. ERM—empirical risk minimization) method or, briefly, "training of an elastic net" is the minimization problem:

$$\sum_{i=1}^{N} J\left(F\left(x_i, w\right), y_i\right) + \lambda |w| + \frac{\mu}{2} \|w\|^2 \longrightarrow \min_{w}, \qquad (1)$$

where $\|\cdot\|$ and $|\cdot|$ stand for l_2- and l_1-norms, respectively, and λ and μ are nonnegative regularization parameters. It is shown experimentally in [3] that varying the parameters λ and μ one can balance between the sparsity of the model and the accuracy of its prediction.

In this paper elastic nets are used to regularize multiclass logistic regression. A method of tuning more general regularization parameters than λ and μ above is described. This method is tested on a handwritten digit recognition problem.

The rest of this paper is organized as follows. Section 2 presents the mathematical model and the elastic net in details. Section 3 describes the learning algorithm and the evidence maximization technique for tuning regularization parameters of the elastic net; this technique is the main subject of this paper. Section 4 describes experiments with elastic nets for digit recognition. Section 5 exposes the results of experiments. Section 6 summarizes the main results of experiments and discusses further possible applications of the proposed technique.

2. Mathematical Model

Consider multinomial classification in its both deterministic and probabilistic variants: given a feature vector $x \in \mathbb{R}^d$ either to predict the correct label y of one of q classes to which the vector x belongs or to estimate the conditional probability $p(y \mid x)$ of each class label. Probabilistic classification is considered primary and in deterministic classification a class label (usually *the* class label) $y \in \operatorname{argmax}_y p(y \mid x)$ will be predicted.

Let $\vec{x} = (1, x) \in \mathbb{R}^{d+1}$ stand for *augmented feature vector*. To estimate $p(y \mid x)$ multinomial linear logistic regression model

$$p(y \mid x, \vec{w}) = \frac{e^{\vec{w}^y \vec{x}}}{\sum_{l=1}^{q} e^{\vec{w}^l \vec{x}}} \tag{2}$$

will be trained. The model parameter matrix $\vec{w}$ consists of $q(d+1)$-dimensional rows $\vec{w}^l = (w_0^l, w_1^l, \ldots, w_d^l)$. To train the model means to choose some "good" parameter $\vec{w}$.

To do this we use a training dataset $\mathbf{T} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ of N couples (x_i, y_i) which are supposed to be i.i.d. random. $\mathbf{T}$ can also be written in a transposed way $\mathbf{T} = \{\mathbf{X}, \mathbf{Y}\}$ where $\mathbf{X} = \{x_1, \ldots, x_N\}$ and $\mathbf{Y} = \{y_1, \ldots, y_N\}$. Training tries to maximize the posterior of $\vec{w}$ given some prior $p_0(\vec{w})$ and the training set $\mathbf{T}$. Since

$$p(\vec{w} \mid \mathbf{T}) = \frac{p_0(\vec{w})\, p(\mathbf{T} \mid \vec{w})}{p(\mathbf{T})} = \frac{p_0(\vec{w})\, p(\mathbf{Y} \mid \mathbf{X}, \vec{w})}{p(\mathbf{Y} \mid \mathbf{X})} \tag{3}$$

and the denominator does not depend on $\vec{w}$, maximization of posterior probability is equivalent to maximization of the numerator or of its logarithm:

$$\begin{aligned} &\ln\left(p_0(\vec{w})\, p(\mathbf{Y} \mid \mathbf{X}, \vec{w})\right) \\ &\quad = \ln\left(p_0(\vec{w})\right) + \sum_{i=1}^{N} \ln p(y_i \mid x_i, \vec{w}) \longrightarrow \max_{\vec{w}}. \end{aligned} \tag{4}$$

The second summand in (4) is the log likelihood of the model $L(\vec{w}; \mathbf{T})$, while the first one depends on the choice of the prior.

Let $(q \times d)$-matrix w stand for $\vec{w}$ without the bias column w_0. The prior is usually taken independent of the bias, so $p_0(\vec{w}) = p_0(w)$. In the simplest cases when spherical Gaussian or Laplacian distributions are taken as priors, training (4) turns to an optimization problem with l_2- or l_1-regularization, respectively.

Similarly, elastic nets are obtained from the prior

$$p_0(\vec{w}) = \frac{1}{Z(\lambda, \mu)} e^{-\lambda|w| - \mu(\|w\|^2/2)}, \tag{5}$$

where

$$\begin{aligned} Z(\lambda, \mu) &= \int e^{-\lambda|w| - \mu(\|w\|^2/2)} dw \\ &= \left(\int_{-\infty}^{\infty} e^{-\lambda|t| - \mu(t^2/2)} dt\right)^{qd} \\ &= \left(\frac{2e^{\lambda^2/2\mu}}{\sqrt{\mu}} \int_{\lambda/\sqrt{\mu}}^{\infty} e^{-\tau^2/2} d\tau\right)^{qd} \\ &= \left(\frac{2e^{\lambda^2/2\mu}}{\sqrt{\mu}} \sqrt{2\pi}\, \Phi\left(-\frac{\lambda}{\sqrt{\mu}}\right)\right)^{qd} \end{aligned} \tag{6}$$

(remember that the space of w is qd-dimensional) and $\Phi(\cdot)$ denotes the cumulative function of the standard one-dimensional Gaussian distribution:

$$\Phi(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-\tau^2/2} d\tau. \tag{7}$$

To simplify calculations instead of the function $\Phi(\cdot)$ we use

$$\Psi(t) = e^{t^2/2} \int_{-\infty}^{t} e^{-\tau^2/2} d\tau = \sqrt{2\pi} e^{t^2/2} \Phi(t). \tag{8}$$

For instance, the normalization factor $Z(\lambda, \mu)$ becomes

$$Z(\lambda, \mu) = \left(\frac{2}{\sqrt{\mu}} \Psi\left(-\frac{\lambda}{\sqrt{\mu}}\right)\right)^{qd}. \tag{9}$$

Plugging (5) into (4) turns training of elastic net into the optimization problem:

$$-\sum_{i=1}^{N} \ln p(y_i \mid x_i \cdot \vec{w}) + \lambda|w| + \frac{\mu}{2}\|w\|^2 \longrightarrow \min_{\vec{w}}. \tag{10}$$

Both prior (5) and regularization summands in (10) are isotropic with respect to all d features. However the features themselves might be unequal by their nature. To respect such an inequality we partition all features into K groups of features of the same nature. For example, all pixel values of the image have the same nature and will belong to the same group of features, while computed features or the aspect ratio falls to other groups.

Let us fix a partition of the set of indices

$$\{1, \ldots, d\} = \bigsqcup_{k=1}^{K} D_k \tag{11}$$

into subsets D_k of cardinalities $d_k = \# D_k$ and define separate regularization parameters λ_k and μ_k for each group. Then training of generic elastic net (10) turns into

$$\begin{aligned} &-\sum_{i=1}^{N} \ln p(y_i \mid x_i \cdot \vec{w}) \\ &\quad + \sum_{k=1}^{K} \left(\lambda_k \sum_{j \in D_k} \left|w_j\right| + \frac{\mu_k}{2} \sum_{j \in D_k} \left\|w_j\right\|^2\right) \longrightarrow \min_{\vec{w}}, \end{aligned} \tag{12}$$

and training of the elastic net for linear logistic regression (2) turns into

$$\begin{aligned} &-\sum_{i=1}^{N} \left(w^{y_i} \vec{x}_i - \ln \sum_{l=1}^{q} e^{w^l \vec{x}_i}\right) \\ &\quad + \sum_{k=1}^{K} \left(\lambda_k \sum_{j \in D_k} \left|w_j\right| + \frac{\mu_k}{2} \sum_{j \in D_k} \left\|w_j\right\|^2\right) \longrightarrow \min_{\vec{w}}. \end{aligned} \tag{13}$$

It is easy to see that optimization problem (13) is convex for any training set $\mathbf{T}$ and nonnegative λ_k and μ_k. Choice of values of $2K$ regularization parameters λ_k and μ_k, which is the subject of this paper, will be discussed later in Section 3.2.

3. Learning Technique

3.1. Nonsmooth Convex Optimization. Standard gradient methods are not applicable to minimization problems (10) and (13) because they contain nonsmooth terms $|w|$ and $|w_j|$. So the algorithm proposed by Nesterov in [4] for minimization of sums of smooth and simple nonsmooth convex functions is used. Nesterov's algorithm provides the best convergence rate at moderate number of steps (less than the number of variables, which is equal to $q(d+1)$ in (10) and (13)) among all known methods of nonsmooth optimization [5].

Nesterov's algorithm can exploit strong convexity (μ-convexity) of the target function and converges the faster, the bigger μ can be guaranteed in advance. The target function in (13) is not strongly convex in the bias column w_0, but it would be strongly convex if l_2-regularization was applied to all parameters $\vec{w}$ including w_0.

Consider the following modification of problem (13).

(1) Estimate the bias column $\widehat{w}_0$:

$$\widehat{w}_0^l = \ln\frac{n_l}{N} \quad \text{for } l = 1, \ldots, q, \tag{14}$$

where n_l is the number of training samples of class l. The estimate $\widehat{w}_0$ is the solution of minimization problem:

$$-\sum_{i=1}^{N}\ln p\left(y_i \mid w_0\right) = -\sum_{i=1}^{N}\ln\frac{e^{w_0^{y_i}}}{\sum_{l=1}^{q}e^{w_0^l}} \longrightarrow \min_{w_0}, \tag{15}$$

which is nothing but maximum likelihood training of the featureless logistic regression model.

(2) Choose some $\mu_0 > 0$ and instead of (13) solve

$$\begin{aligned}&-\sum_{i=1}^{N}\left(\vec{w}^{y_i}\vec{x}_i - \ln\sum_{l=1}^{q}e^{\vec{w}^l\vec{x}_i}\right) + \frac{\mu_0}{2}\left\|w_0 - \widehat{w}_0\right\|^2\\ &\quad + \sum_{k=1}^{K}\left(\lambda_k\sum_{j\in D_k}\left|w_j\right| + \frac{\mu_k}{2}\sum_{j\in D_k}\left\|w_j\right\|^2\right) \longrightarrow \min_{\vec{w}}.\end{aligned} \tag{16}$$

The target function in (16) is strongly convex with nonnegative parameter $\mu = \min_{k=0,1,\ldots,K}\mu_k$.

3.2. Evidence Maximization. To train elastic nets (10), (13), or (16) successfully some reasonable values of regularization parameters λ and μ (hyperparameters) are required. In machine learning problems with one or at most two hyperparameters (e.g., in SVM [1]) their values can be found by grid search. However, there are $2K+1$ hyperparameters in generalized elastic net (16) and we are interested in the case $K > 1$. In this case, a reasonable way to optimize them is evidence maximization. The use of evidence maximization for estimation of hyperparameters of ridge regression and other Gaussian-based models is well known [6]. For non-Gaussian elastic nets the evidence of hyperparameters can be neither computed nor maximized exactly and will be approximated rather roughly.

Let prior $p_0(\vec{w})$ depend on two hyperparameters λ and μ like in (5). Then posterior (3) with λ and μ indicated explicitly is

$$\begin{aligned}p\left(\vec{w} \mid \mathbf{T}, \lambda, \mu\right) &= \frac{p\left(\mathbf{T}, \vec{w}\right)p_0\left(w \mid \lambda, \mu\right)}{p\left(\mathbf{T} \mid \lambda, \mu\right)}\\ &= \frac{p\left(\mathbf{Y} \mid \mathbf{X}, \vec{w}\right)p_0\left(w \mid \lambda, \mu\right)}{p\left(\mathbf{Y} \mid \mathbf{X}, \lambda, \mu\right)}\\ &= \frac{L\left(\vec{w}; \mathbf{T}\right)p_0\left(w \mid \lambda, \mu\right)}{\int L\left(\vec{w}; \mathbf{T}\right)p_0\left(w \mid \lambda, \mu\right)d\vec{w}}\\ &= \frac{L\left(\vec{w}; \mathbf{T}\right)p_0\left(w \mid \lambda, \mu\right)}{E\left(\lambda, \mu; \mathbf{T}\right)}.\end{aligned} \tag{17}$$

The denominator is ignored in maximization of posterior (3) because it does not depend on $\vec{w}$. However it depends on λ and μ. This denominator $E(\lambda, \mu; \mathbf{T})$ is called the evidence of parameters λ and μ with respect to the training set $\mathbf{T}$. Despite its special name, it is a usual likelihood, not the likelihood of a single model like $L(\vec{w}; \mathbf{T})$ in (4), but the likelihood of the whole probability space of models defined by hyperparameters λ and μ.

For prior (5) the evidence of pair (λ, μ) is

$$\begin{aligned}E\left(\lambda, \mu; \mathbf{T}\right) &= \int L\left(\vec{w}; \mathbf{T}\right)p_0\left(w \mid \lambda, \mu\right)d\vec{w}\\ &= \frac{1}{Z\left(\lambda, \mu\right)}\int e^{\ln L(\vec{w};\mathbf{T})-\lambda|w|-\mu(\|w\|^2/2)}d\vec{w}\end{aligned} \tag{18}$$

and the evidence maximization is equivalent to minimization

$$\begin{aligned}-\ln E\left(\lambda, \mu; \mathbf{T}\right) &= qd\ln\left(\frac{2}{\sqrt{\mu}}\Psi\left(-\frac{\lambda}{\sqrt{\mu}}\right)\right)\\ &\quad - \ln\int e^{\ln L(\vec{w};\mathbf{T})-\lambda|w|-\mu(\|w\|^2/2)}d\vec{w}\\ &\quad \longrightarrow \min_{\lambda,\mu}.\end{aligned} \tag{19}$$

The normalization factor $Z(\lambda, \mu)$ is rewritten using formula (9) here.

The gradient of (19) is

$$\begin{aligned}\nabla_\lambda\left(-\ln E\left(\lambda, \mu; \mathbf{T}\right)\right) &= -\frac{qd}{\lambda}\left(\frac{\lambda/\sqrt{\mu}}{\Psi\left(-\lambda/\sqrt{\mu}\right)} - \frac{\lambda^2}{\mu}\right)\\ &\quad + \mathbf{E}_{\lambda,\mu}\left[|w|\right],\\ \nabla_\mu\left(-\ln E\left(\lambda, \mu; \mathbf{T}\right)\right) &= -\frac{qd}{2\mu}\left(1 - \frac{\lambda/\sqrt{\mu}}{\Psi\left(-\lambda/\sqrt{\mu}\right)} + \frac{\lambda^2}{\mu}\right)\\ &\quad + \frac{1}{2}\mathbf{E}_{\lambda,\mu}\left[\|w\|^2\right],\end{aligned} \tag{20}$$

where $\mathbf{E}_{\lambda,\mu}[f]$ stands for the expectation of $f(w)$ with respect to posterior distribution of w proportional to $L(\vec{w}; \mathbf{T})p_0(w \mid \lambda, \mu)$:

$$\mathbf{E}_{\lambda,\mu}\left[f\right] = \frac{\int f\left(w\right)e^{\ln L(\vec{w};\mathbf{T})-\lambda|w|-\mu(\|w\|^2/2)}d\vec{w}}{\int e^{\ln L(\vec{w};\mathbf{T})-\lambda|w|-\mu(\|w\|^2/2)}d\vec{w}}. \tag{21}$$

To minimize (19) instead of traditional gradient steps the transformation

$$\begin{aligned} \lambda &\longleftarrow \frac{qd\left(\lambda/\sqrt{\mu}/\Psi\left(-\lambda/\sqrt{\mu}\right)-\lambda^2/\mu\right)}{\mathbf{E}_{\lambda,\mu}\left[|w|\right]}, \\ \mu &\longleftarrow \frac{qd\left(1-\lambda/\sqrt{\mu}/\Psi\left(-\lambda/\sqrt{\mu}\right)+\lambda^2/\mu\right)}{\mathbf{E}_{\lambda,\mu}\left[\|w\|^2\right]} \end{aligned} \tag{22}$$

is used iteratively.

Formulas (20) imply that each point of maximum of the evidence is a fixed point of transformation (22). No convergence of transformation (22) is guaranteed. But in the experiments several iterations of this transformation allowed training more accurate model.

For modified elastic net (16), transformation (22) turns into

$$\begin{aligned} \lambda_k &\longleftarrow \frac{qd_k\left(\lambda_k/\sqrt{\mu_k}/\Psi\left(-\lambda_k/\sqrt{\mu_k}\right)-\lambda_k^2/\mu_k\right)}{\sum_{j\in D_k}\sum_{l=1}^{q}\mathbf{E}_{\lambda,\mu}\left[\left|w_j^l\right|\right]}, \\ \mu_k &\longleftarrow \frac{qd_k\left(1-\lambda_k/\sqrt{\mu_k}/\Psi\left(-\lambda_k/\sqrt{\mu_k}\right)+\lambda_k^2/\mu_k\right)}{\sum_{j\in D_k}\sum_{l=1}^{q}\mathbf{E}_{\lambda,\mu}\left[\left\|w_j^l\right\|^2\right]} \end{aligned} \tag{23}$$

for $k = 1, \ldots, K$ and

$$\mu_0 \longleftarrow \frac{q}{\sum_{l=1}^{q}\mathbf{E}_{\lambda,\mu}\left[\left\|w_0^l-\widehat{w}_0^l\right\|^2\right]}. \tag{24}$$

Expectations $\mathbf{E}_{\lambda,\mu}[|w_j^l|]$, $\mathbf{E}_{\lambda,\mu}[\|w_j^l\|^2]$, and $\mathbf{E}_{\lambda,\mu}[\|w_0^l-\widehat{w}_0^l\|^2]$ cannot be computed exactly because posterior $p(\vec{w} \mid \lambda, \mu)$ is rather complicated and high-dimensional. They are estimated using diagonal Laplace approximation [7] of posterior $p(\vec{w} \mid \lambda, \mu)$ at trained model (16) $w^* = w^*(\lambda, \mu)$ instead of the $p(\vec{w} \mid \lambda, \mu)$ itself.

3.3. Stopping Criterion. To stop either training (16) with fixed regularization parameters (λ, μ) or iterations of transformations (23) and (24) of (λ, μ), the following validation technique is used. The available dataset $\mathbf{T}$ is partitioned into training set $\mathbf{T}_{\text{train}}$ of N_{train} samples and validation set $\mathbf{T}_{\text{val}}$ of N_{val} samples. The first one is used to train elastic nets (16) while the second one is used to decide whether further training becomes senseless and should be stopped. Namely, training of the elastic net is stopped if likelihood $L(\vec{w}; \mathbf{T}_{\text{val}})$ has not increased after several (about 30) last optimization steps, and tuning of the regularization parameters (λ, μ) is stopped if likelihood $L(w^*(\lambda, \mu); \mathbf{T}_{\text{val}})$ of the trained model has not increased after several (about 5) last iterations.

This criterion is a kind of well-known early stopping method [8]. On one hand, such an early stopping speeds up the training significantly. On the other hand, it is a regularization technique [9] by itself and can hide the effect of tuning the regularization parameters via evidence maximization, which is the subject of the study here. To find a balance, the delays between nonincreasing of the validation likelihood and stopping were chosen empirically.

4. Experiments

The method described in Sections 2 and 3 was applied to recognition of handwritten digits from MNIST database (see [10]). This database contains grayscale raster images of 28 × 28 = 784 pixels each, which belong to one of $q = 10$ classes. Traditionally it is partitioned into $N = 60000$ samples for training and $M = 10000$ for testing. 15% of training samples were left out for validation, so $N_{\text{train}} = 51000$ and $N_{\text{val}} = 9000$.

Both to make linear logistic regression more powerful and to test the proposed method of estimation of numerous regularization parameters more features were added to the model. Besides the 784 primary features (the pixel intensities) several groups of secondary features were generated. Then all the features, both secondary and primary, were normalized to zero mean and unit variance.

The following groups of secondary features were used in experiments.

(1) Horizontal and vertical components of the gradient of the pixel intensity (784 + 784 = 1568 features).

(2) Amplitudes and phases of the discrete Fourier transform [11] of the pixel intensity (784 + 784 = 1568 features).

(3) Projection histograms [11], that is, the number of nonzero pixels and positions of the first and the last one within each row and each column of the image (28 + 28 + 28 ∗ 2 + 28 ∗ 2 = 168 features).

(4) The corner metric matrix of the image, which for each pixel of the image contains the estimated "likelihood" to be its corner point. The corner metric matrix is calculated by MATLAB function *cornermetric* [12] (784 features).

(5) The local standard deviation matrix, which for each pixel of the image contains the standard deviation of the intensity over 9-by-9 neighborhood of the pixel. The local standard deviation is calculated by MATLAB function *stdfilt* [12] (784 features).

This amounts to $d = 5656$ primary and secondary features in total.

Remember that the proposed learning technique consists of two levels: the inner level is training of elastic net (16) with fixed regularization parameters (λ, μ) using Nesterov's optimization algorithm and the outer level inspired by maximum evidence principle is iterative transformations (23) and (24) of λ and μ. Several different partitions (11) of features into groups were tried.

Each line in Tables 1, 2, and 3 represents single experiment for training of elastic net. Each row of the table represents elastic net (16) trained with some λ and μ. Each experiment was repeated for 20 times. Estimated intervals of the measured values, shown in tables, are intervals of two standard deviations around the mean.

Table 1: Elastic nets trained with several fixed regularization parameters λ and μ.

λ	μ	Sparseness (%)	Mean log likelihood	Error (%)
0	0	2.46 ± 0.00	0.0638 ± 0.0007	2.06 ± 0.05
3		10.32 ± 1.56	0.0583 ± 0.0011	1.85 ± 0.05
10		16.84 ± 2.49	0.0609 ± 0.0007	1.81 ± 0.06
30		45.87 ± 4.03	0.0823 ± 0.0005	2.18 ± 0.05
100		63.26 ± 2.90	0.1419 ± 0.0003	3.41 ± 0.05
300		75.77 ± 4.37	0.2503 ± 0.0004	5.19 ± 0.05
1	1	3.75 ± 0.16	0.0621 ± 0.0007	2.00 ± 0.04
1	10	3.81 ± 0.19	0.0621 ± 0.0007	2.00 ± 0.04
1	30	6.71 ± 2.54	0.0607 ± 0.0013	1.95 ± 0.07
10	1	16.73 ± 2.40	0.0609 ± 0.0007	1.81 ± 0.06
10	10	16.56 ± 2.46	0.0613 ± 0.0007	1.81 ± 0.06
10	30	16.25 ± 2.53	0.0621 ± 0.0006	1.82 ± 0.05
10	100	16.20 ± 3.09	0.0649 ± 0.0005	1.86 ± 0.05
30	100	38.42 ± 2.42	0.0862 ± 0.0005	2.20 ± 0.05
100	30	61.51 ± 2.45	0.1428 ± 0.0003	3.41 ± 0.05
100	100	59.18 ± 2.09	0.1445 ± 0.0003	3.41 ± 0.05
0	1	2.46 ± 0.00	0.0638 ± 0.0007	2.06 ± 0.05
	10	2.46 ± 0.00	0.0638 ± 0.0007	2.06 ± 0.04
	100	2.46 ± 0.00	0.0638 ± 0.0007	2.05 ± 0.04
	300	2.46 ± 0.00	0.0659 ± 0.0006	2.05 ± 0.06

Table 2: Elastic nets trained with the evidence maximization technique.

K	Sparseness (%)	Mean log likelihood	Error (%)
1	12.53 ± 3.18	0.0580 ± 0.0010	1.83 ± 0.06
8	9.99 ± 1.40	0.0557 ± 0.0008	1.70 ± 0.05
13	9.54 ± 1.22	0.0560 ± 0.0010	1.69 ± 0.04
40	10.17 ± 1.40	0.0555 ± 0.0007	1.71 ± 0.05
136	8.74 ± 1.38	0.0581 ± 0.0006	1.81 ± 0.04
385	8.05 ± 1.28	0.0587 ± 0.0004	1.82 ± 0.04
1456	8.35 ± 1.53	0.0581 ± 0.0005	1.81 ± 0.04
5656	10.82 ± 2.68	0.0582 ± 0.0010	1.80 ± 0.06

Tables 1, 2, and 3 contain the following three columns of properties of trained models.

Sparseness. It is the share of features unused in the model that is $\#\{j \geq 1 \mid \vec{w}_j = 0\}/d$.

Mean Log Likelihood. It is the mean over the M-element test set of *minus* logarithm of the predicted probability of the true class label of the sample, $(1/M)\sum_{i=1}^{M}(-\ln(p(y_i \mid x_i, \vec{w})))$.

Error. It is the misclassification rate measured on the same M-element test set, provided the most probable class is predicted, $(1/M)\#\{i \mid p(y_i \mid x_i, \vec{w}) = \max_{1\leq l\leq q} p(l \mid x_i, \vec{w})\}$.

Sparseness of the trained model appears due to l_1-regularization in the elastic net and increases with λ.

Table 3: Sparse elastic nets trained with the evidence maximization technique.

K	Sparseness (%)	Mean log likelihood	Error (%)
1	56.72 ± 6.94	0.0916 ± 0.0027	2.32 ± 0.08
8	75.04 ± 0.86	0.0898 ± 0.0026	2.70 ± 0.08
13	75.56 ± 1.43	0.0960 ± 0.0029	2.97 ± 0.08
40	84.58 ± 1.21	0.1054 ± 0.0029	3.13 ± 0.14
136	85.41 ± 0.64	0.0816 ± 0.0011	2.32 ± 0.06
385	87.55 ± 1.89	0.0804 ± 0.0032	2.47 ± 0.09
1456	85.72 ± 0.57	0.0745 ± 0.0008	2.28 ± 0.05
5656 (= d)	88.62 ± 0.50	0.0739 ± 0.0009	2.28 ± 0.04

4.1. Constant Regularization Parameters.

4.1. Constant Regularization Parameters. First, several control experiments with fixed scalar values of regularization parameters λ and μ were performed. Their results are shown in Table 1.

The minimal average test error 1,81% was achieved with parameters $\lambda = 10$ and $\mu = 1$.

4.2. Tuning Regularization Parameters by Evidence Maximization. Next, experiments with automatic tuning of regularization parameters λ and μ were performed. Since all features had been normalized, the learning was started from $\lambda_k^0 = 1$ and $\mu_k^0 = 1$ for all $k = 1, \ldots, K$. The results are shown in Table 2. Each row represents the elastic net obtained by the described two-level learning process for certain partition (11) of features.

Several different partition schemes were tested.

> $K = 1$, trivial partition: all features belong to the same group.
>
> $K = 8$, rough partition: primary features, horizontal and vertical components of the gradients, amplitudes and phases of the Fourier transform, and three other types of secondary features each form a separate group.
>
> $K = 13, 40, 136, 385, 1456$: the whole image ($28 \times 28$ pixels) is split into $k \times k$ equal squares and, roughly speaking, the groups are formed by features of certain type calculated for certain squares. The exceptions are projection histograms calculated not for squares, but for rows or columns of squares (k groups for each of 6 histograms) and amplitudes and phases of the Fourier transform, both partitioned into $k \times k$ equal squares in the frequency space. So the total number of groups of the partition is equal to $7k^2 + 6k$. For $k = 1, 2, 4, 7, 14$ this gives $K = 13, 40, 136, 385, 1456$.
>
> $K = 5656 = d$, fine partition: each feature forms a separate group.

These experiments show that the evidence maximization technique allows one to obtain more accurate elastic nets than elastic nets with guessed scalar regularization parameters. Indeed, compare the last column of Table 1 with lines $K = 8$, 13, and 40 of Table 2. These lines represent elastic nets trained with certain values of 17-, 27-, and 81-dimensional regularization parameters, which can hardly be guessed.

4.3. Sparse Elastic Net. Last, we performed a series of experiments trying to train very sparse but reasonably accurate models. Sparseness of the model trained with elastic net depends mostly on its parameter(s) λ or λ_k. In the described technique these parameters are tuned in order to get elastic nets with higher evidence. However, experiments show that iterations of the transformations (23) and (24) with the stopping criterion of Section 3.3 tend to stop before they reach any (local!) maximum of the evidence, and where they stop depends on the initial parameters λ_k^0 and μ_k^0.

Experiments of Section 4.2 (Table 2) started from $\lambda_k^0 = 1$ and $\mu_k^0 = 1$ for all k. Then sparseness was low but the trained models made more accurate predictions. If $\lambda_k > \lambda_k^{\max} = \max_{j \in D_k, l=1,\dots,q} |\partial \ln L(0; \mathbf{T}_{\text{train}}) / \partial w_j^l|$, optimization problem (16) has unique solution $w = 0$, the most sparse one, but not accurate. Starting iterations from $\lambda_k^0 = \lambda_k^{\max}$ allows one to get sparse elastic net with reasonable accuracy.

Table 3 shows the results of training elastic net with starting parameters $\lambda_k^0 = \lambda_k^{\max}$, $k = 1, \dots, K$. These results are discussed in the following section.

5. Results and Discussion

5.1. Accuracy of the Trained Model. The best model trained with the evidence maximization technique shown in Table 2 has 1,69% average test error, which is significantly less than 1,81% obtained by guessing of scalar regularization parameters (Table 1). In our experiments each learning with evidence maximization took only 5–10 reestimations of the regularization parameters. So the numbers of elastic nets trained to fill in Tables 1 and 2 are comparable (moreover, not all guesses are shown in Table 1).

The evidence maximization technique allows one to guess only an appropriate partitioning of the features instead of particularly good values of the regularization parameters. Still, this technique is not fully automated. None of the two obvious extreme partitions (the roughest and the finest ones) leads to the best model. 1,83% in the first line of Table 2 compared to 1,81% achieved in Table 1 shows that the evidence maximization not necessarily leads to the best accuracy. But it can be used when regularization parameters are multidimensional and naive attempts to guess a good value of them are unfeasible.

The obtained accuracy is much lower than best state-of-the-art results obtained by convolutional neural networks, deep learning, and augmentation of training dataset. But the elastic net with precisely tuned regularization parameters can achieve higher accuracy than other traditional models of the same complexity (e.g., 1- or 2-layer neural networks or SVM with Gaussian kernel) (see [10]).

5.2. Sparseness of the Trained Model. In some practical classification problems high sparseness of the model takes priority over its high accuracy. The proposed method allows one to train models with various tradeoff between sparsity and accuracy.

The last elastic net shown in Table 3 provides test error 2,28% and sparseness 88,62%, so only 644 of 5656 features are used. Compared to the most accurate elastic net from Table 2, the error increased by 0,59%, while the number of used features decreased more than sevenfold, from 5116 to 644. This result was achieved by tuning individual regularization parameters for each feature starting from the biggest reasonable λ_k^0.

6. Conclusion

This paper describes a method of machine learning based on a technique of adjusting of regularization parameters of elastic nets inspired by evidence maximization principle. The method is able to cope with multidimensional regularization parameters using only rough simple ideas about their initial values and about the nature of the features used in the models to be learned.

This method was tested on MNIST database of handwritten digits and allowed training more accurate elastic net than could be trained with traditional grid search of one or two scalar regularization parameters. It allowed also training very sparse models with reasonable accuracy.

Still the primary goal of the proposed method of learning lies beyond the scope of this paper. It is to develop a mechanism of feature selection based on training of elastic nets with controlled tradeoff between their sparseness and accuracy. In future the proposed method is going to be applied to other machine learning problems, including problems with very large number of features.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was partially supported by the Russian Foundation for Basic Research Grants no. 15-29-06081 “ofi-m” and no. 16-07-00616 “A.”

References

[1] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 103 of *Springer Texts in Statistics*, Springer, New York, NY, USA, 2013.

[2] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[3] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, vol. 67, no. 2, pp. 301–320, 2005.

[4] Y. Nesterov, *Gradient Methods for Minimizing Composite Objective Function*, 2007.

[5] P. Richtarik and M. Schmidt, “Modern convex optimization methods for large-scale empirical risk minimization,” in *Proceedings of the International Conference on Machine Learning*, July 2015.

[6] C. M. Bishop, *Pattern recognition and machine learning*, Information Science and Statistics, Springer, New York, NY, USA, 2006.

[7] A. I. Prilepko and D. Ph. Kalinichenko, *Asymptotic Methods and Special Functions*, MIPI, 1980.

[8] Y. Yao, L. Rosasco, and A. Caponnetto, “On early stopping in gradient descent learning,” *Constructive Approximation*, vol. 26, no. 2, pp. 289–315, 2007.

[9] D. F. Morgado, A. Antunes, and A. M. Mota, “Regularization versus early stopping: a case study with a real system,” in *Proceedings of the 2nd IFAC Conference Control Systems Design*, Bratislava, Slovakia, 2003.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[11] Ø. D. Trier, A. K. Jain, and T. Taxt, “Feature extraction methods for character recognition—a survey,” *Pattern Recognition*, vol. 29, no. 4, pp. 641–662, 1996.

[12] The MathWorks Inc, MATLAB Image Processing Toolbox documentation, http://www.mathworks.com/help/images/.

Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
Algebra
Journal of Probability and Statistics
The Scientific World Journal
International Journal of Differential Equations
International Journal of Combinatorics
Advances in Mathematical Physics
Journal of Complex Analysis
Journal of Mathematics
Mathematical Problems in Engineering
Abstract and Applied Analysis
Discrete Dynamics in Nature and Society
International Journal of Mathematics and Mathematical Sciences
Journal of Discrete Mathematics
Journal of Function Spaces
International Journal of Stochastic Analysis
Journal of Optimization