Annals of the University of Bucharest (mathematical series)
2 (LX) (2011), 179–206

Multivariate linear systems for learning from data

Iuliana Paraschiv-Munteanu

Communicated by Ioan Tomescu

Abstract - Learning from data means a process of information extraction from finite size samples in order to estimate an unknown dependency. A series of problems can be modeled in terms of a system that computes according to an unknown rule a response (output) for each input. The paper provides a series of results concerning the learning from data a linear regressive model in a multivariate framework. The parameter estimates of the regressive model are determined using the maximum likelihood principle and the adaptive learning algorithms are derived using the gradient ascent technique. In the second section of the paper the parameters of the linear regressive model are determined by minimizing the arithmetic mean of square errors and an adaptive learning scheme of gradient descent type is also considered. We consider a probabilistic approach in the third section for modeling the effects of both the latent variables and noise. The cumulative effects of latent variables and noise are modeled in terms of multivariate Gaussian repartitions.

Key words and phrases : linear regressive models, learning from data, supervised learning, maximum likelihood principle, adaptive learning.

Mathematics Subject Classification (2000) **:** 62J05, 62J12.

1. Introduction

Machine Learning deals with programming computers to optimize a performance criterion on the basis of finite sets of example data or past experience. Dealing with the design of algorithms and techniques that allow machines to learn in the sense that they improved the performance through experience, machine learning can be viewed as a branch of artificial intelligence.

The tremendous growth in practical applications of machine learning over the past decade has been accompanied by a wide variety of important developments in the underlying algorithms and techniques that make use of concepts and results coming from several areas as mathematical statistics, computer science and engineering.

Since the main aim of machine learning is to obtain computer programs that are able to extract information from samples of data and as well as knowledge from the past experience and include them the process of solving

problems of high complexity, the research methodology in the field of machine learning is essentially based on a large class of concepts and results coming from mathematical statistics, neural and evolutionary computation, brain models, adaptive control theory and so on.

Learning from data means a process of information extraction from finite size samples in order to estimate an unknown dependency. A series of problems can be modeled in terms of a system that computes according to an unknown rule a response (output) for each input. In supervised learning the basis of an inference process concerning the unknown dependency between the inputs and the outputs of the system is represented by a finite size set of labeled examples (x_i, y_i), $1 \leq i \leq N$, where for each (x_i, y_i), y_i is the output computed by the system for the input x_i. Learning from data comprises a class of model-free methods and algorithms that estimate the unknown dependency without assuming the existence of a model for data, neither in the input space nor in the space of responses. Once such a dependency has been accurately estimated, it can be used for prediction of future system outputs for known input values. The paper provides a series of results concerning the learning from data a linear regressive model in a multivariate framework. The parameter estimates of the regressive model are determined using the maximum likelihood principle and the adaptive learning algorithms are derived using the gradient ascent technique. In the second section of the paper the parameters of the linear regressive model are determined by minimizing the arithmetic mean of square errors and an adaptive learning scheme of gradient descent type is also considered. We consider a probabilistic approach in the third section for modeling the effects of both the latent variables and noise. The cumulative effects of latent variables and noise are modeled in terms of multivariate Gaussian repartitions. The predicted output is expressed as the sum of a linear combination of the entries of the input and the random vector that represents the effects of the unobservable factors and noise. The parameters of the regressive model are estimated by maximizing the likelihood function for given finite length sequence of observations, and an adaptive learning algorithm of gradient ascent type is proposed in the final part of the section. The final section of the paper contains a series of concluding remarks and suggestions for further work.

We consider the learning environment described in [10] and [12], where $\mathbf{S}$ is a system that for any n-dimensional input x computes an m-dimensional output y according to an unknown law. In the simplest approach we can assume that the output y is uniquely determined by the input x. However, the output can be influenced by a series of unobservable factors, and the dependency between the inputs and outputs of $\mathbf{S}$ could be of non-deterministic type. Consequently, in a more sophisticated approach we are forced to take into account a non-deterministic dependency, modelled for instance in probabilistic terms, as a reasonable hypothesis concerning the unknown law. The

Generator, denoted by **G**, is the source that generates the inputs. Mainly, there are two ways to model **G**, namely when the mechanism of generating inputs is known by the observer and when the law according to which the inputs are generated is also unknown, respectively. The third component of the learning environment denoted by **L**, is responsible with possible models of the unknown dependency corresponding to **S**. The learning component **L** implements a class of hypothesis (models) Ω, such that to each particular hypothesis $\omega \in \Omega$ corresponds a function $\varphi_\omega : \mathbb{R}^n \to \mathbb{R}^m$ defined on the space of inputs $\mathbb{R}^n$ and taking values in the space of outputs $\mathbb{R}^m$. For each particular input x_0, $\widehat{y}_0 = \varphi_\omega(x_0)$ is the estimate of the **S**'s output corresponding to x_0 in case of the model ω. Being given a criterion function $\mathcal{C}$ that expresses numerically the fitness of each model with respect to the available evidence E, about **S**, the best model $\omega_0(E)$ is a solution of the optimization problem

$$\arg\left(\text{optimize}_{\omega\in\Omega}\,\mathcal{C}(\omega, E)\right) . \tag{1.1}$$

In the case of supervised learning the available evidence E is represented by a finite set of pairs $\{(x_i, y_i)\ ,\ 1 \le i \le N\} \subset \mathbb{R}^n \times \mathbb{R}^m$, where each y_i is the actual output of **S** for the input x_i. If we assume that the unknown dependency is of deterministic type, that is the inputs and the outputs of **S** are functionally related a reasonable choice of the criterion function $\mathcal{C}$ is the arithmetic mean of the square errors, that is for each $\omega \in \Omega$,

$$\mathcal{C}(\omega, E) = \frac{1}{N}\sum_{i=1}^{N} \|y_i - \varphi_\omega(x_i)\|^2 . \tag{1.2}$$

The optimization problem (1.1) becomes

$$\arg\left(\min_{\omega\in\Omega} \mathcal{C}(\omega, E)\right) , \tag{1.3}$$

and its solutions are called the Minimum Square Errors (MSE) models computed on the basis of $\{(x_i, y_i)\ ,\ 1 \le i \le N\}$.

In case we adopt a more complex approach by including the effects of possible existing latent variables, each hypothesis $\omega \in \Omega$ corresponds to a probabilistic model for the latent vector. For simplicity sake, we consider that the latent vector is a continuous random vector, that is to each $\omega \in \Omega$ corresponds a conditional density function $f(\cdot|\cdot, ; \omega)$. Put in other words, for each $\omega \in \Omega$, $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, $f(y|x\,;\omega)$ expresses 'the chance' of getting the output y for the input x in case of the model ω. If the available evidence about **S** is $\{(x_i, y_i)\ ,\ 1 \le i \le N\}$ then a reasonable choice of $\mathcal{C}(\omega, E)$ is the likelihood function. If we assume that the inputs $x_1, \ldots, x_N$ are independently generated by **G** then

$$\mathcal{C}(\omega, E) = \prod_{i=1}^{N} f\left(y_i|x_i\,;\omega\right) , \tag{1.4}$$

and the optimization problem (1.1) becomes

$$\arg\left(\max_{\omega\in\Omega}\mathcal{C}(\omega,E)\right). \tag{1.5}$$

The solutions of (1.5) are the Maximum Likelihood ML models computed on the basis of $\{(x_i,y_i)\ ,\ 1\leq i\leq N\}$.

2. Modeling the learning system in terms of linear hypothesis

Let $\mathcal{P}$ be the procedure used by the learning component, **L**, to extract information from the available data $(x_1,y_1),\ldots,(x_N,y_N)$ in order to compute an approximation of the actual but unknown dependency between the inputs and the outputs of **S**. If we denote by $\widehat{\omega}$ the model computed by $\mathcal{P}$ then $\widehat{y}=\varphi_{\widehat{\omega}}(x)$ is the predicted output if the input x is applied to **S** where, $\varphi_{\widehat{\omega}}:\mathbb{R}^n\longrightarrow\mathbb{R}^m$. The simplest class of hypothesis is the linear class where each individual hypothesis corresponds to a linear transform. In this case for each model $\omega\in\Omega$, the predicted output is $\varphi_\omega(x)=\beta^T\begin{pmatrix}1\\x\end{pmatrix}$, $\beta\in\mathcal{M}_{(n+1)\times m}(\mathbb{R})$. From the point of view of the MSE (Minimum Square Errors) criterion, being given the data $(x_1,y_1),\ldots,(x_N,y_N)$, the optimal model is $\widehat{\omega}=\omega_{\widehat{\beta}_{MSE}}$ where

$$\widehat{\beta}_{MSE}=arg\left(\min_{\beta\in\mathcal{M}_{(n+1)\times m}(\mathbb{R})}F_N(\beta)\right), \tag{2.1}$$

$F_N(\beta)=\frac{1}{N}\sum_{i=1}^{N}\left\|y_i-\beta^Tz_i\right\|^2$, $z_i=\begin{pmatrix}1\\x_i\end{pmatrix}$, $1\leq i\leq N$.

Using straightforward computations we get

$$F_N(\beta)=\operatorname{tr}\left(\widehat{P}_N\right)-2\operatorname{tr}\left(\beta^T\widehat{Q}_N\right)+\operatorname{tr}\left(\beta^T\widehat{S}_N\beta\right),$$

where

$$\widehat{P}_N=\frac{1}{N}\sum_{i=1}^{N}y_iy_i^T\in\mathcal{M}_m(\mathbb{R}),\quad \widehat{Q}_N=\frac{1}{N}\sum_{i=1}^{N}z_iy_i^T\in\mathcal{M}_{(n+1)\times m}(\mathbb{R}),$$
$$\widehat{S}_N=\frac{1}{N}\sum_{i=1}^{N}z_iz_i^T\in\mathcal{M}_{n+1}(\mathbb{R})$$

If we denote by

$$Z=(z_1,\ldots,z_N)\in\mathcal{M}_{(n+1)\times N}(\mathbb{R}),\quad Y=(y_1,\ldots,y_N)\in\mathcal{M}_{m\times N}(\mathbb{R}).$$

the matrices of augmented inputs and corresponding outputs respectively, we get the compact forms

$$NF_N(\beta)=\mathrm{tr}\left(\left(Y-\beta^T Z\right)\left(Y-\beta^T Z\right)^T\right),$$

$$N\widehat{P}_N=YY^T,\ N\widehat{Q}_N=ZY^T,\ N\widehat{S}_N=ZZ^T.$$

The generalized gradient is

$$\nabla_\beta F_N(\beta) = -2\,\widehat{Q}_N + 2\,\widehat{S}_N\beta\,,$$

that is the space of critical points of the objective function F_N is

$$\mathcal{S} = \left\{\beta\in\mathcal{M}_{(n+1)\times m}(\mathbb{R}) \;\middle|\; \widehat{S}_N\beta=\widehat{Q}_N\right\}.$$

Let us denote by Z^+ the Penrose pseudo-inverse matrix. Then, for any $\beta\in\mathcal{M}_{(n+1)\times m}(\mathbb{R})$,

$$\begin{aligned}NF_N(\beta)=&\,\mathrm{tr}\left(Y\left(\mathbf{I}_N-Z^+Z\right)Y^T\right)+\mathrm{tr}\left(\left(YZ^+-\beta^T\right)ZZ^T\left(YZ^+-\beta^T\right)^T\right)\geq\\ &\,\mathrm{tr}\left(Y\left(\mathbf{I}_N-Z^+Z\right)Y^T\right)=NF_N\left(\left(YZ^+\right)^T\right),\end{aligned}$$

that is $\widehat{\beta}_{MSE}=\left(YZ^+\right)^T$ is the best model for $\mathbf{S}$.

Obviously, $\widehat{S}_N^+=N\left(Z^T\right)^+Z^+$ and $\widehat{S}_N^+\widehat{Q}_N=\left(YZ^+\right)^T$, therefore the expression of $\widehat{\beta}_{MSE}$ becomes $\widehat{\beta}_{MSE}=\left(YZ^+\right)^T=\widehat{S}_N^+\widehat{Q}_N\in\mathcal{S}$. Consequently, for the input x, the best prediction about the output of $\mathbf{S}$ is $\widehat{y}=\widehat{\beta}_{MSE}x$, computed on the basis of the available data. So, we can formulate the following result:

Theorem 2.1. (see [10], [15]) *The MSE estimation for parameter β for given data (x_1,y_1), $(x_2,y_2),\ldots,(x_N,y_N)$, is*

$$\widehat{\beta}_{MSE}=\left(YZ^+\right)^T\,. \tag{2.2}$$

Adaptive learning based on data $(x_1,y_1),\ldots,(x_N,y_N)$ can be done using gradient descent methods and/or stochastic gradient methods.

A learning scheme obtained using the gradient descent method ("batch") updates the parameter β according to the rule

$$\beta^{new}\leftarrow\beta^{old}+\rho\left(Q-S\beta^{old}\right).$$

The stochastic gradient method is the sequential version of the gradient descent procedure, where each example determines new values for the parameter entries instead of cumulating of the contributions of all at the

updating step. Put in other words, the new updated entries resulted after testing the example z_i are

$$\beta_{jk}^{new} \leftarrow \beta_{jk} + \rho z_i^{(j)} \left(y_i^{(k)} - \sum_{p=1}^{n+1} z_i^{(p)} \beta_{pk} \right), 1 \leq k \leq m\,,\ 1 \leq j \leq n+1\,.$$

Note that the stochastic gradient learning scheme has a "locality feature", in the sense that the updating of each entry β_{jk} involves only the current example and the entries of the column of index k of the matrix β. This particularity allows the implementation on a simple feed-forward neural network having an unique computation layer, where each column of β corresponds to the synaptic memory of just one neuron belonging to the computation layer. Consequently, we can conclude that the advantages of using the stochastic gradient method scheme reside, on one hand from its computational simplicity, and on the other hand from the "locality feature" that allows implementation on a simple feed-forward neural network (see [5]). The stochastic gradient descent learning scheme is briefly,

Input: $(x_1, y_1), \ldots, (x_N, y_N)$
Initializations: β_0, $\rho > 0$, $\mathcal{C}$, $z_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix}$, $1 \leq i \leq N$;
$\beta^{old} \leftarrow \beta_0$
repeat
  $\beta = \beta^{old}$
  for $i \leftarrow 1, N$
    for $k \leftarrow 1, m$
      for $j \leftarrow 1, n+1$

$$\beta_{jk}^{new} \leftarrow \beta_{jk} + \rho z_i^{(j)} \left(y_i^{(k)} - \sum_{p=1}^{n+1} z_i^{(p)} \beta_{pk} \right)$$

      end for
    end for
    $\beta \leftarrow \beta^{new}$
  end for
  evaluate $\mathcal{C}$
  $\beta^{old} \leftarrow \beta^{new}$
until $\mathcal{C}$
Output: β^{new}

where $\mathcal{C}$ is a stopping condition and a learning rate ρ is a conventionally selected positive number.

The stochastic gradient learning algorithm can be implemented on a simple two-layer feed-forward neural architecture where the input layer F_X and the output layer F_Y consist of $n+1$ and m neurons respectively, the synaptic memories of the neurons in the output layer F_Y being the columns of the currently computed matrix β. The scheme is presented in Figure 1.

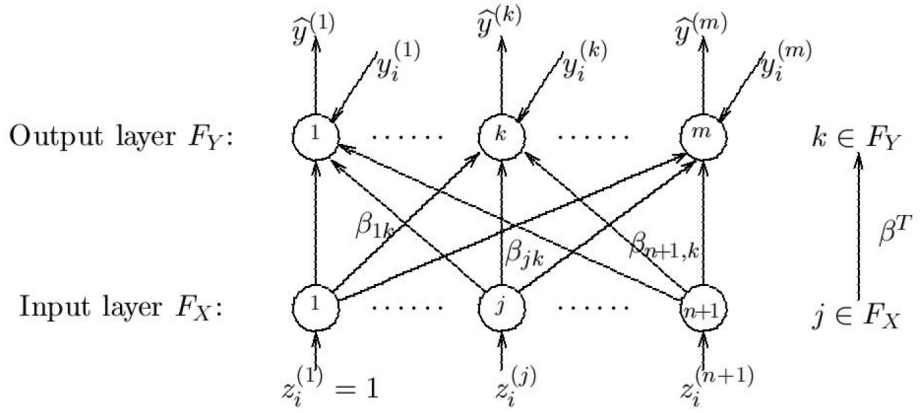

Figure 1. *Architecture feed-forward type neural model of* **S**.

During the learning process, if (x_i, y_i) is the current test example, $x_i = \left(x_i^{(1)},\ x_i^{(2)}, \ldots, x_i^{(n)}\right)$, $y_i = \left(y_i^{(1)}, \ldots, y_i^{(m)}\right)$, then the entries of $z_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix}$ are applied as inputs to the neurons of F_X and the entries of y_i are combined at the level of each neuron k of F_Y to update its synaptic memory according to the rule

$$\beta_{jk} \leftarrow \beta_{jk} + \rho z_i^{(j)} \left(y_i^{(k)} - \sum_{p=1}^{n+1} z_i^{(p)} \beta_{pk} \right), \quad j = 1, \ldots, n+1 \,.$$

3. Probabilistic model

An alternative approach to the modeling of the unknown input-output dependency can be considered by postulating certain parametric expression for the conditional repartition of the outputs on the inputs. For each hypothesis ω, input x and output y, let $f\,(y|x,\omega)$ be the probability to obtain the output y for the input x, being given the model ω. Several intuitively justified criteria can be considered in order to identify the "fittest" model on the basis of the available data $\mathcal{S}_N = \{(x_i, y_i)\ ,\ 1 \le i \le N\}$. In the following we consider the likelihood function $L\,(\omega, x_1, \ldots, x_N, y_1, \ldots, y_N)$ to express the quality of each model ω to explain the available data.

Assuming that the inputs are independent and the output depends only on the applied input, the likelihood of $\mathcal{S}_N$ is

$$L\,(\omega, x_1, \ldots, x_N, y_1, \ldots, y_N) = \prod_{i=1}^{N} f\,(y_i|x_i,\omega) \overset{not}{=} L\,(\omega, \mathcal{S}_N)\,,$$

an optimal model $\widehat{\omega}_{MLE}$ according to the principle of maximum likelihood

being a solution of the optimization problem

$$\arg\left(\max_{\omega\in\Omega}\left(L\left(\omega,\mathcal{S}_N\right)\right)\right).$$

Usually, the output of $\mathbf{S}$ is conditioned not only by the observable input, and some unobservable latent variables as well as noise can also influence its value. In other words, for each input x_i the output y_i of $\mathbf{S}$ depends on x_i and on unobservable variable ε. If we consider a parametric expression to model the effect of x_i on y_i then $y_i = g\left(\beta, x_i\right) + \varepsilon$.

A natural extension of the model proposed in section 2 can result by additively combining a linear dependency on the input entries with a Gaussian multi-variational model for the effect of noise and/or latent variables. In this framework the estimate of the unknown conditional repartition of the output of $\mathbf{S}$ on the input is $\widehat{p}(y|x)=\beta^T\begin{pmatrix}1\\x\end{pmatrix}+h(\varepsilon)$, where h is the density function of the m-dimensional Gaussian repartition $h(\varepsilon)\sim\mathcal{N}(\mu,\Sigma)$. Put in other words, the estimate of the output of $\mathbf{L}$ for the input x_i is $\widetilde{y}_i=\beta^T z_i+\varepsilon$, where $z_i=\begin{pmatrix}1\\x_i\end{pmatrix}$, $\varepsilon\sim\mathcal{N}\left(\mu,\Sigma\right)$, $\mu\in\mathbb{R}^m$ and $\Sigma\in\mathcal{M}_m(\mathbb{R})$ is a symmetric positive definite matrix.

In this case, each particular hypothesis $\omega\in\Omega$ corresponds to a tuple (β,μ,Σ), the conditional repartition of the output of $\mathbf{L}$ for each input x_i being modeled by the density function of $\widetilde{y}_i$. Obviously, if the normal model for the effect of noise and latent variables is assumed, we get $\widetilde{y}_i\sim\mathcal{N}\left(\beta^T z_i+\mu,\Sigma\right)$, that is the expression of the density function in the hypothesis $\omega=(\beta,\mu,\Sigma)$ is

$$f\left(y_i|x_i,\beta,\mu,\Sigma\right)=\frac{1}{\sqrt{(2\pi)^m\left|\Sigma\right|}}\exp\left\{-\frac{1}{2}\left(y_i-\beta^T z_i-\mu\right)^T\Sigma^{-1}\left(y_i-\beta^T z_i-\mu\right)\right\},$$

and the log-likelihood function is

$$l\left(\beta,\mu,\Sigma,\mathcal{S}_N\right)=-\frac{Nm}{2}\ln(2\pi)-\frac{N}{2}\ln\left|\Sigma\right|-\frac{1}{2}\sum_{i=1}^{N}\left(y_i-\beta^T z_i-\mu\right)^T\Sigma^{-1}\left(y_i-\beta^T z_i-\mu\right)$$

In the following will try to fit the best model using the paradigm of the maximum likelihood being given the set of examples $\{z_1,\ldots,z_N\}$. For simplicity sake, we consider two special cases. In the first special case we consider that the latent variable ε is white noise $\varepsilon\sim\mathcal{N}\left(\mathbf{0}_m,\mathbf{I}_m\right)$. So, the log-likehood function is

$$l\left(\beta,\mathbf{0}_m,\mathbf{I}_m,\mathcal{S}_N\right)=-\frac{Nm}{2}\ln(2\pi)-\frac{1}{2}\sum_{i=1}^{N}\left(y_i-\beta^T z_i\right)^T\left(y_i-\beta^T z_i\right)\overset{not}{=}l\left(\beta,\mathcal{S}_N\right),$$

therefore the best model is

$$\widehat{\beta}_{MLE} = \arg\left(\max_{\beta\in\mathcal{M}_{(n+1)\times m}(\mathbb{R})} l\left(\beta, \mathcal{S}_N\right)\right) = \arg\left(\min_{\beta\in\mathcal{M}_{(n+1)\times m}(\mathbb{R})} \sum_{i=1}^{N} \left(y_i - \beta^T z_i\right)^T \left(y_i - \beta^T z_i\right)\right).$$

Theorem 3.1. *If $\varepsilon \sim \mathcal{N}\left(\mathbf{0}_m, \mathbf{I}_m\right)$ then the MLE estimation for parameter β, for given data $(x_1, y_1)\ldots, (x_N, y_N)$, is*

$$\widehat{\beta}_{MLE} = \left(YZ^+\right)^T . \tag{3.1}$$

Proof. The gradient of the log-likelihood function with respect to β is

$$\nabla_\beta l\left(\beta, \mathcal{S}_N\right) = ZY^T - ZZ^T\beta\,,$$

that is the space of critical points is the set of solutions of the equation

$$ZY^T - ZZ^T\beta = \mathbf{O}_{(n+1)\times m}.$$

Since the unique critical point is $\beta_0 = \left(YZ^+\right)^T$, the value of the log-likehood function is

$$l\left(\beta_0, \mathcal{S}_N\right) = -\frac{mN}{2}\ln(2\pi) - \frac{1}{2}\mathrm{tr}\left(Y\left(\mathbf{I}_N - Z^+Z\right)\left(\mathbf{I}_N - Z^+Z\right)^T Y^T\right).$$

In order to prove that β_0 is the maxima point of the log-likelihood function, let β be an arbitrary point. Hence

$$l\left(\beta, \mathcal{S}_N\right) = -\frac{mN}{2}\ln(2\pi) - \frac{1}{2}\mathrm{tr}\left(\left(Y - \beta^T Z\right)^T\left(Y - \beta^T Z\right)^T\right).$$

Using the relations

$$\begin{aligned}&\left(\mathbf{I}_N - Z^+Z\right)\left(\mathbf{I}_N - Z^+Z\right)^T = \left(\mathbf{I}_N - Z^+Z\right),\\ &Y - \beta^T Z = Y\left(\mathbf{I}_N - Z^+Z\right) + \left(\beta_0 - \beta\right)^T Z\end{aligned}$$

we obtain

$$l\left(\beta, \mathcal{S}_N\right) = l\left(\beta_0, \mathcal{S}_N\right) - \frac{1}{2}\mathrm{tr}\left(\left(\beta_0 - \beta\right)^T ZZ^T\left(\beta_0 - \beta\right)\right) - \mathrm{tr}\left(Y\left(\mathbf{I}_N - Z^+Z\right)Z^T\left(\beta_0 - \beta\right)^T\right).$$

Obviously, $\left(\mathbf{I}_N - Z^+Z\right)AZ^T = \mathbf{O}_{N,n+1}$, and therefore

$$l\left(\beta, \mathcal{S}_N\right) = l\left(\beta_0, \mathcal{S}_N\right) - \frac{1}{2}\mathrm{tr}\left(\left(\beta_0 - \beta\right)^T ZZ^T\left(\beta_0 - \beta\right)\right) \leq l\left(\beta_0, \mathcal{S}_N\right).$$

Consequently, $\widehat{\beta}_{MLE} = \beta_0$. □

In the second special case we suppose that the latent variable is $\varepsilon \sim \mathcal{N}(\mu, \Sigma_0)$, where Σ_0 is a fixed variance matrix, that is the model is given by $\omega = (\beta, \mu)$, and the log-likelihood function is

$$l(\beta, \mu, \Sigma_0, \mathcal{S}_N) \overset{not}{=} l(\beta, \mu, \mathcal{S}_N) =$$
$$-\frac{mN}{2}\ln(2\pi) - \frac{N}{2}\ln|\Sigma_0| - \frac{1}{2}\sum_{i=1}^{N}\left((y_i-\mu)^T - z_i^T\beta\right)\Sigma_0^{-1}\left((y_i-\mu) - \beta^T z_i\right).$$

Therefore the best MLE model is

$$\left(\widehat{\beta}_{MLE}, \widehat{\mu}_{MLE}\right) = \arg\left(\max_{\beta\in\mathcal{M}_{(n+1)\times m}(\mathbb{R}),\mu\in\mathbb{R}^m} l(\beta, \mu, \mathcal{S}_N)\right) =$$
$$\arg\left(\min_{\beta\in\mathcal{M}_{(n+1)\times m}(\mathbb{R}),\mu\in\mathbb{R}^m} \sum_{i=1}^{N}\left((y_i-\mu)^T - z_i^T\beta\right)\Sigma_0^{-1}\left((y_i-\mu) - \beta^T z_i\right)\right).$$

Theorem 3.2. *If $\varepsilon \sim \mathcal{N}(\mu, \Sigma_0)$ then the maximum likelihood estimates of the parameters β, and μ are*

$$\widehat{\beta}_{MLE} = \left(Y(ZA)^+\right)^T, \quad \widehat{\mu}_{MLE} = \frac{1}{N}\left(Yu - Y(ZA)^+ Zu\right), \tag{3.2}$$

where $u = (1, \ldots, 1)^T \in \mathbb{R}^N$, $A = \mathbf{I}_N - \frac{1}{N}uu^T$.

Proof. The gradients of the log-likelihood function with respect to β, and μ are

$$\nabla_\beta l(\beta, \mu, \mathcal{S}_N) = \left(ZY^T - (Zu)\mu^T - ZZ^T\beta\right)\Sigma_0^{-1}, \text{ and}$$
$$\nabla_\mu l(\beta, \mu, \mathcal{S}_N) = \Sigma_0^{-1}\left(Yu - N\mu - \beta^T(Zu)\right),$$

that is the space of critical points is the set of the solution of the system

$$\begin{cases} ZY^T - (Zu)\mu^T - ZZ^T\beta = \mathbf{O}_{(n+1)\times m} \\ Yu - N\mu - \beta^T(Zu) = \mathbf{0}_m. \end{cases}$$

Since from the second vectorial equation we get $\mu = \frac{1}{N}\left(Yu - \beta^T Zu\right)$, by replacing it in the first vectorial equation we obtain $\beta_0 = \left(ZAZ^T\right)^+\left(ZAY^T\right)$. Using the obvious properties $A^2 = A = A^T$ and $A^+ = A$, we obtain

$$\beta_0 = \left(Y(ZA)^+\right)^T.$$

Therefore, by replacing it in the expression of μ, we get

$$\mu_0 = \frac{1}{N}Y\left(\mathbf{I}_N - Y(ZA)^+Z\right)u.$$

Using straightforward computations we obtain

$$l\left(\beta_0,\mu_0,\mathcal{S}_N\right)=-\frac{mN}{2}\ln(2\pi)-\frac{N}{2}\ln\left|\Sigma_0\right|-\\ \frac{1}{2}\mathrm{tr}\left(\Sigma_0^{-1}Y\left(A-(ZA)^{+}ZA\right)\left(A-(ZA)^{+}ZA\right)^{T}Y^{T}\right).$$

In order to prove that (β_0,μ_0) is the maxima point of the log-likelihood function, for arbitrary (β,μ) we get

$$l\left(\beta,\mu,\mathcal{S}_N\right)=-\frac{mN}{2}\ln(2\pi)-\frac{N}{2}\ln\left|\Sigma_0\right|-\\ \frac{1}{2}\mathrm{tr}\left(\Sigma_0^{-1}\left(Y-\mu u^T-\beta^TZ\right)\left(Y-\mu u^T-\beta^TZ\right)^{T}\right).$$

In order to compare the values of the log-likelihood function for (β,μ) and (β_0,μ_0), since

$$\left(A-(ZA)^{+}ZA\right)\left(A-(ZA)^{+}ZA\right)^{T}=A-A\left(ZA\right)^{+}(ZA)\,,\\ Y-\mu u^T-\beta^TZ=Y\left(A-(ZA)^{+}(ZA)\right)+(\mu_0-\mu)\,u^T+(\beta_0-\beta)^TZ,$$

we obtain

$$l\left(\beta,\mu,\mathcal{S}_N\right)=l\left(\beta_0,\mu_0,\mathcal{S}_N\right)-\\ \frac{1}{2}\,\mathrm{tr}\left(\Sigma_0^{-1}\left(\mu_0-\mu\right)^Tu^Tu\left(\mu_0-\mu\right)\right)-\frac{1}{2}\,\mathrm{tr}\left(\Sigma_0^{-1}\left(\beta_0-\beta\right)^TZZ^T\left(\beta_0-\beta\right)\right)-\\ \mathrm{tr}\left(\Sigma_0^{-1}Y\left(\mathbf{I}_N-(ZA)^{+}Z\right)Au\left(\mu_0-\mu\right)^T\right)-\\ \mathrm{tr}\left(\Sigma_0^{-1}Y\left(\mathbf{I}_N-(ZA)^{+}Z\right)AZ^T\left(\beta_0-\beta\right)^T\right).$$

Obviously, $Au=\mathbf{0}_N$ and $\left(\mathbf{I}_N-(ZA)^{+}Z\right)AZ^T=\mathbf{O}_{N,n+1}$. Therefore

$$l\left(\beta,\mu,\mathcal{S}_N\right)=l\left(\beta_0,\mu_0,\mathcal{S}_N\right)-\frac{1}{2}\,\mathrm{tr}\left(\Sigma_0^{-1}\left(\mu_0-\mu\right)^Tu^Tu\left(\mu_0-\mu\right)\right)-\\ \frac{1}{2}\,\mathrm{tr}\left(\Sigma_0^{-1}\left(\beta_0-\beta\right)^TZZ^T\left(\beta_0-\beta\right)\right)\leq l\left(\beta_0,\mu_0,\mathcal{S}_N\right)\,.$$

□

The adaptive learning of the parameters μ and β may be installed using for instance the gradient ascent method ("batch" version), yielding to the following learning algorithm.

Input: $(x_1,y_1),\ldots,(x_N,y_N)$, Σ_0
Initializations: β_0, μ_0, $\mathcal{C}$, $\rho>0$
Compute $z_i=\begin{pmatrix}1\\ x_i\end{pmatrix}$, $1\leq i\leq N$; $Q=\sum_{i=1}^{N}z_iy_i^T$; $S=\sum_{i=1}^{N}z_iz_i^T$
$\Sigma_{01}=\Sigma_0^{-1}$, $\beta^{old}\leftarrow\beta_0$, $\mu^{old}\leftarrow\mu_0$
repeat

$$\beta^{new}\leftarrow\beta^{old}+\rho\left(Q-\left(\sum_{i=1}^{N}z_i\right)\left(\mu^{old}\right)^T-S\beta^{old}\right)\Sigma_{01}$$

$$\mu^{new} \leftarrow \mu^{old} + \rho\Sigma_{01}\left(\left(\sum_{i=1}^{N} y_i\right) - \sum_{i=1}^{N}\left(\beta^{old}\right)^T z_i - N\mu^{old}\right)$$

`evaluate` $\mathcal{C}$

$\beta^{old} \leftarrow \beta^{new}$; $\mu^{old} \leftarrow \mu^{new}$

`until` $\mathcal{C}$

`Output:` β^{new}, μ^{new}.

The stopping condition $\mathcal{C}(\delta)$ may be

$$\mathcal{C}(\delta) = true \iff \left\|\beta^{new} - \beta^{old}\right\| + |\mu^{new} - \mu^{old}| < \delta$$

or $$\mathcal{C}(\delta) = true \iff \left\|\beta^{new} - \beta^{old}\right\| < \delta \text{ and } |\mu^{new} - \mu^{old}| < \delta.$$

The components of the parameters β and μ are updated by the above procedure as

$$\beta_{jk}^{new} \leftarrow \beta_{jk}^{old} + \rho\sum_{i=1}^{N} z_i^{(j)}\left(\sum_{r=1}^{m}\left(y_i^{(r)} - \left(\mu^{old}\right)^{(r)} - \sum_{p=1}^{n+1} z_i^{(p)}\beta_{pr}^{old}\right)\left(\Sigma_0^{-1}\right)_{rk}\right),$$

$$1 \le j \le n+1,\ 1 \le k \le m,$$

$$(\mu^{new})^{(k)} \leftarrow \left(\mu^{old}\right)^{(k)} + \rho\sum_{i=1}^{N}\left(\sum_{r=1}^{m}\left(\left(\Sigma_0^{-1}\right)_{kr}\left(y_i^{(r)} - \left(\mu^{old}\right)^{(r)} - \sum_{p=1}^{n+1} z_i^{(p)}\beta_{pr}^{old}\right)\right)\right),$$

$$1 \le k \le m.$$

According to these relations the following stochastic training scheme can be derived

`Input:` $(x_1, y_1), \ldots, (x_N, y_N)$; Σ_0

`Initializations:` β_0, μ_0, $\mathcal{C}(\delta)$, $\delta > 0$, $\rho > 0$

$z_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix}$, $1 \le i \le N$;

$\Sigma_{01} \leftarrow \Sigma_0^{-1}$

$\beta^{old} \leftarrow \beta_0$, $\mu^{old} \leftarrow \mu_0$

`repeat`

  $\beta \leftarrow \beta^{old}$, $\mu \leftarrow \mu^{old}$

  `for` $i \leftarrow \overline{1, N}$

    `for` $k \leftarrow \overline{1, m}$

      `for` $j \leftarrow \overline{1, n+1}$

        $\beta_{jk}^{new} \leftarrow \beta_{jk} + \rho z_i^{(j)}\sum_{r=1}^{m}\left(y_i^{(r)} - \mu^{(r)} - \sum_{p=1}^{n+1} z_i^{(p)}\beta_{pr}\right)(\Sigma_{01})_{rk}$

      `end for`

$$(\mu^{new})^{(k)} \leftarrow \mu^{(k)} + \rho \sum_{r=1}^{m} (\Sigma_{01})_{kr} \left(y_i^{(r)} - \mu^{(r)} - \sum_{p=1}^{n+1} z_i^{(p)} \beta_{pr} \right)$$

```
    end for
    β ← β^new , μ ← μ^new
  end for
  compute C(δ)
  β^old ← β^new , μ^old ← μ^new
until C(δ)
Output: β^new , μ^new .
```

Unlike the "batch" procedure, the updating of the parameters β and μ in the stochastic procedure is performed as a consequence of each tested example, therefore the order in which examples are considered may influence both the duration of training, and the accuracy of the resulted estimates.

If $\Sigma_0 = \mathbf{I}_m$, then the updating of the entries of the parameters β and μ is performed according to,

$$\beta_{jk}^{new} \leftarrow \beta_{jk}^{old} + \rho \sum_{i=1}^{N} z_i^{(j)} \left(y_i^{(k)} - \left(\left(\mu^{old} \right)^{(k)} + \beta_{1k}^{old} \right) - \sum_{p=2}^{n+1} z_i^{(p)} \beta_{pk}^{old} \right), \quad 1 \le j \le n+1,\ 1 \le k \le m,$$

$$(\mu^{new})^{(k)} \leftarrow \left(\mu^{old} \right)^{(k)} + \rho \sum_{i=1}^{N} \left(y_i^{(k)} - \left(\left(\mu^{old} \right)^{(k)} + \beta_{1k}^{old} \right) - \sum_{p=2}^{n+1} z_i^{(p)} \beta_{pk}^{old} \right), \quad 1 \le k \le m\,.$$

In this case the training corresponding to the stochastic procedure has the "locality feature", that is it can be also implemented on a simple two-layer feed-forward neural architecture. The input layer F_X and the output layer F_Y consist of $n+1$ and m neurons respectively, the synaptic memories of the neurons in the output layer F_Y being $\mu+\beta_1, \beta_2, \ldots, \beta_m$, where $\beta_1, \beta_2, \ldots, \beta_m$ are the columns of the currently computed matrix β. The scheme is presented in Figure 2.

During the learning process, if (x_i, y_i) is the current test example, $x_i = \left(x_i^{(1)}, \ldots, x_i^{(n)} \right)$, $y_i = \left(y_i^{(1)}, \ldots, y_i^{(m)} \right)$, then the entries of $z_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix}$ are applied as inputs to the neurons of F_X and the entries of y_i are combined at the level of each neuron k of F_Y to update its synaptic memory according

to the rule

$$\beta_{jk} \leftarrow \beta_{jk}^{old} + \rho \sum_{i=1}^{N} z_i^{(j)} \left(y_i^{(k)} - \left(\mu^{(k)} + \beta_{1k} \right) - \sum_{p=2}^{n+1} z_i^{(p)} \beta_{pk} \right),$$

$$\mu^{(k)} \leftarrow \mu^{(k)} + \rho \sum_{i=1}^{N} \left(y_i^{(k)} - \left(\mu^{(k)} + \beta_{1k} \right) - \sum_{p=2}^{n+1} z_i^{(p)} \beta_{pk} \right).$$

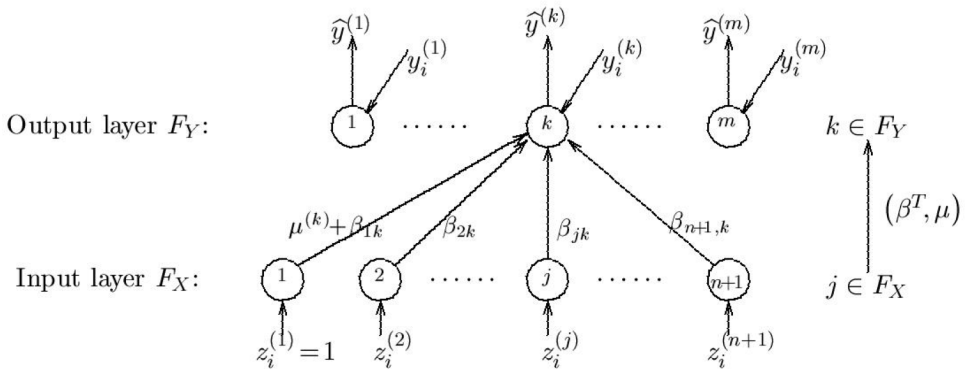

Figure 2: *Architecture feed-forward type neural model of* $\mathbf{S}$*, in case* $\mu \in \mathbb{R}^m$ *and* $\Sigma_0 = \mathbf{I}_m$.

In the general case of the probabilistic model the dimension of the space of hypotheses is $m(m+n+2)$, and each particular tuple $\omega = (\beta, \mu, \Sigma)$ defines a model of $\mathbf{S}$. The log-likelihood function is

$$l(\beta, \mu, \Sigma, \mathcal{S}_N) = -\frac{mN}{2} \ln(2\pi) - \frac{N}{2} \ln|\Sigma| - \frac{1}{2} \sum_{i=1}^{N} \left((y_i - \mu)^T - z_i^T \beta \right) \Sigma^{-1} \left((y_i - \mu) - \beta^T z_i \right),$$

and the best model from the point of view of maximum likelihood principle is a solution of the constrained optimization problem

$$\begin{cases} \max\limits_{\beta,\mu,\Sigma} \left(l(\beta, \mu, \Sigma, \mathcal{S}_N) \right) \\ \Sigma \in \mathcal{M}_m(\mathbb{R}) \text{ symmetric and positive defined.} \end{cases} \tag{3.3}$$

Theorem 3.3. *The objective function* $l(\beta, \mu, \Sigma, \mathcal{S}_N)$ *has an unique critical point* $(\beta_0, \mu_0, \Sigma_0)$ *where*

$$\beta_0 = \left(Y (ZA)^+ \right)^T, \; \mu_0 = \frac{1}{N} \left(Yu - Y (ZA)^+ Zu \right), \quad \Sigma_0 = \frac{1}{N} Y \left(A - (ZA)^+ (ZA) \right) Y^T, \tag{3.4}$$

and Σ_0 *is a symmetric and positive semi-defined matrix.*

Proof. The generalized gradients of $l(\beta, \mu, \Sigma, \mathcal{S}_N)$ with respect to β, μ, and Σ respectively, are

$$\begin{cases} \nabla_\beta l(\beta, \mu, \Sigma, \mathcal{S}_N) = \left(ZY^T - Zu\,\mu^T - ZZ^T\beta\right)\Sigma^{-1}, \\ \nabla_\mu l(\beta, \mu, \Sigma, \mathcal{S}_N) = \Sigma^{-1}\left(Yu - N\mu - \beta^T Zu\right), \\ \nabla_\Sigma l(\beta, \mu, \Sigma, \mathcal{S}_N) = -\Sigma^{-1}D\Sigma^{-1} + \dfrac{1}{2}\operatorname{diag}\left(\Sigma^{-1}D\Sigma^{-1}\right), \end{cases}$$

where

$$D = N\Sigma - YY^T + \Big(\mu\,(Yu)^T + (Yu)\,\mu^T\Big) + \Big(\left(ZY^T\right)^T\beta + \beta^T ZY^T\Big) - N\mu\mu^T - \Big(\mu\,(Zu)^T\beta + \beta^T(Zu)\,\mu^T\Big) - \beta^T ZZ^T\beta\,.$$

From the system

$$\begin{cases} \nabla_\mu l(\beta, \mu, \Sigma, \mathcal{S}_N) = \mathbf{0}_m \\ \nabla_\beta l(\beta, \mu, \Sigma, \mathcal{S}_N) = \mathbf{O}_{n+1,m} \end{cases},$$

since $|\Sigma| \neq 0$, we get

$$\begin{cases} Yu - N\mu - \beta^T Zu = \mathbf{0}_m \\ ZY^T - Zu\,\mu^T - ZZ^T\beta = \mathbf{O}_{n+1,m}\,, \end{cases}$$

that is

$$\beta = \left(Y\,(ZA)^+\right)^T = \beta_0,\ \mu = \frac{1}{N}\left(Yu - Y\,(ZA)^+\,Zu\right) = \mu_0.$$

Replacing μ_0, β_0 in the system

$$\nabla_\Sigma l(\beta, \mu, \Sigma, \mathcal{S}_N) = \mathbf{O}_m$$

we obtain $-\Sigma^{-1}D\Sigma^{-1} + \dfrac{1}{2}\operatorname{diag}\left(\Sigma^{-1}D\Sigma^{-1}\right) = \mathbf{O}_m$,
where $\operatorname{diag}\left(\Sigma^{-1}D\Sigma^{-1}\right) \in \mathcal{M}_m(\mathbb{R})$ is the diagonal matrix that retains only the entries placed on the main diagonal of $\Sigma^{-1}D\Sigma^{-1}$. Since Σ is a positive definite matrix, we get $D = \mathbf{O}_m$ and consequently,

$$\Sigma = \frac{1}{N}\left(YY^T + \beta_0^T ZZ^T\beta_0 - YZ^T\beta_0 - \beta_0^T ZY^T + \frac{1}{N}\left(\beta_0^T Zuu^T Y^T - \beta_0^T Zuu^T Z^T\beta_0 + Yuu^T Z^T\beta_0 - Yuu^T Y^T\right)\right) = \frac{1}{N}\left(YAY^T + \beta_0^T ZAZ^T\beta_0 - YAZ^T\beta_0 - \beta_0^T ZAY^T\right).$$

Using the well-known properties of the Penrose pseudo-inverse, the expression of Σ becomes

$$\Sigma = \frac{1}{N}\Big(YAY^T + Y\,(ZA)^+\,(ZA)\,AZ^T\left(Y\,(ZA)^+\right)^T - YAZ^T\left(Y\,(ZA)^+\right)^T - Y\,(ZA)^+\,ZAY^T\Big) = \frac{1}{N}Y\Big(A + \left((ZA)^+(ZA)(ZA)^+(ZA)\right)^T - (ZA)^+(ZA) - (ZA)^+(ZA)\Big)Y^T = \frac{1}{N}Y\left(A - (ZA)^+\,(ZA)\right)Y^T \overset{not}{=} \Sigma_0\,.$$

A further simplification can be obtained by noting that $B = A - (ZA)^+ ZA$ is a symmetric matrix and $B^2 = B$, that is the expression of Σ_0 can be written as

$$\Sigma_0 = \frac{1}{N} Y B B^T Y^T .$$

Obviously, Σ_0 is a symmetric and positive semi-defined matrix. □

Theorem 3.4. *Let β_0 and μ_0 be given by Theorem 3.3. Then for any (β, μ, Σ) in the parameter space*

$$l(\beta, \mu, \Sigma, \mathcal{S}_N) \leq l(\beta_0, \mu_0, \Sigma, \mathcal{S}_N) . \tag{3.5}$$

Proof. Using $v^T v = \operatorname{tr}\left(vv^T\right)$ the expression of the log-likelihood function can be written as

$$l(\beta, \mu, \Sigma, \mathcal{S}_N) = -\frac{mN}{2}\ln(2\pi) - \frac{N}{2}\ln|\Sigma| - \frac{1}{2}\operatorname{tr}\left(\Sigma^{-1}\left(Y - \mu u^T - \beta^T Z\right)\left(Y - \mu u^T - \beta^T Z\right)^T\right).$$

Therefore

$$l(\beta_0, \mu_0, \Sigma, \mathcal{S}_N) = -\frac{mN}{2}\ln(2\pi) - \frac{N}{2}\ln|\Sigma| - \frac{1}{2}\operatorname{tr}\left(\Sigma^{-1} Y \left(A - (ZA)^+ ZA\right)\left(A - (ZA)^+ ZA\right)^T Y^T\right) .$$

Using the relations $A = A^2 = A^T$ we get

$$\left(A - (ZA)^+ ZA\right)\left(A - (ZA)^+ ZA\right)^T = A - A(ZA)^+ (ZA) ,$$

and the term $Y - \mu u^T - \beta^T Z$ becomes

$$Y - \mu u^T - \beta^T Z = Y\left(A - (ZA)^+ (ZA)\right) + \frac{1}{N}\left(Yu - Y(ZA)^+ Zu - N\mu\right) u^T + \left(Y(ZA)^+ - \beta^T\right) Z = Y\left(A - (ZA)^+ (ZA)\right) + (\mu_0 - \mu) u^T + (\beta_0 - \beta)^T Z.$$

Consequently

$$\begin{aligned} l(\beta, \mu, \Sigma, \mathcal{S}_N) = {} & -\frac{mN}{2}\ln(2\pi) - \frac{N}{2}\ln|\Sigma| - \\ & \frac{1}{2}\operatorname{tr}\left(\Sigma^{-1} Y\left(A - (ZA)^+ ZA\right)\left(A - (ZA)^+ ZA\right)^T Y^T\right) - \\ & \operatorname{tr}\left(\Sigma^{-1} Y\left(A - (ZA)^+ ZA\right)\left((\mu_0 - \mu) u^T\right)^T\right) - \\ & \operatorname{tr}\left(\Sigma^{-1} Y\left(A - (ZA)^+ ZA\right)\left((\beta_0 - \beta)^T Z\right)^T\right) - \frac{1}{2}\operatorname{tr}\left(\Sigma^{-1}(\mu_0 - \mu)\right. \\ & \left. u^T u (\mu_0 - \mu)^T\right) - \frac{1}{2}\operatorname{tr}\left(\Sigma^{-1}(\beta_0 - \beta)^T Z Z^T (\beta_0 - \beta)\right) = \\ & l(\beta_0, \mu_0, \Sigma, \mathcal{S}_N) - \frac{1}{2}\operatorname{tr}\left(\Sigma^{-1}(\mu_0 - \mu) u^T u (\mu_0 - \mu)^T\right) \\ & - \frac{1}{2}\operatorname{tr}\left(\left(\Sigma^{-1}\beta_0 - \beta\right)^T Z Z^T (\beta_0 - \beta)\right) \operatorname{tr}\left(\Sigma^{-1} Y\left(\mathbf{I}_N - (ZA)^+ Z\right)\right. \\ & \left. Au(\mu_0 - \mu)^T\right) - \operatorname{tr}\left(\Sigma^{-1} Y\left(\mathbf{I}_N - (ZA)^+ Z\right) AZ^T(\beta_0 - \beta)\right) . \end{aligned}$$

Since $Au = \left(\mathbf{I}_N - \frac{1}{N}\, uu^T\right) u = u - \frac{1}{N} u \left(u^T u\right) = \mathbf{0}_N$, we get

$$Y\left(\mathbf{I}_N - (ZA)^+ Z\right) Au \left(\mu_0 - \mu\right)^T = \mathbf{O}_m.$$

Also, using the properties of the Penrose pseudo-inverse, we get

$$\left(\mathbf{I}_N - (ZA)^+ Z\right) AZ^T = \mathbf{O}_{N,n+1}\,,$$

that is

$$Y\left(\mathbf{I}_N - (ZA)^+ Z\right) AZ^T \left(\beta_0 - \beta\right) = \mathbf{O}_m.$$

Taking into account these arguments we finally obtain

$$l\left(\beta, \mu, \Sigma, \mathcal{S}_N\right) = l\left(\beta_0, \mu_0, \Sigma, \mathcal{S}_N\right) - \frac{1}{2} \operatorname{tr}\left(\Sigma^{-1}(\mu_0 - \mu) u^T u \left(\mu_0 - \mu\right)^T\right) - \frac{1}{2} \operatorname{tr}\left(\Sigma^{-1}(\beta_0 - \beta)^T ZZ^T (\beta_0 - \beta)\right).$$

Obviously, since Σ a positive definite matrix

$$\operatorname{tr}\left(\Sigma^{-1}(\mu_0 - \mu)\, u^T u \left(\mu_0 - \mu\right)^T\right) = N(\mu_0 - \mu)^T \Sigma^{-1} (\mu_0 - \mu) \geq 0$$

and

$$\operatorname{tr}\left(\Sigma^{-1} \left(\beta_0 - \beta\right)^T ZZ^T (\beta_0 - \beta)\right) = \operatorname{tr}\left(Z^T (\beta_0 - \beta)\, \Sigma^{-1} (\beta_0 - \beta)^T Z\right) = \operatorname{tr}\left(\left((\beta_0 - \beta)^T Z\right)^T \Sigma^{-1} (\beta_0 - \beta)^T Z\right) \geq 0\,,$$

that is $l\left(\beta, \mu, \Sigma, \mathcal{S}_N\right) \leq l\left(\beta_0, \mu_0, \Sigma, \mathcal{S}_N\right)$. □

Remark. Although, a long series of tests pointed out that the estimate Σ_0 given by Theorem 2.1 is a positive matrix, the mathematical proof is still an open problem. Also, it is not known whether the unique critical point $(\beta_0, \mu_0, \Sigma_0)$ corresponds to the best model in the sense of the maximum likelihood principle.

An adaptive learning procedure can be obtained using the gradient ascent method applied to the log-likelihood criterion function. The search developed by the adaptive procedure in a $m(m + n + 2)$-dimensional space aims to adjust the model parameters β, μ, Σ in order to maximize the log-likelihood function or, equivalently, to minimize $\Phi(\beta, \mu, \Sigma)$. The procedure should be implemented using a control parameter $\delta > 0$ and a stopping condition $\mathcal{C}(\delta)$ usually expressed in terms of the magnitude of the displacement in the parameter space due to the current iteration. In our tests $\mathcal{C}(\delta) = true$ if

$$\|\beta^{new} - \beta^{old}\| < \delta\,,\ \|\mu^{new} - \mu^{old}\| < \delta\,,\ \|\Sigma^{new} - \Sigma^{old}\| < \delta$$

where $\|\cdot\|$ is a conventionally norm, for instance Euclidian norm.

Also, the implementation of the procedure can be done using either a constant learning rate $\rho > 0$ or a decreasing sequence of positive learning rates (ρ_k) that refines the search while the search advances.

Since during the search process the estimates of Σ are not guaranteed to be invertible, the implementation procedure describe in procedure `MLE_gradient_ascent` uses approximations of the actual generalized gradients where the generalized inverse is used instead.

procedure MLE_gradient_ascent
Input: $\{(x_1, y_1), \ldots, (x_N, y_N)\}$
Initializations: $\delta > 0$, $\rho > 0$, $\widetilde{\beta}$, $\widetilde{\mu}$, $\widetilde{\Sigma}$,
$Z = \begin{pmatrix} 1 & \ldots, & 1 \\ x_1 & , \ldots, & x_N \end{pmatrix}$, $Y = (y_1, \ldots, y_N)$,
$u = (1, \ldots, 1)^T$
$\beta^{old} \leftarrow \widetilde{\beta}$, $\mu^{old} \leftarrow \widetilde{\mu}$, $\Sigma^{old} \leftarrow \widetilde{\Sigma}$
Compute $S = ZZ^T$, $Q = ZY^T$, $P = YY^T$
$Z_1 = Zu$, $Y_1 = Yu$
repeat
$\Sigma_1 \leftarrow \left(\Sigma^{old}\right)^+$
$\beta^{new} \leftarrow \beta^{old} + \rho\left(Q - Z_1\left(\mu^{old}\right)^T - S\beta^{old}\right)\Sigma_1$
$\mu^{new} \leftarrow \mu^{old} + \rho\Sigma_1\left(Y_1 - \left(\beta^{old}\right)^T Z_1 - N\mu^{old}\right)$
$D = N\Sigma^{old} - P + \mu^{old}Y_1^T + \left(\mu^{old}Y_1^T\right)^T + Q^T\beta^{old} +$
$\left(Q^T\beta^{old}\right)^T - N\mu^{old}\left(\mu^{old}\right)^T - \mu^{old}(Z_1)^T\beta^{old} -$
$\left(\mu^{old}(Z_1)^T\beta^{old}\right)^T - \left(\beta^{old}\right)^T S\beta^{old}$
$\Sigma^{new} \leftarrow \Sigma^{old} + \rho\left(-\Sigma_1 D\Sigma_1 + \frac{1}{2}\,\mathtt{diag}\,(\Sigma_1 D\Sigma_1)\right)$
evaluate $\mathcal{C}(\delta)$
$\beta^{old} \leftarrow \beta^{new}$, $\mu^{old} \leftarrow \mu^{new}$, $\Sigma^{old} \leftarrow \Sigma^{new}$
until $\mathcal{C}(\delta)$
Output: β^{new}, μ^{new}, Σ^{new}.

4. Experimental analysis

Being given that the model $(\beta_0, \mu_0, \Sigma_0)$ given by (3.4) is not theoretically guaranteed as the best model from the point of view of maximum likelihood principle, we have performed a long series of tests aiming to derive conclusions concerning the performance of the proposed method on experimental way. The test examples x_i's were randomly generated from n-dimensional Gaussian repartition $\mathcal{N}(\mu_1, \Sigma_1)$. The target responses y_i's were computed

as $y_i = \widetilde{\beta}^T x_i + \varepsilon$ for given β and ε randomly generated from known Gaussian repartition $\mathcal{N}\left(\widetilde{\mu}, \widetilde{\Sigma}\right)$.

According to the previous arguments the expression of conditional density function on the output space corresponding to each example x_i being given the model $\omega = (\beta, \mu, \Sigma)$ is

$$f\left(y|x_i, \omega\right) = \frac{\exp\left\{-\frac{1}{2}\left(y-\beta^T z_i-\mu\right)^T \Sigma^{-1}\left(y-\beta^T z_i-\mu\right)\right\}}{\sqrt{(2\pi)^m \left|\Sigma\right|}},$$

where $z_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix}$, therefore the most likely output predicted value is $y_i' = \beta^T z_i + \mu$.

In order to evaluate the quality of the resulted model we use some indicators to evaluate the overall error (see [10], [16]).

The first indicator evaluates the overall mean error of miss-prediction for the given set of example $\{(x_i, y_i) \mid 1 \leq i \leq N\}$ corresponding to each possible model ω

$$error_1 = \frac{1}{N}\sum_{i=1}^{N}\left(1 - f\left(y_i|x_i, \omega\right)\right). \tag{4.1}$$

The second indicator is a mean error computed in terms of the actual responses and the most likely predicted values,

$$error_2 = \frac{1}{N}\sum_{i=1}^{N}\|y_i-y_i'\|^2 = \frac{1}{N}\sum_{i=1}^{N}\|y_i-\beta^T z_i-\mu\|^2. \tag{4.2}$$

The third measure, of informational type, aims to evaluate the informational correlation between the input and the computed output corresponding to each model, and it is expressed in terms of the empirical mutual information.

Since $x_1, \ldots, x_N$ are randomly generated $\mathcal{N}\left(\mu_1, \Sigma_1\right)$, the probability distribution $\widetilde{p} = \left(\widetilde{p}\left(x_1\right), \ldots, \widetilde{p}\left(x_N\right)\right)$ characterizes the collection of examples

$$\widetilde{p}\left(x_j\right) = \frac{p\left(x_j\right)}{\sum_{i=1}^{N} p\left(x_i\right)},$$

where $p\left(x_j\right) = \frac{1}{\sqrt{(2\pi)^n \left|\Sigma_1\right|}} \exp\left\{\frac{1}{2}\left(x_j-\mu_1\right)^T \Sigma_1^{-1}\left(x_j-\mu_1\right)\right\}$, $1 \leq j \leq N$.

The empirical entropy of the input samples $x_1, \ldots, x_N$ is given by the Shannon entropy corresponding to $\widetilde{p}$

$$H\left(\widetilde{p}\right) = -\sum_{i=1}^{N}\widetilde{p}\left(x_i\right)\ln\widetilde{p}\left(x_i\right). \tag{4.3}$$

Using the transition probabilities

$$\widetilde{p}\,(y_j|\,x_i,\omega)=\frac{f\,(y_j|\,x_i,\omega)}{\sum\limits_{k=1}^{N}f\,(y_k|\,x_i,\omega)}\,,\quad 1\leq i,j\leq N\,,$$

we define the probability distribution $\widetilde{q}=(\widetilde{q}\,(y_1)\,,\ldots,\widetilde{q}\,(y_N))$ on the set of target responses by

$$\widetilde{q}\,(y_j)=\sum_{i=1}^{N}\widetilde{p}\,(x_i)\,\widetilde{p}\,(y_j|\,x_i,\omega)\;,\quad 1\leq j\leq N\,,$$

and let

$$H\,(\widetilde{q})=-\sum_{i=1}^{N}\widetilde{q}\,(y_i)\ln\widetilde{q}\,(y_i)\;, \tag{4.4}$$

be the empirical Shannon entropy of the set of **S**'s outputs.

Using the well-known expression of the relative information (see [3]), the empirical relative information we defined by

$$\mathcal{I}\,(\mathcal{S}_N)=H\,(\widetilde{q})-\sum_{i=1}^{N}\sum_{j=1}^{N}\widetilde{p}\,(x_i)\,\widetilde{p}\,(y_j|\,x_i,\omega)\ln\widetilde{p}\,(y_j|\,x_i,\omega)\,.$$

The fourth measure is based on the relative entropy (Kullback-Leibler). Being given two probability distributions $p=(p_1,\ldots,p_d)$ and $q=(q_1,\ldots,q_d)$ such that q is absolutely continuous with respect to p, the relative entropy is defined by

$$\mathcal{K}\,(p,q)=\sum_{i=1}^{d}p_i\ln\left(\frac{p_i}{q_i}\right).$$

Using straightforward computation one can prove (see [3]) that

$$\mathcal{K}\,(p,q)\geq 0 \text{ and } \mathcal{K}\,(p,q)=0 \text{ if and only if } p=q.$$

In our work we introduce some indicators of Kullback-Leibler type in order to evaluate the quality of the current computed model $\omega=(\beta,\mu,\Sigma)$ at each iteration of the procedure **MLE_gradient_ascent** with respect to the quasi-optimal model $\omega_0=(\beta_0,\mu_0,\Sigma_0)$ given by Theorem 3.3.

In case of a classification problem, the outputs $y_1,\ldots,y_N$ represent the labels of the provenance classes corresponding to the inputs. The informational distance from any model ω to ω_0 computed on the basis of $\mathcal{S}_N$ can be expressed many ways. In our work we introduced the following informational distances of relative entropy type.

Because for each example $(x_i, y_i) \in \mathcal{S}_N$, $f(y_i|\, x_i, \omega)$ can be viewed as a strength degree of the association between the input x_i and the output y_i in the model ω, the probability distribution on $\mathcal{S}_N$ defined by

$$\bar{p}(y_i|\, x_i, \omega) = \frac{f(y_i|\, x_i, \omega)}{\sum_{k=1}^{N} f(y_k|\, x_k, \omega)}, \quad 1 \le i \le N,$$

characterizes in some way each possible model. Also, the Kullback-Leibler-like indicator KL_1 measures the informational distance between the models ω and ω_0 with respect to the input-output dependency revealed by $\mathcal{S}_N$

$$KL_1 = \mathcal{K}\left((\bar{p}(y_i|\, x_i, \omega_0))_{1\le i\le N}, (\bar{p}(y_i|\, x_i, \omega))_{1\le i\le N}\right) = \sum_{i=1}^{N} \bar{p}(y_i|\, x_i, \omega_0) \ln\left(\frac{\bar{p}(y_i|\, x_i, \omega_0)}{\bar{p}(y_i|\, x_i, \omega)}\right).$$

Also, for each model ω the following empirical distribution on the set $\{y_1, \ldots, y_N\}$ can be defined in a natural way,

$$\widetilde{\bar{p}}(y_i|\, \omega) = \frac{\widetilde{p}(y_i|\, \omega)}{\sum_{j=1}^{N} \widetilde{p}(y_j|\, \omega)}$$

where $\widetilde{p}(y_j|\, \omega) = \sum_{k=1}^{N} \widetilde{p}(x_k)\, f(y_j|\, x_k, \omega)$, $1 \le j \le N$. The informational distance of the model ω to ω_0 can be also expressed by

$$KL_2 = \mathcal{K}\left(\left(\widetilde{\bar{p}}(y_i|\, \omega_0)\right)_{1\le i\le N}, \left(\widetilde{\bar{p}}(y_i|\, \omega)\right)_{1\le i\le N}\right) = \sum_{i=1}^{N} \widetilde{\bar{p}}(y_i|\, \omega_0) \ln\left(\frac{\widetilde{\bar{p}}(y_i|\, \omega_0)}{\widetilde{\bar{p}}(y_i|\, \omega)}\right).$$

Although $(f(y_i|\, x_i, \omega))_{1\le i\le N}$ is not a probability distribution on $\mathcal{S}_N$, Kullback-Leibler like expression denoted KL_3 can be introduced as a measure of the quality of ω with respect to ω_0 from the point of view of the input-output dependencies represented by $\mathcal{S}_N$

$$KL_3 = \mathcal{K}\left((f(y_i|\, x_i, \omega_0))_{1\le i\le N}, (f(y_i|\, x_i, \omega))_{1\le i\le N}\right) = \sum_{i=1}^{N} f(y_i|\, x_i, \omega_0) \ln\left(\frac{f(y_i|\, x_i, \omega_0)}{f(y_i|\, x_i, \omega)}\right).$$

The indicator, $\frac{1}{N}\sum_{i=1}^{N} f(y_i|\, x_i, \omega)$ is a measure of the average strength degree of the input-output associations in $\mathcal{S}_N$. Obviously, using a well-known

inequality we get

$$\mathcal{K}\left(\left(f\left(y_i|\,x_i,\omega_0\right)\right)_{1\leq i\leq N},\left(f\left(y_i|\,x_i,\omega\right)\right)_{1\leq i\leq N}\right)\geq \sum_{i=1}^{N}\left(f\left(y_i|\,x_i,\omega_0\right)-f\left(y_i|\,x_i,\omega\right)\right)\overset{not}{=}DKL\,.$$

Another informational distance can be defined by averaging the particular KL-type distances corresponding to each of the inputs $x_1,\ldots,x_N$. For each input x_j

$$\widetilde{p}_j\left(y_i|\,\omega\right)=\frac{f\left(y_i|\,x_j,\omega\right)}{\sum_{k=1}^{N}f\left(y_k|\,x_j,\omega\right)},$$

represents the probability of predicting y_i as being the provenance class of x_j being given the model ω. Obviously, for each x_j, $\left(\widetilde{p}_j\left(y_i|\,\omega_0\right)\right)_{1\leq i\leq N}$ is a probability distribution on the space of class labels and

$$\mathcal{K}\left(\left(\widetilde{p}_j\left(y_i|\,\omega_0\right)\right)_{1\leq i\leq N},\left(\widetilde{p}_j\left(y_i|\,\omega\right)\right)_{1\leq i\leq N}\right)$$

can be taken as measure of the distance from ω to ω_0 from the point of view of the particular input x_j. The overall informational distance KL_4 is defined as an average of them

$$KL_4=\sum_{j=1}^{N}\bar{\bar{p}}\left(x_j\right)\mathcal{K}\Big(\left(\widetilde{p}_j(y_i|\,\omega_0)\right)_{1\leq i\leq N},\left(\widetilde{p}_j(y_i|\,\omega)\right)_{1\leq i\leq N}\Big)= \sum_{j=1}^{N}\bar{\bar{p}}\left(x_j\right)\sum_{i=1}^{N}\widetilde{p}_j\left(y_i|\,\omega\right)\ln\left(\frac{\widetilde{p}_j\left(y_i|\,\omega_0\right)}{\widetilde{p}_j\left(y_i|\,\omega\right)}\right),$$

where $\bar{\bar{p}}\left(x_j\right)=p\left(x_j\right)/\left(\sum_{i=1}^{N}p\left(x_i\right)\right)$, $p\left(x_j\right)$ being the probability that $\mathbf{G}$ generates the input x_j.

For instance, if the inputs are generated by randomly sampling from n-dimensional Gaussian repartition $\mathcal{N}\left(\mu_1,\Sigma_1\right)$ (μ_1,Σ_1 known), then

$$p\left(x_j\right)=\frac{\exp\left\{-\frac{1}{2}\left(x_j-\mu_1\right)^T\Sigma_1^{-1}\left(x_j-\mu_1\right)\right\}}{\sqrt{\left(2\pi\right)^n\left|\Sigma_1\right|}}.$$

In cases when the generating mechanism used by $\mathbf{G}$ is not known by the observer, then according to principle of maxim entropy the values $p\left(x_j\right)=\frac{1}{N}$ should be used in the expression of KL_4.

In the case of regression type problems, the outputs of **S** can be, in principle, any m-dimensional tuple from $\mathbb{R}^m$. In such cases for each generated input x_i the KL-type distance from the model ω to ω_0 is expressed by

$$\mathcal{K}\left(f\left(\cdot|x_i,\omega_0\right),f\left(\cdot|x_i,\omega\right)\right)=\int_{\mathbb{R}^m} f\left(y|x_i,\omega_0\right)\ln\left(\frac{f\left(y|x_i,\omega_0\right)}{f\left(y|x_i,\omega\right)}\right)\mathrm{d}y\,,$$

where $f\left(\cdot|x_i,\omega_0\right)$ and $f\left(\cdot|x_i,\omega\right)$ are the conditional densities functions on $\mathbb{R}^m$ corresponding to x_i being given the models ω_0, ω respectively.

In case f_1, and f_2 are the density functions of the m-dimensional Gaussian repartitions $\mathcal{N}\left(\mu_1,\Sigma_1\right)$ and $\mathcal{N}\left(\mu_2,\Sigma_2\right)$ respectively, the expression of the Kullback-Leibler measure is

$$\mathcal{K}\left(f_1,f_2\right)=\int_{\mathbb{R}^m} f_1(y)\ln\left(\frac{f_1(y)}{f_2(y)}\right)\mathrm{d}y=-\frac{m}{2}+\frac{1}{2}\ln\left(\frac{|\Sigma_2|}{|\Sigma_1|}\right)+\frac{1}{2}\operatorname{tr}\left(\Sigma_2^{-1}\Sigma_1\right)+\frac{1}{2}\left(\mu_1-\mu_2\right)^T\Sigma_2^{-1}\left(\mu_1-\mu_2\right)\,.$$

In our work, being given the model $\omega=(\beta,\mu,\Sigma)$, for each input x_i the conditional density function is also of normal type $f\left(\cdot|x_i,\omega\right)\sim\mathcal{N}\left(\beta^Tx_i+\mu,\Sigma\right)$. Taking $f_1=f\left(\cdot|x_i,\omega_0\right)$ and $f_2=f\left(\cdot|x_i,\omega\right)$, in the expression of $\mathcal{K}\left(f_1,f_2\right)$ we get

$$\mathcal{K}\left(f\left(\cdot|x_i,\omega_0\right),f\left(\cdot|x_i,\omega\right)\right)=-\frac{m}{2}+\frac{1}{2}\ln\left(\frac{|\Sigma|}{|\Sigma_0|}\right)+\frac{1}{2}\operatorname{tr}\left(\Sigma^{-1}\Sigma_0\right)+\frac{1}{2}\left(\left(\beta_0-\beta\right)^Tx_i+\mu_0-\mu\right)^T\Sigma^{-1}\left(\left(\beta_0-\beta\right)^Tx_i+\mu_0-\mu\right),$$

that is the average KL-type measure is

$$\mathcal{K}\left(\omega_0,\omega\right)=\sum_{i=1}^N\bar{\bar{p}}\left(x_i\right)\mathcal{K}\left(f\left(\cdot|x_i,\omega_0\right),f\left(\cdot|x_i,\omega\right)\right)=-\frac{m}{2}+\frac{1}{2}\ln\left(\frac{|\Sigma|}{|\Sigma_0|}\right)+\frac{1}{2}\sum_{i=1}^N\bar{\bar{p}}\left(x_i\right)\left(\left(\beta_0-\beta\right)^Tx_i+\mu_0-\mu\right)^T\Sigma^{-1}\left(\left(\beta_0-\beta\right)^Tx_i+\mu_0-\mu\right).$$

We performed a long series of tests aiming to establish conclusions concerning the efficiency of the **MLE_gradient_ascent** and the quality of the quasi-optimal model given by Theorem 3.3. All tests proved quick convergence properties toward to the quasi-optimal model ω_0. The results of some tests are presented in Table 1 and Table 2.

Test 1. The settings are $n=2$, $m=1$, $\mu_1=\begin{pmatrix}1\\2\end{pmatrix}$, $\Sigma_1=\begin{pmatrix}1&0\\0&1\end{pmatrix}$, $\widetilde{\beta}=\begin{pmatrix}0&2&4\end{pmatrix}^T$, $\widetilde{\mu}=0.25$ and $\widetilde{\Sigma}=1$. Some of the results for data of different sizes N, corresponding to the quasi-optimal model ω_0 computed for each data set are summarized in Table 1.

Several tests aimed to establish conclusions concerning the generalization capacity of the quasi-optimal model ω_0. For instance, in case of a training

sequence of volume $N = 100$, the computed quasi-optimal model $\omega_0 = (\beta_0, \mu_0, \Sigma_0)$ is

$$\beta_0 = \begin{pmatrix} 0 & 2.614 & 4.077 \end{pmatrix}^T, \ \mu_0 = -0.218, \ \Sigma_0 = 0.457\,.$$

The values of some of the previously introduced indicators for new test data of different sizes are summarized in Table 2.

Table 1: *Model evaluation in case $n = 2$, $m = 1$ and for different volumes of learning data.*

N	$error_1$	$error_2$	$H(\widetilde{p})$	$H(\widetilde{q})$	$\mathcal{I}(\mathcal{S}_N)$
10	0.603	0.533	2.214	2.205	1.335
20	0.702	0.802	2.767	2.796	1.304
50	0.711	0.892	3.768	3.831	1.226
100	0.712	0.950	4.477	4.524	1.233
200	0.716	0.959	5.098	5.188	1.221
300	0.721	1.019	5.510	5.590	1.106
400	0.713	0.953	5.822	5.897	1.155
500	0.725	1.025	6.023	6.113	1.150

Table 2: *Model evaluation in case $n = 2$, $m = 1$, $N = 100$ and for different volumes of test samples.*

N_{test}	$error_1$	$error_2$	$H(\widetilde{p})$	$H(\widetilde{q})$	$\mathcal{I}(\mathcal{S}_N)$
15	0.706	1.149	2.458	2.582	1.346
30	0.710	1.257	3.243	3.276	1.367
50	0.683	1.039	3.752	3.829	1.573
70	0.650	1.136	4.146	4.172	1.573
90	0.707	1.292	4.309	4.400	1.637
100	0.709	1.276	4.376	4.483	1.662
200	0.715	1.361	5.124	5.235	1.616
300	0.719	1.375	5.547	5.633	1.579

Also, in case of a training sequence of volume $N = 20$, the computed quasi-optimal model is

$$\beta_0 = \begin{pmatrix} 0 & 1.643 & 3.894 \end{pmatrix}^T, \ \mu_0 = 2.147, \ \Sigma_0 = 0.620\,,$$

the values of the Kullback-Leibler-like indicators DKL and KL_i, $i = 1, 2, 3, 4$ during the adaptive learning procedure **MLE_gradient_ascent** are summarized in Table 3.

Test 2. The settings are $n = 2$, $m = 2$, $\mu_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, $\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\widetilde{\beta} = \begin{pmatrix} 0 & 2 & 4 \\ 0 & 1 & 5 \end{pmatrix}$, $\widetilde{\mu} = \begin{pmatrix} 0.25 \\ 0.25 \end{pmatrix}$ and $\widetilde{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. Some of the results for data of

different sizes N, corresponding to the quasi-optimal model ω_0 computed for each data set are summarized in Table 4.

Table 3: *Model evaluation based on relative entropy Kullback-Leibler, in case* $n=2$, $m=1$, $N=20$.

Iteration	KL_1	KL_2	KL_3	DKL	KL_4
2	0.898	0.243	29.234	6.697	2.134
10	0.110	0.139	13.769	5.881	1.261
50	0.093	0.041	9.964	5.133	1.025
100	0.093	0.040	9.914	5.119	1.018
200	0.093	0.040	9.810	5.091	1.005
500	0.092	0.038	9.462	4.997	0.966
900	0.091	0.035	8.861	4.824	0.902
1500	0.085	0.026	7.207	4.269	0.726
1700	0.078	0.021	5.930	3.751	0.583
1900	0.001	0.000	0.174	0.167	0.003

Table 4: *Model evaluation in case* $n = 2$, $m = 2$ *and for different volumes of learning data.*

N	$error_1$	$error_2$	$H(\widetilde{p})$	$H(\widetilde{q})$	$\mathcal{I}(\mathcal{S}_N)$
15	0.887	1.436	2.500	2.483	1.610
20	0.877	1.282	2.816	2.720	1.521
50	0.892	1.473	3.748	3.675	1.810
100	0.926	2.030	4.441	4.369	1.583
200	0.908	1.774	5.090	5.065	1.704
300	0.923	2.072	5.513	5.495	1.638
400	0.919	1.982	5.810	5.768	1.661
500	0.918	1.954	6.014	5.987	1.763

Several tests aimed to establish conclusions concerning the generalization capacity of the quasi-optimal model. For instance, in case of a training sequence of volume $N = 100$, the computed quasi-optimal model is

$$\beta_0=\begin{pmatrix} 0 & 0 \\ 1.87 & 1.01 \\ 4.13 & 5.04 \end{pmatrix},\ \mu_0=\begin{pmatrix} 0.27 \\ 0.10 \end{pmatrix},\ \Sigma_0=\begin{pmatrix} 0.67 & 0 \\ 0 & 0.89 \end{pmatrix},$$

the results are summarized in Table 5.

Also, in case of a training sequence of volume $N = 10$, the computed quasi-optimal model is

$$\beta_0=\begin{pmatrix} 0 & 0 \\ 1.80 & 0.89 \\ 3.77 & 5.96 \end{pmatrix},\ \mu_0=\begin{pmatrix} 1.14 \\ -1.23 \end{pmatrix},\ \Sigma_0=\begin{pmatrix} 1.35 & 0.02 \\ 0.02 & 1.27 \end{pmatrix},$$

and the results are summarized in Table 6.

Table 5: *Model evaluation in case $n = 2$, $m = 2$, $N = 100$ and for different volumes of test samples.*

N_{test}	$error_1$	$error_2$	$H(\widetilde{p})$	$H(\widetilde{q})$	$\mathcal{I}(\mathcal{S}_N)$
15	0.917	2.154	2.542	2.311	1.277
30	0.912	1.874	3.190	3.116	1.763
50	0.890	1.438	3.764	3.699	1.627
70	0.910	2.216	4.042	3.889	1.772
90	0.907	1.881	4.349	4.277	1.769
100	0.912	2.014	4.432	4.292	1.783
200	0.906	1.938	5.098	5.062	1.698
300	0.902	1.860	5.512	5.457	1.822

Table 6: *Model evaluation based on relative entropy Kullback-Leibler, in case $n=2$, $m=2$, $N=10$.*

Iteration	KL_1	KL_2	KL_3	DKL	KL_4
27	18.021	0.506	13.396	0.593	8.753
35	0.754	0.182	1.568	0.507	1.134
50	0.267	0.170	1.047	0.464	0.909
100	0.120	0.145	0.823	0.429	0.677
300	0.078	0.094	0.625	0.372	0.501
700	0.049	0.068	0.521	0.336	0.407
1000	0.045	0.058	0.471	0.314	0.352
1200	0.043	0.051	0.428	0.293	0.309
1500	0.036	0.036	0.333	0.243	0.223
1700	0.025	0.020	0.219	0.173	0.129
1800	0.013	0.008	0.125	0.106	0.055
1900	0.003	0.001	0.028	0.026	0.004

5. Conclusion and future work

In this paper we present two linear regressive models: one is a linear regressive model determined by minimizing the arithmetic mean of square errors, and the other is a probabilistic model which includes effects of the latent variables and the noise. For both models we found solutions for the parameters. For two special cases, in Theorem 3.1 and Theorem 3.2 we give the exact expressions of the parameters for proposed probabilistic model. The research aimed to establish theoretical conclusions concerning the extent to which a multivariate noisy system can be learned on the basis of a finite size sequence of observations. The theoretical development has led to the conclusions formulated in Theorem 3.3 and in Theorem 3.4. The unique critical point given by Theorem 3.3 has not been proved yet as being the optimal model from the point of view of the maximum likelihood principle, the op-

timality being partially confirmed by Theorem 3.4. However, a long series of tests pointed out quick convergence toward this quasi-optimal solution.

The research aiming to extend these results to more general types of systems is still in progress and the results will be published in the near future.

Acknowledgments

The research was developed in the framework of the Ph.D. program at University of Piteşti, Romania, under the supervision of Professor Luminiţa State.

References

[1] E. ALPAYDIN, *Introduction to Machine Learning*, The MIT Press, Massachusetts, 2010.

[2] V. CHERKASSKY and F. MULIER, *Learning from Data Concepts, Theory, and Methods*, second edition, John Wiley & Sons, Inc., 2007.

[3] T.M. COVER and J.A. THOMAS, *Elements of information theory*, second edition, John Wiley & Sons, Inc., 2006.

[4] T. HASTIE, R. TIBSHIRANI and J. FRIEDMAN, *The Elements of Statistical Learning - Data mining, Inference and Prediction*, second edition, Springer-Verlag, 2009.

[5] S. HAYKIN, *Neural Networks:A Comprehensive Foundation*, Prentice Hall, 2000.

[6] A.J. IZENMAN, *Modern Multivariate Statistical Techniques: Regressian, Classification and Manifold Learning*, Springer, 2008.

[7] A.K. JAIN, R. DUIN and J. MAO, Statistical Pattern Recognition: A Review, *IEEE Trans. Pattern Anal. Mach. Intell.*, **22** (2000), 4-37.

[8] S. MARSLAND, *Machine Learning: An Algorithmic Perspective*, CRC Press, Taylor & Francis Group, Boca Raton - London - New York, 2009.

[9] K.E. MULLER and P.W. STEWART, *Linear Model Theory - Univariate, Multivariate and Mixed Models*, Wiley-Interscience, 2006.

[10] I. PARASCHIV-MUNTEANU, Model-free approaches in learning the multivariate linear regressive models, *An. Univ. Craiova Ser. Mat. Inform.*, **38** (2011), 87-94.

[11] I. PARASCHIV-MUNTEANU, Regressive models in supervised learning from data, *The 7th Congress of Romanian Mathematicians*, June 29 - July 5, 2011, Braşov, Romania, Abstracts, pp. 89.

[12] I. PARASCHIV-MUNTEANU and L. STATE, Learning from Data using Multivariate Linear Models, *Coping with Complexity, COPCOM 2011*, Editura Casa Cărţii de Ştiinţă, Eds.: D. Dumitrescu, R.I. Lung, L. Cremene, pp. 20-31, 2011.

[13] A.C. RENCHER, *Methods of Multivariate Analysis*, Wiley-Interscience, 2002.

[14] S. SHARMA, *Applied Multivariate Techniques*, John Wiley & Sons, Inc., 1996.

[15] L. STATE and I. PARASCHIV-MUNTEANU, *Introducere in teoria statistica a recunoasterii formelor*, Editura Universitatii din Pitesti, Romania, 2009 (in Romanian).

[16] L. STATE and I. PARASCHIV-MUNTEANU, A probabilistic model-free approach in learning multivariate noisy linear systems, *Proceedings 13th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing SYNASC-2011*, Editors D. Wang, V. Negru, T. Ida, T. Jebelean, D. Petcu, S. Watt and D. Zaharie, IEEE Computer Society Conference Publishing Services, pp. 239-246.

[17] K. TAKEAKI and K. HIROSHI, *Generalized Least Squares*, John Wiley & Sons, Inc., 2004.

Iuliana Paraschiv-Munteanu

University of Bucharest, Faculty of Mathematics and Computer Science

14 Academiei Street, 010014 Bucharest, Romania

E-mail: pmiulia@fmi.unibuc.ro