Digital Signal Processing 19 (2009) 715–725

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

On vector averaging over the unit hypersphere

Simone Fiori

Dipartimento di Ingegneria Biomedica, Elettronica, e Telecomunicazioni (DIBET), Facoltà di Ingegneria, Università Politecnica delle Marche, Via Brecce Bianche, Ancona I-60131, Italy

ARTICLE INFO

Article history:
Available online 17 July 2008

Keywords:
Divergence
Mean-value computation on curved spaces
Optimization on manifolds
Unit hypersphere

ABSTRACT

Sample averaging is a commonly used way to smooth out irregularities of data and to get rid of random fluctuations in measurements analysis. In adaptive signal processing, where an adaptive system learns its own parameters in order to perform a predefined task, the learnt parameters-pattern may depend on the initial learning state and on the fluctuations of the statistical features of the input signals to the system. In adaptive system learning, averaging may be employed as a method to merge several learnt parameters-patterns in order to get a better representative pattern. Even in the case of scalar parameters, the concept of averaging is not uniquely defined as scalar parameters spaces may exhibit a rich structure to be dealt with. The case of multiple parameter patterns where single parameters are mutually constrained to each other may exhibit an even richer structure. In the present paper, we deal with the case of parameters-patterns belonging to the unit hypersphere and develop an averaging technique based on the differential geometrical structure of such a curved space. Numerical experiments illustrate the behavior of the developed averaging algorithm.

1. Introduction

Averaging is a straightforward way to smooth out data in measurement analysis. Available data from an experiment may be regarded as samples drawn from a random variable and their average value may thus be regarded as the expected value of the associated random variable.

A common way to empirically estimate the expected value of a *real-valued* random variable is to compute the *arithmetic mean* of the available samples. It is known from basic statistics that, if the expected value of the random variable at hand exists, arithmetic averaging estimates the true expected value in an unbiased manner and has the property of minimizing the sum of the squared differences between the available samples and the mean-value estimate. Such minimal sum of squared differences may then be regarded as a measure of dispersion of the samples around the mean value, which is termed 'variance' of the data. In formulas, if we denote the available samples by $x_k \in \mathbb{R}$, with $k = 1, \ldots, N$, the expected value as $\mu \in \mathbb{R}$ and the samples variance by σ^2, we have:

$$\mu \stackrel{\text{def}}{=} \arg\min_{x\in\mathbb{R}} \frac{1}{N}\sum_{k=1}^{N}(x - x_k)^2, \qquad \sigma^2 \stackrel{\text{def}}{=} \min_{x\in\mathbb{R}} \frac{1}{N}\sum_{k=1}^{N}(x - x_k)^2, \tag{1}$$

E-mail address: s.fiori@univpm.it.

1051-2004/$ – see front matter
doi:10.1016/j.dsp.2008.07.005

namely:

$$\mu = \frac{1}{N}\sum_{k=1}^{N} x_k, \qquad \sigma^2 = \frac{1}{N}\sum_{k=1}^{N}(\mu - x_k)^2. \tag{2}$$

The law of large numbers demonstrates that, under mild conditions, as the size N of the sample-set gets larger, the variance of this estimate gets smaller [2].

In cases of interest, the space $\mathbb{X}$ that the measurements/samples x_k belong to, may exhibit a more involved structure than the real line $\mathbb{R}$. Such a richer structure influences the way that averaging should be conceived of. In order to get acquainted with this concept, let us consider the case that the available samples to be averaged belong to the real half-line $\mathbb{R}^+ \stackrel{\text{def}}{=} \{x \in \mathbb{R} \mid x > 0\}$ and let us figure out how to define a mean value and a dispersion measure of the sample-values around the mean value on the set $\mathbb{R}^+$. (For notational convenience, we also define the half-line $\mathbb{R}_0^+ \stackrel{\text{def}}{=} \{x \in \mathbb{R} \mid x \geqslant 0\}$.)

The basic tool needed to extend the definitions in (1) to a different sample-space $\mathbb{X}$ is a measure of 'how far' two objects in $\mathbb{X}$ lie one to another. As an example of ready applicability to the case of the half-line $\mathbb{R}^+$, we may recall here the concept of Bregman divergence $D_\varphi : \mathbb{X} \times \text{int}(\mathbb{X}) \to \mathbb{R}_0^+$, defined as [3]:

$$D_\varphi(x, y) \stackrel{\text{def}}{=} \varphi(x) - \varphi(y) - \varphi'(y)(x - y), \tag{3}$$

where $\varphi : \mathbb{X} \subseteq \mathbb{R} \to \mathbb{R}$ is a strictly convex function with continuous first derivative and $\text{int}(\mathbb{X})$ denotes the interior of the set $\mathbb{X}$. It is worth noting at this point that (Bregman) divergences are asymmetric in general, i.e., $D_\varphi(x, y) \neq D_\varphi(y, x)$. By definition, the following important properties hold:

$$D_\varphi(x, y) \geqslant 0 \quad \forall (x, y) \in \mathbb{X} \times \text{int}(\mathbb{X}), \qquad D_\varphi(y, y) = 0 \quad \forall y \in \text{int}(\mathbb{X}). \tag{4}$$

We may now extend the simple definition in (1) to what we may refer to as 'Bregman mean' and 'Bregman variance' on the set $\mathbb{R}^+$:

$$\mu \stackrel{\text{def}}{=} \arg\min_{x \in \mathbb{R}^+} \frac{1}{N}\sum_{k=1}^{N} D_\varphi(x, x_k), \qquad \sigma^2 \stackrel{\text{def}}{=} \min_{x \in \mathbb{R}^+} \frac{1}{N}\sum_{k=1}^{N} D_\varphi(x, x_k). \tag{5}$$

The definition given in (5) does not ensure the Bregman mean to be unique.

Some examples of choice of function φ are in order (as a recent reference for the Bregman divergence and the special cases cite below, we take the report [16]):

- *Squared Euclidean distance.* The simplest example of Bregman divergence arises when $\varphi(x) = x^2$. In this case, it is readily verified that $D_\varphi(x, y) = (x - y)^2$, therefore the Bregman divergence collapses into a squared Euclidean distance. In this case, mean-value and sample-variance are computed as:

  $$\mu \stackrel{\text{def}}{=} \arg\min_{x \in \mathbb{R}^+} \frac{1}{N}\sum_{k=1}^{N}(x - x_k)^2, \qquad \sigma^2 \stackrel{\text{def}}{=} \min_{x \in \mathbb{R}^+} \frac{1}{N}\sum_{k=1}^{N}(x - x_k)^2.$$

  Not surprisingly, the solution of the minimization problem coincides to solutions (2), namely, the mean value of positive real numbers may be computed as their arithmetic mean and the associated variance as the arithmetic mean of the squared differences $(\mu - x_k)^2$.
- *Itakura–Saito divergence [16].* A more interesting example arises when $\varphi(x) = -\log x$. In this case, straightforward calculations show that $D_\varphi(x, y) = \frac{x}{y} - 1 - \log\frac{x}{y}$. In this case, the mean value and the sample-variance may be computed as:

  $$\mu \stackrel{\text{def}}{=} \arg\min_{x \in \mathbb{R}^+} \frac{1}{N}\sum_{k=1}^{N}\left(\frac{x}{x_k} - 1 - \log\frac{x}{x_k}\right),$$

  $$\sigma^2 \stackrel{\text{def}}{=} \min_{x \in \mathbb{R}^+} \frac{1}{N}\sum_{k=1}^{N}\left(\frac{x}{x_k} - 1 - \log\frac{x}{x_k}\right).$$

  By setting the first derivative of the function to optimize to zero and solving for the variable x, we obtain:

  $$\frac{1}{\mu} = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{x_k}, \qquad \sigma^2 = \frac{1}{N}\sum_{k=1}^{N}\log\frac{x_k}{\mu}.$$

  The obtained expression for the mean value is known as 'harmonic mean' (or 'subcontrary mean') [5]. The obtained expression for the variance is, clearly, well-defined only if all the available samples posses equal sign, as it is the case for samples belonging to the half-line $\mathbb{R}^+$.

- *Kullback–Leibler divergence [16].* Another interesting example arises when we set $\varphi(x) = x \log x$. In this case, calculations show that $D_\varphi(x, y) = x \log \frac{x}{y} - x + y$, which is the well-known Kullback–Leibler divergence.[1] In this case, the mean value and the sample-variance may be computed as:

$$\mu \stackrel{\text{def}}{=} \arg\min_{x \in \mathbb{R}^+} \frac{1}{N} \sum_{k=1}^{N} \left(x \log \frac{x}{x_k} - x + x_k \right),$$

$$\sigma^2 \stackrel{\text{def}}{=} \min_{x \in \mathbb{R}^+} \frac{1}{N} \sum_{k=1}^{N} \left(x \log \frac{x}{x_k} - x + x_k \right).$$

By solving the above optimization problem, we obtain:

$$\mu = \sqrt[N]{x_1 \cdot x_2 \cdot x_3 \cdots x_N}, \qquad \sigma^2 = \frac{1}{N} \sum_{k=1}^{N} (x_k - \mu).$$

The obtained expression for the mean value is known in the statistics literature as 'geometric mean' [5].

It is worth observing that, in the special case of negligible differences between the positive numbers to be averaged, namely, when $x_1 = x_2 = \cdots = x_N$, using all considered Bregman divergences leads to the intuitive results $\mu = x_1 = x_2 = \cdots = x_N$ and $\sigma^2 = 0$.

It is also worth remarking that the quantity that is here referred to as sample-variance in (5) is given by other authors a different interpretation (see, for instance, [1], where the quantity $\min_{x \in \mathbb{X}} \frac{1}{N} \sum_{k=1}^{N} D_\varphi(x, x_k)$ is interpreted as 'Bregman information').

From the above examples, we may learn that: (1) the notion of 'mean value' of objects in a space of observations should reflect our intuitive understanding that the mean-value is an element of the space of observations that locates amidst the available samples. Therefore, a fundamental tool in the definition of a sample mean and associate sample variance is a measure of 'how far' two elements in the sample space lie one to another; (2) the notion of 'mean value' of objects in a space depends on how the dissimilarity of such objects is measured; (3) the notion of 'variance,' that accounts for the dispersion of the objects about the mean value, changes accordingly.

In applications, the sample-set $\mathbb{X}$ may exhibit much more structure than the half-line $\mathbb{R}^+$. For example, samples may be of vector or matrix type and vectors/matrices may belong to curved spaces, such as differential manifolds, that embody possible constraints that the vector/matrix-type samples might be subjected to. In a given application, we might thus extend the optimization principle expressed in (5) to the general optimization principle:

$$\mu \stackrel{\text{def}}{=} \arg\min_{x \in \mathbb{X}} \frac{1}{N} \sum_{k=1}^{N} D(x, x_k), \qquad \sigma^2 \stackrel{\text{def}}{=} \min_{x \in \mathbb{X}} \frac{1}{N} \sum_{k=1}^{N} D(x, x_k), \tag{6}$$

where operator $D : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_0^+$ denotes a divergence.

There is no warranty that the optimization problem (6) can be solved in closed form. In the case that it cannot be coped with exactly, it would be necessary to resort to some appropriate numerical optimization method. Appropriateness of the numerical optimization algorithm is referred to the algebraic–geometrical structure of the sample space $\mathbb{X}$, which needs to be taken into account when designing any numerical optimization method.

When the divergence operator in (6) is set to be a squared distance operator, then the mean-value defined in (6) coincides to the so-termed Fréchet mean [13]. Namely, if the divergence $D(x, x_k)$ is replaced by any squared distance $d^2(x, x_k)$ in the optimization problem (6), then the resulting value μ is what is referred to as Fréchet mean-value. There is an interesting observation about the definition of the Fréchet mean, reported in the note [15], that we shall invoke later in the present paper: There appears to be no reason to assume a *squared* distance as measure of closeness between two points in a sample set but for mathematical tractability of the optimization problem. In other terms, in cases of interest, a measure of closeness between points may be set just to a (non-squared) distance and yet the optimization problem may be tractable, if not easier to cope with.

A case of interest of structured samples-set in signal processing arises when dealing with adaptive signal processors that learn how to solve a pre-defined task on the basis of input signals and a learning procedure that drives processor actions. In fact, let us imagine, to fix the ideas, that an adaptive signal processing system is described by a vector $x \in \mathbb{X}$ of parameters that the system adapts through a learning procedure. In this case, the set $\mathbb{X}$ denotes the ensemble of all feasible parameter-vectors for the system. If a given learning algorithm is launched several times on different input signals drawn from the

[1] Those readers familiar with probability and information theory will note that the general definition of Kullback–Leibler divergence differs from that of the Kullback–Leibler divergence of probability distributions. On two probability sets $\{p_{x,k}\}$ and $\{p_{y,k}\}$, in fact, it holds $D_\varphi(\{p_{x,k}\}, \{p_{y,k}\}) = \sum_k p_{x,k} \log \frac{p_{x,k}}{p_{y,k}} + \sum_k p_{y,k} - \sum_k p_{x,k}$. The last two sums, however, both equal 1, thus the quantity $D_\varphi(\{p_{x,k}\}, \{p_{y,k}\})$ reduces to the familiar expression of the Kullback–Leibler divergence.

same distribution, or the learning algorithm is launched several times from different initial parameter-vectors patterns, then the same algorithm produces a set $\{x_k\}_{k=1}^N$ of parameter-vectors that allegedly should not differ much from each other, yet different one to another. Averaging on such a set of approximate solutions to the same learning task is of use to smooth out the unavoidable fluctuations in the learning process.

In the present manuscript, we deal with the special case of averaging over the unit hypersphere $\mathbb{S}^{p-1} \stackrel{\text{def}}{=} \{x \in \mathbb{R}^p \mid x^T x = 1\}$. There are a number of signal processing algorithms that learn parameter-vectors on the manifold $\mathbb{S}^{p-1}$ as, for instance, blind deconvolution algorithms [10,11,18], one-unit independent component analysis algorithms [14], robust constrained beamforming algorithms [8] and data classification by linear discrimination based on non-Gaussianity discovery [17].

In the present paper, we briefly survey (and make use of) the differential–geometric structure of the smooth manifold $\mathbb{S}^{p-1}$. In order to define vector averaging over the unit hypersphere, we set up an appropriate optimization problem in the spirit of principles expressed, e.g., by (1) and (5). We then discuss a fixed-point-type optimization algorithm that allows finding a mean value on the manifold $\mathbb{S}^{p-1}$ and the associated variance.

2. An algorithm for vector averaging over the unit hypersphere

In the present section, we briefly survey the geometry of the unit hypersphere by recalling concepts as tangent/normal spaces, Riemannian gradient, geodesic arcs and associated geodesic distances. We then proceed to develop an optimization principle that leads to the definition of mean-vector and associated variance over the unit hypersphere embedded into a Euclidean space.

2.1. Brief survey of $\mathbb{S}^{p-1}$'s geometry

As a reference for the concepts recalled below, readers might, e.g., refer to the recent contribution [11].

At every point $x \in \mathbb{S}^{p-1}$, the linear space $T_x\mathbb{S}^{p-1}$ tangent to the hypersphere has structure $T_x\mathbb{S}^{p-1} \stackrel{\text{def}}{=} \{v \in \mathbb{R}^p \mid v^T x = 0\}$. The normal space $N_x\mathbb{S}^{p-1}$ at every point of the hypersphere, which is the orthogonal complement of the tangent space with respect to the ambient space $\mathbb{R}^p$ that the manifold $\mathbb{S}^{p-1}$ is embedded in, has structure $N_x\mathbb{S}^{p-1} \stackrel{\text{def}}{=} \{\lambda x \mid \lambda \in \mathbb{R}\}$ in the case that the ambient space is equipped with the standard Euclidean scalar product.

The smooth manifold $\mathbb{S}^{p-1}$ is turned into a Riemannian manifold by endowing it with a local scalar product. We select the natural scalar product $\langle v_1, v_2\rangle_x^{\mathbb{S}^{p-1}} \stackrel{\text{def}}{=} v_1^T v_2$.

Given a regular function $f : \mathbb{S}^{p-1} \to \mathbb{R}$, its Riemannian gradient is the vector $\nabla_x^{\mathbb{S}^{p-1}} f \in T_x\mathbb{S}^{p-1}$ that, with the above choice for a scalar product, has the structure:

$$\nabla_x^{\mathbb{S}^{p-1}} f = (I_p - xx^T)\frac{\partial f}{\partial x}, \tag{7}$$

where I_p denotes the $p \times p$ identity matrix and $\frac{\partial}{\partial x}$ denotes the component-wise partial-derivative operator (also referred to as Jacobian).

The points $x \in \mathbb{S}^{p-1}$ where the function $f : \mathbb{S}^{p-1} \to \mathbb{R}$ assumes extremal values coincides to the points where Riemannian gradient $\nabla_x^{\mathbb{S}^{p-1}} f$ vanishes to zero.

Geodesics represent the counterparts of 'straight lines' on curved spaces. On the space $\mathbb{S}^{p-1}$ embedded in the Euclidean space $\mathbb{R}^p$, a geodesic may be conceived of as a curve on which a particle, departing from the point $x \in \mathbb{S}^{p-1}$ with velocity $v \in T_x\mathbb{S}^{p-1}$, slides with constant scalar speed $\|v\|$, where $\|\cdot\|$ denotes the standard L_2 vector norm. On the hypersphere, we denote such a curve as $c_{x,v}(t)$, where the variable $t \in [0, 1]$ provides a parametrization of the curve. The equation of the geodesic may be found by observing that the acceleration of the particle is either null or normal to the tangent space at any point, namely $\frac{d^2 c_{x,v}(t)}{dt^2} \in N_x\mathbb{S}^{p-1}$. In explicit form, the equation of the geodesic on the unit hypersphere writes [6]:

$$c_{x,v}(t) = x\cos\big(\|v\|t\big) + v\sin\big(\|v\|t\big)\|v\|^{-1}. \tag{8}$$

The relationship (8) for the geodesic represents a 'great circle' on the hypersphere.

On the basis of geodesic curves, it is possible to conceive a metrization of the hypersphere. Namely, given two points $x_1, x_2 \in \mathbb{S}^{p-1}$ and a geodesic arc $c_{x_1,v}(t)$ connecting them, namely, such that $v \in T_{x_1}\mathbb{S}^{p-1}$, $c_{x_1,v}(0) = x_1$ and $c_{x_1,v}(1) = x_2$, we define 'geodesic distance between points x_1 and x_2' the quantity:

$$d_{\text{geo}}(x_1, x_2) \stackrel{\text{def}}{=} \int_0^1 \sqrt{\left\langle \frac{dc_{x_1,v}(t)}{dt}, \frac{dc_{x_1,v}(t)}{dt} \right\rangle_{x_1}^{\mathbb{S}^{p-1}}}\, dt. \tag{9}$$

From the expression (8) of the geodesic curve on the hypersphere, we have:

$$\frac{dc_{x_1,v}(t)}{dt} = -x_1\|v\|^2\sin\big(\|v\|t\big) + v\cos\big(\|v\|t\big),$$
$$x_2 = x_1\cos\big(\|v\|\big) + v\sin\big(\|v\|\big)\|v\|^{-1},$$

therefore, by using the facts that $x_1^T x_1 = 1$ and $x_1^T v = 0$, an expression for the geodesic distance is easily obtained as:

$$d_{\text{geo}}(x_1, x_2) = \arccos\left(x_1^T x_2\right), \tag{10}$$

where function 'arccos' denotes the inverse cosine function whose image is here supposed to be the interval $[0, \pi]$.

2.2. An algorithm to compute the sample-mean-value on a hypersphere and the associated variance

Having defined a measure of 'how far' points on a hypersphere are from each other, we can set up an optimization problem to define averaging over the hypersphere. In the spirit of the principle expressed in (6), of the notion of Fréchet mean and on the basis of the observation that it is needless to use squared differences in its definition, we may define the mean value and the sample-variance of a set of N samples x_k over the manifold $\mathbb{S}^{p-1}$ as:

$$\mu \stackrel{\text{def}}{=} \arg\min_{x \in \mathbb{S}^{p-1}} \frac{1}{N} \sum_{k=1}^{N} d_{\text{geo}}(x, x_k), \qquad \sigma^2 \stackrel{\text{def}}{=} \min_{x \in \mathbb{S}^{p-1}} \frac{1}{N} \sum_{k=1}^{N} d_{\text{geo}}(x, x_k). \tag{11}$$

The function to minimize is continuous, bounded and defined on a compact set, it thus admits a maximum and a minimum [19].

In order to find the minimum of function $\sum_{k=1}^{N} \arccos(x^T x_k)$, we should first calculate the Jacobian:

$$\frac{\partial}{\partial x} \sum_{k=1}^{N} \arccos\left(x^T x_k\right) = \sum_{k=1}^{N} \frac{-x_k}{\sqrt{1 - (x^T x_k)^2}},$$

then compute the Riemannian gradient (7) and set it to zero. This leads to the equation for the *geodesic mean* $\mu \in \mathbb{S}^{p-1}$:

$$\sum_{k=1}^{N} \frac{x_k}{\sqrt{1 - (\mu^T x_k)^2}} = \mu \sum_{k=1}^{N} \frac{\mu^T x_k}{\sqrt{1 - (\mu^T x_k)^2}}. \tag{12}$$

(Note that it admits only solutions that satisfy condition $\mu^T \mu = 1$, as it should clearly be.) The above non-linear equation on $\mathbb{S}^{p-1}$ cannot be solved in closed form. It is, therefore, necessary to resort to an iterative algorithm to solve it. We suggest here a fixed-point-type iterative algorithm. By dividing both sides of equation (12) by $\sum_{k=1}^{N} \frac{\mu^T x_k}{\sqrt{1-(\mu^T x_k)^2}}$ and then applying to both sides a projection operator over the unit hypersphere, it is easily gotten:

$$\mu \leftarrow \Pi_{\mathbb{S}^{p-1}} \left(\sum_{k=1}^{N} \frac{x_k}{\sqrt{1 - (\mu^T x_k)^2}} \right), \tag{13}$$

where operator $\Pi_{\mathbb{S}^{p-1}} : \mathbb{R}^p - \{0\} \to \mathbb{S}^{p-1}$ is any suitable projector over the unit hypersphere, like:

$$\Pi_{\mathbb{S}^{p-1}}(z) \stackrel{\text{def}}{=} z\left(z^T z\right)^{-\frac{1}{2}}. \tag{14}$$

The iterative algorithm (13) to find a mean-vector $\mu \in \mathbb{S}^{p-1}$ does not need any stepsize to be set.

Once the sample-mean is found on the unit hypersphere, the associated sample *geodesic variance* may be calculated as:

$$\sigma^2 = \frac{1}{N} \sum_{k=1}^{N} \arccos\left(\mu^T x_k\right). \tag{15}$$

An equivalent set of equations to implement algorithm (13) and to compute variance (15) is:

$$\begin{cases} \psi_k \stackrel{\text{def}}{=} \arccos(\mu^T x_k), \quad \mu \leftarrow \Pi_{\mathbb{S}^{p-1}}(\sum_{k=1}^{N} x_k |\sin \psi_k|^{-1}), \\ \sigma^2 = N^{-1} \sum_{k=1}^{N} \psi_k. \end{cases} \tag{16}$$

It comes from the fact that $\mu^T x_k = \cos \psi_k$, therefore $\sqrt{1 - (\mu^T x_k)^2} = |\sin \psi_k|$.

Algorithm (13) or (16) is iterative, therefore, it should be run over a suitable iteration span until convergence is achieved.

3. Exemplary numerical results

In the present section, the behavior of the proposed algorithm is illustrated via numerical simulations, both on synthetic data and on unitary vector sets generated by real algorithms.

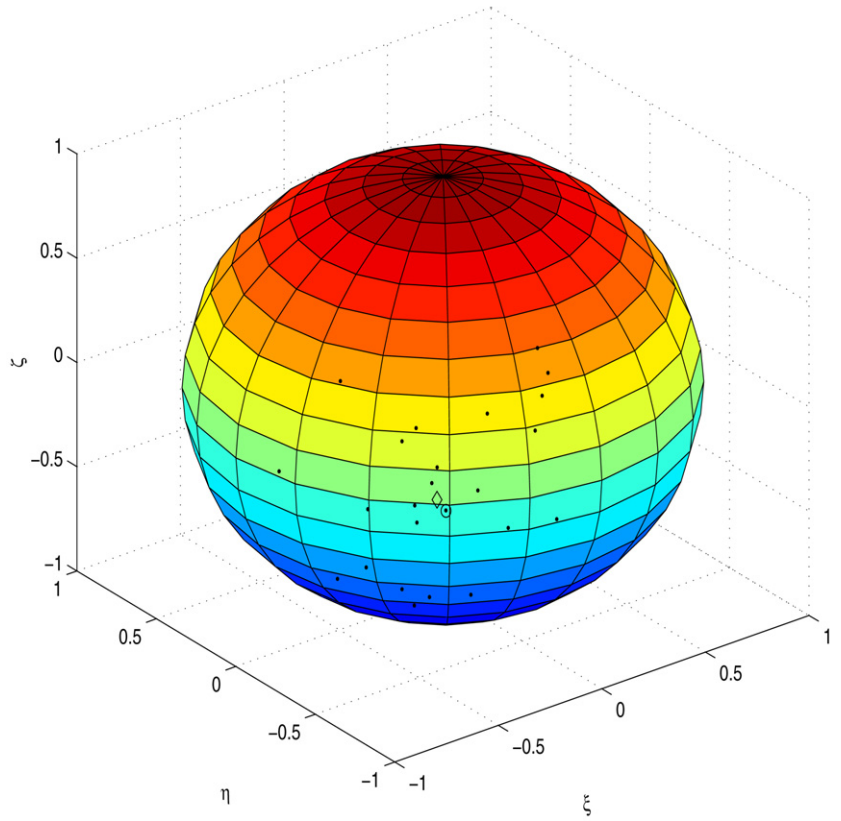

Fig. 1. Example with $p = 3$ and $N = 24$. Dots: Samples to be averaged. Open circle: Center of samples constellation. Open diamond: Computed mean value. Any vector $x \in \mathbb{S}^2$ is represented by triples of coordinates (ξ, η, ζ).

3.1. Experiments with synthetic data

The first set of experiments refer to synthetically generated unitary vectors to average. In particular, the N samples $x_k \in \mathbb{S}^{p-1}$ are generated by randomly fixing a point $x_0 \in \mathbb{S}^{p-1}$, which should be regarded as the 'center' of the sample constellation, and by producing random deviations to x_0. Details of samples generation are as follows. For each sample, a random vector $\bar{v}_k \in \mathbb{R}^p$ with Gaussian distribution for each entry is generated and then projected over $T_{x_0}\mathbb{S}^{p-1}$ by $v_k = (I_p - x_0 x_0^T)\bar{v}_k$; then, a random value t_k is generated within the interval $[0, 1]$; finally, each sample is generated by a random step of extension t_k along a geodesic arc departing from point x_0 directed along v_k, namely, as $x_k = c_{x_0, v_k}(t_k)$.

We may then measure the distances $d_{\text{geo}}(x_k, x_0)$ to quantify the deviations of the samples x_k from the center of the sample constellation over the unit hypersphere and $d_{\text{geo}}(\mu, x_0)$ to quantify the deviation of the computed mean-value from the center of the sample constellation.

The algorithm (13) was run over 10 iterations in all experiments.

3.1.1. Numerical example on a three-dimensional space

An experiment that allows visualizing the behavior of the discussed averaging algorithm pertains to the three-dimensional problem-case $p = 3$, for which we further set $N = 24$.

Fig. 1 shows the samples to be averaged, placed over the unit-hypersphere, as well as the center of samples constellation and the computed mean value. Fig. 2 compares the distances between the samples and their center with the distance between the computed mean value and the center of the sample constellation. Fig. 2 also illustrates the course of the algorithm (13) during iterations.

In this experiment, the computed mean-value is much closer to the samples center than the samples themselves.

3.1.2. Numerical example on a high-dimensional space

On a further experiment, we set $p = 10$ and $N = 49$.

Fig. 3 compares again the distances between the samples and their center with the distance between the computed mean value and the center of the sample constellation. Fig. 3 also illustrates the course of the algorithm (13) during iterations.

In this experiment, the computed mean-value is much closer to the samples center than most of the available samples.

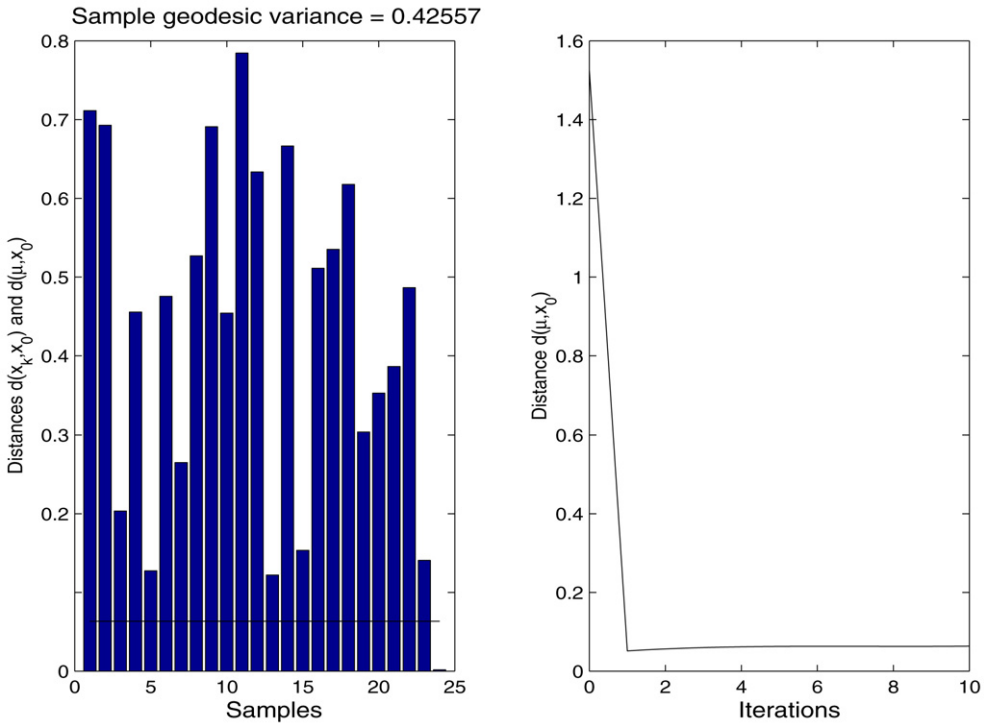

Fig. 2. Example with $p = 3$ and $N = 24$. Left-hand panel: Comparison of the distances between the samples and their center $d_{\mathrm{geo}}(x_k, x_0)$ with the distance between the computed mean value and the center of the sample constellation $d_{\mathrm{geo}}(\mu, x_0)$ (solid horizontal line). Right-hand panel: Course of the algorithm (13) during iterations.

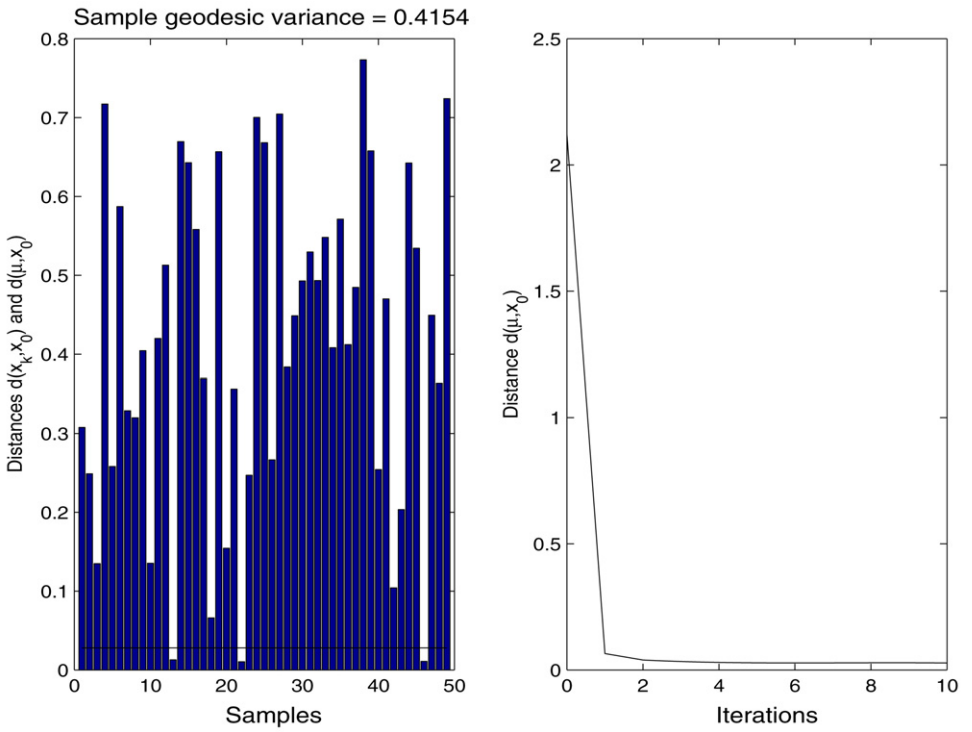

Fig. 3. Example with $p = 10$ and $N = 49$. Left-hand panel: Comparison of the distances between the samples and their center $d_{\mathrm{geo}}(x_k, x_0)$ with the distance between the computed mean value and the center of the sample constellation $d_{\mathrm{geo}}(\mu, x_0)$ (solid horizontal line). Right-hand panel: Course of the algorithm (13) during iterations.

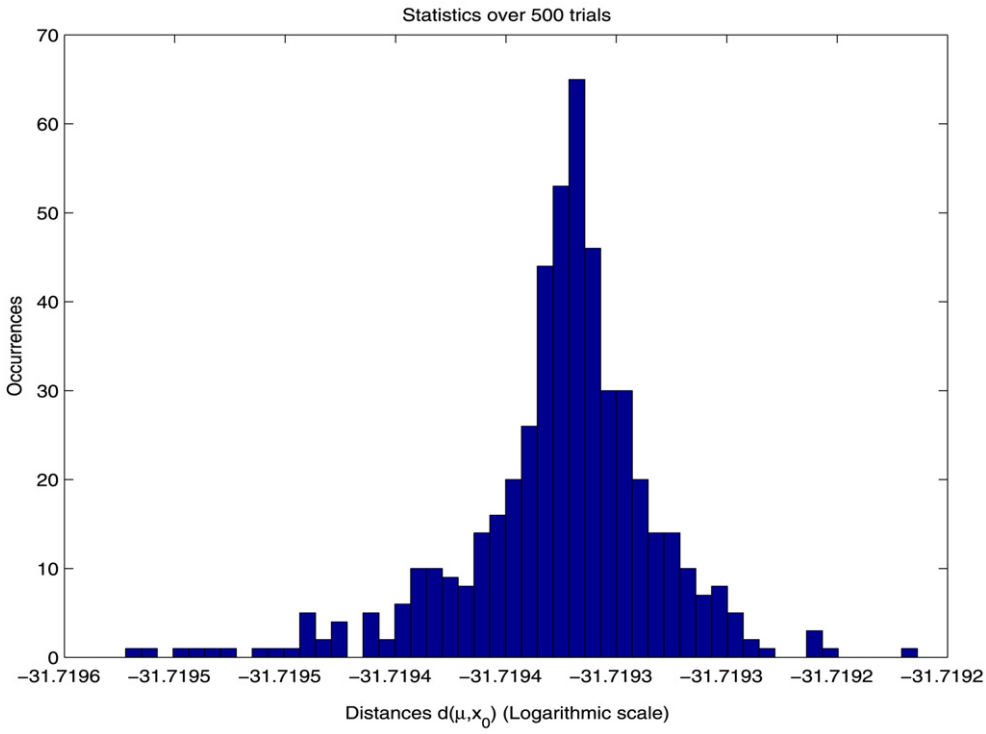

Fig. 4. Example with $p = 10$ and $N = 49$. Distribution of values $d_{\text{geo}}(\mu, x_0)$ over independent trials obtained by running the averaging algorithm from randomly generated initial states.

3.1.3. Empirical statistical analysis of random selection of initial guess

The averaging algorithm (13) needs an initial point x_{i} to start iteration, which is usually referred to as 'initial guess.'

In order to understand empirically the sensitivity of the algorithm with respect to the choice of an initial guess, we performed a empirical statistical analysis by randomly generating starting points. In particular, on the same problem tackled in Section 3.1.2, the averaging algorithm was run over 500 independent trials and was run from different initial guess-vectors having all random values with normal distribution (duly projected over the unit hypersphere) over each trial.

The result of such statistical analysis may be observed in Fig. 4. In particular, the figure shows the distribution of the values $d_{\text{geo}}(\mu, x_0)$ at the end of each trial. The spread of results is negligible in practice, so that we may conclude that every initial vector in the unit hypersphere is fine in practice, including randomly generated ones.

3.2. Experiments with real data

The second set of experiments refers to real unitary vectors to average. In particular, the following experiments refer to applications related to independent component analysis and blind channel deconvolution. Details about independent-component-analysis and blind-deconvolution algorithms are not reported here because it would be out of scope and because *the averaging algorithm (13) was developed in a way that is independent of the nature of the problem/algorithm that it is applied to.*

3.2.1. Experiments on one-unit independent component analysis

Independent component analysis is a signal/data processing technique that allows one to extract independent factors from their mixtures. One-unit independent component analysis (uICA) allows one to extract only one independent factor from a mixture, according to some optimality condition. A widely known uICA technique bases upon kurtosis maximization [14]. One-unit independent component analysis is related to general independent component analysis by the concept of deflation: Once a component has been extracted by a mixture, it may be 'subtracted' from the mixture so that iterating uICA will result in another independent component at any time [14]. Indeed, in some applications, a mixture contains a useful signal mixed with noises or its warped replicas, so that one-unit independent component analysis is enough to get rid of disturbances and clear out the useful signal. This is the case, for instance, in 'synthetic aperture radar' (SAR) imagery analysis [9] or in non-destructive material testing/evaluation (NDT/NDE) [12].

In the present example, we consider an algorithm presented in [12] based on artificial neural network learning and use it with a single weight vector on kurtosis optimization. The numerical experiment is repeated from [12] with gray-level images as source signals and randomly generated mixtures. In summary, uICA by kurtosis optimization is based on a single neuron that learns a weight-vector $x \in \mathbb{R}^p$ that optimizes a criterion function $F(x)$ based on statistical kurtosis. This ensures

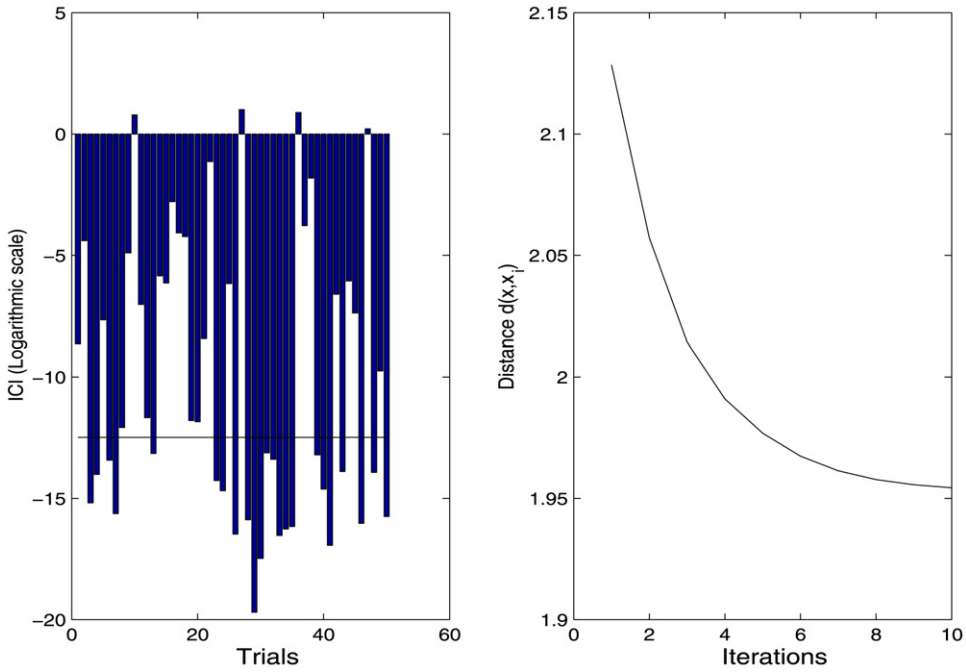

Fig. 5. Example on one-unit independent component analysis. Left panel: Values of inter-channel interference ICI(x_k) over 50 trials and ICI value pertaining to the average vector, ICI(μ). Right panel: Distance $d_{\mathrm{geo}}(x, x_{\mathrm{i}})$, with x_{i} being a randomly chosen initial point, during iteration of algorithm (13).

that the neuron outputs a maximally (or, upon sign switch, minimally) kurtotic signal. The learning criterion function is homogeneous, i.e., it satisfies condition $F(x) = F(\alpha x)$ for every $\alpha \in \mathbb{R} - \{0\}$. This means that the norm of the weight-vector x does not matter, therefore, optimization is performed under the constraint that $x \in \mathbb{S}^{p-1}$. Such mathematical property also reflects the known fact that independent components of a mixture in uICA may be recovered up to sign and amplitude.

In order to test the averaging algorithm, we may proceed by the following guidelines:

- Run the uICA algorithm a number of times ($N = 50$ times, here) and collect the learnt neuron's weight-vector $x_k \in \mathbb{S}^{p-1}$ over each trial. Evaluate the quality of each collected weight-vector by means of an 'inter-channel interference index' (ICI) [14]. The closer the ICI value to 0, the better the quality of a uICA solution. The quality of each weight-vector solution will be denoted by ICI(x_k).
- Run the averaging algorithm and compute the average vector μ. Evaluate the quality of the average by computing its ICI index, namely, ICI(μ).
- Compare the quality of the average vector with the quality of the set of neuron's weight-vectors.

In this example, as opposite to toy cases of study, the true average vector is not known a priori, therefore the quality of average is measured through an index related to the application at hand.

Fig. 5 shows a comparison of the quality of the average vector with the quality of the set of weight vector learnt by the uICA algorithm. Fig. 5 also shows the course of the value $d_{\mathrm{geo}}(x, x_{\mathrm{i}})$ during iteration, with x_{i} being a randomly chosen initial point for the averaging algorithm. Its absolute value does not matter, it is used to confirm that the algorithm steadily converges toward an average solution.

Results show that the averaging algorithm is able to compute a weight-vector solution whose quality may be deemed good, as it takes on the best solutions (but, of course, it gets spoiled by low-quality solutions as well).

3.2.2. Experiments on blind channel deconvolution

Blind channel deconvolution (or equalization) is a signal processing technique used, e.g., in data communication and storage, that allows to restore data warped by the medium that it propagates/stores within. The basic hypothesis is that the data get warped in a convolutional way (i.e., by delays, weights and sum) so that inverse operation, termed de-convolution, may be performed by proper delay, weighting and sum operators [7]. Typical applications of blind deconvolution (BD) are communication channel equalization [7], mass data storage/retrieval performance enhancement [4] and stratified materials analysis [20].

In blind deconvolution, a filter parameter/weight-vector x is sought for that allows an adaptive system to recover a signal from its convolutionally-warped version. It is known that a source signal in BD can be recovered up to sign and amplitude (and time lag), therefore the norm of the weight vector x does not matter. Moreover, for BD-filter parameters optimization

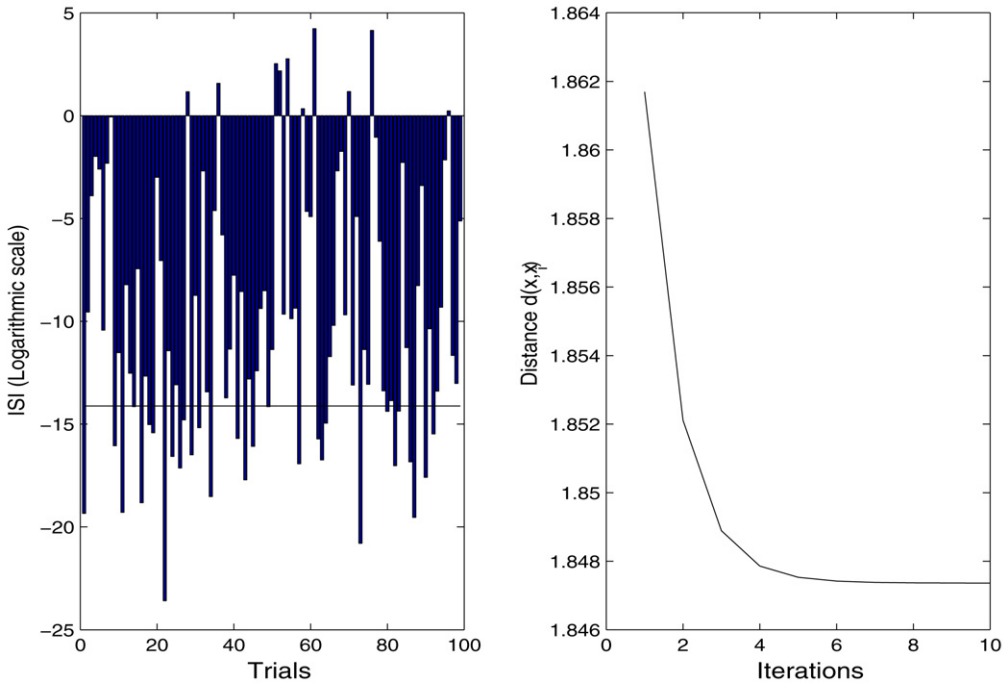

Fig. 6. Example on blind channel deconvolution. Left panel: Values of inter-symbol interference (ISI) over 99 trials and ISI value pertaining to the average vector. Right panel: Distance $d_{\mathrm{geo}}(x, x_{\mathrm{i}})$ with x_{i} being a randomly chosen initial point, during iteration of algorithm (13).

purpose, imposing a constraint on the norm of the weight-vector is in order, which is termed 'automatic gain control' [10]. Such constraint may be safely assumed to be $x \in \mathbb{S}^{p-1}$.

The numerical experiment presented here was repeated from [10], about the deconvolution of a real-world (non-minimum phase) telephonic channel. (In particular, the 'super-exponential' algorithm was made use of.) The adaptive filter has $p = 14$ taps. In order to test the averaging algorithm, we proceeded as follows:

- Run a BD algorithm a number of times ($N = 99$ times, in this case) and collect the adapted parameter-vector $x_k \in \mathbb{S}^{p-1}$ over each trial. Evaluate the quality of each collected parameter-vector by means of an 'inter-symbol interference index' (ISI) [10]. The closer the ISI value to 0, the better the quality of a BD solution. The quality of each parameter-vector solution will be denoted by $\mathrm{ISI}(x_k)$.
- Run the averaging algorithm and compute the average vector x. Evaluate the quality of the average parameter-vector by computing its ISI index value, namely, $\mathrm{ISI}(\mu)$.
- Compare the quality of the average parameter-vector with the quality of the set of parameter-vectors.

Even in this example, the exact average parameter-vector is not known a priori, therefore the quality of average was measured through an index related to the application.

Fig. 6 shows a comparison of the quality of the average parameter-vector with the quality of the set of weight-vector adapted by the BD algorithm. Fig. 6 also shows the course of the value $d_{\mathrm{geo}}(x, x_{\mathrm{i}})$ during iteration, with x_{i} being a randomly chosen initial point for the averaging algorithm.

Again, results of application of the averaging algorithm proposed here show that the algorithm steadily converges toward an average solution and is able to compute a fairly good parameter-vector.

4. Conclusion

We developed a unit-norm vector averaging technique based on the differential geometrical structure of the unit hypersphere. The envisaged averaging algorithm may be employed to merge several parameters-patterns learnt by an adaptive signal processing system in order to obtain a better representative parameter-pattern.

The envisaged algorithm (13) is easy to implement and was found to converge steadily and in a few iterations over tested problems.

The behavior of the developed averaging algorithm was illustrated via numerical experiments, on both toy and application-oriented data. Results of numerical experiments showed that the computed mean-value is much closer to the samples center than most of the available samples.

References

[1] A. Banerjee, S. Merugu, I. Dhillon, J. Ghosh, Clustering with Bregman divergences, J. Mach. Learn. Res. 6 (2005) 1705–1749.
[2] J. Bernoulli, Ars conjectandi: Usum & applicationem praecedentis, in: Doctrinae Civilibus, Moralibus & Oeconomicis, 1713, Chapter 4.
[3] L.M. Bregman, The relaxation method of finding the common point of convex sets and its applications to the solution of problems in convex programming, USSR Comput. Math. Math. Phys. 7 (3) (1967) 200–217.
[4] S. Choi, S. Ong, J. Cho, C. You, D. Hong, Performances of neural equalizers on partial erasure model, IEEE Trans. Magn. 33 (5) (1997) 2788–2790.
[5] Y.-L. Chou, Statistical Analysis: With Business and Economic Applications, Holt, Rinehart and Winston Inc., New York, 1969.
[6] N. Del Buono, L. Lopez, Runge–Kutta type methods based on geodesics for systems of ODEs on the Stiefel manifold, BIT—Numer. Math. 41 (5) (2001) 912–923.
[7] Z. Ding, Y. Li, Blind Equalization and Identification, Marcel Dekker, New York, 2001.
[8] S. Fiori, Neural minor component analysis approach to robust constrained beamforming, IEE Proc. Vision Image Signal Process. 150 (4) (2003) 205–218.
[9] S. Fiori, Overview of independent component analysis technique with an application to synthetic aperture radar (SAR) imagery processing, Neural Netw. (Special Issue on "Neural Networks for Analysis of Complex Scientific Data: Astronomy, Geology and Geophysics") 16 (3–4) (2003) 453–467.
[10] S. Fiori, A fast fixed-point neural blind deconvolution algorithm, IEEE Trans. Neural Netw. 15 (2) (2004) 455–459.
[11] S. Fiori, Geodesic-based and projection-based neural blind deconvolution algorithms, Signal Process. 88 (3) (2008) 521–538.
[12] S. Fiori, P. Burrascano, One-unit 'rigid-bodies' learning rule for principal/independent component analysis with application to ECT-NDE signal processing, Neurocomputing 56 (2004) 233–255.
[13] M. Fréchet, Les élements aléatoires de nature quelconque dans un espace distancié, Ann. Inst. H. Poincaré 10 (1948) 215–310.
[14] A. Hyvärinen, Fast and robust fixed-point algorithms for independent component analysis, IEEE Trans. Neural Netw. 10 (3) (1999) 626–634.
[15] R. Koenker, The median is the message: Toward the Fréchet median, J. Soc. Fr. Stat. 147 (2) (2006) 61–64.
[16] F. Nielsen, J.-D. Boissonnat, R. Nock, Bregman Voronoi diagrams: Properties, algorithms and applications, Institut National de Recherche en Informatique et en Automatique (INRIA Sophia Antipolis), Research Report No. 6154, 2007.
[17] P. Pajunen, M. Girolami, Implementing decisions in binary decision trees using independent component analysis, in: Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation, Helsinki, Finland, June 19–22, 2000, pp. 477–481.
[18] O. Shalvi, E. Weinstein, Super-exponential methods for blind deconvolution, IEEE Trans. Inform. Theory 39 (1993) 504–519.
[19] R.K. Sundaram, Existence of solutions: The Weierstrass Theorem, in: A First Course in Optimization Theory, Cambridge University Press, 1996, Chapter 3.
[20] R.A. Wiggins, Minimum entropy deconvolution, Geoexploration 16 (1978) 21–35.

Simone Fiori was born in Rimini (Italy) in June 1971. He received the Italian Laurea (Dr.Eng.) cum laude in electronics engineering in July 1996 from the University of Ancona (Italy), and the Ph.D. degree in electrical engineering (circuit theory) in March 2000 from the University of Bologna (Italy). His research interests include unsupervised learning theory for artificial neural networks, linear and nonlinear adaptive discrete-time filter theory, vision and image processing by neural networks, continuous-time and discrete-time circuits for stochastic information processing. He is author of more than 130 refereed journal and conference papers on these topics. Dr. Fiori was the recipient of the 2001 "E.R. Caianiello Award" for the best Ph.D. dissertation in the artificial neural network field. He is currently serving as Associate Editor of *Neurocomputing* journal, the *Computational Intelligence and Neuroscience* journal and of *Cognitive Computation* journal; also, he is serving as member of the board of *International Journal of Computational Intelligence Studies*.