LETTER **Communicated by Shun-ichi Amari**

Parameter Learning for Alpha Integration

Heeyoul Choi
heeyoul@gmail.com
Samsung Advanced Institute of Technology, Samsung Electronics, Yongin, Gyeonggi 446-712, Republic of Korea

Seungjin Choi
seungjin@postech.ac.kr
Department of Computer Science and Engineering, IT Convergence Engineering, and Department of Creative IT Excellence Engineering, Pohang University of Science and Technology, Pohang 790-784, Korea

Yoonsuck Choe
choe@cs.tamu.edu
Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, U.S.A.

In pattern recognition, data integration is an important issue, and when properly done, it can lead to improved performance. Also, data integration can be used to help model and understand multimodal processing in the brain. Amari proposed α-integration as a principled way of blending multiple positive measures (e.g., stochastic models in the form of probability distributions), enabling an optimal integration in the sense of minimizing the α-divergence. It also encompasses existing integration methods as its special case, for example, a weighted average and an exponential mixture. The parameter α determines integration characteristics, and the weight vector w assigns the degree of importance to each measure. In most work, however, α and w are given in advance rather than learned. In this letter, we present a parameter learning algorithm for learning α and w from data when multiple integrated target values are available. Numerical experiments on synthetic as well as real-world data demonstrate the effectiveness of the proposed method.

1 Introduction

When we make an educated guess in a recognition task, we use sensory data from multiple modalities (visual, audio, taste, smell, and touch) and integrate them. Several hypotheses have been proposed to account for the multimodal integration mechanism in the brain at many different levels (Wolpert & Kawato, 1998; Amari, 2007). Also, in pattern recognition,

data integration has become an important issue since aggregating information from multiple sources helps with disambiguation and generally leads to higher performance. Several data integration algorithms have been proposed to address the issues (Hall & Llinas, 1997), such as Bayesian inference (Pearl, 1988), evidence theory (Dempster, 1967; Shafer, 1976), clustering algorithms (Duda, Hart, & Stork, 2001), and neural networks (Bishop, 1995). Kernel-based integration methods are also commonly used (Lanckriet, Deng, Cristianini, Jordan, & Noble, 2004; Choi, Choi, & Choe, 2008).

Recently, a general framework, α-integration, was proposed by Amari (2007) for stochastic model integration of multiple positive measures. α-integration is a one-parameter family of integration, where the parameter α determines the characteristics of integration. Given a number of stochastic models in the form of probability distributions, it finds out the optimal integration of the sources in the sense of minimizing the α-divergence. Many artificial neural models for stochastic models, such as the mixture (or product) of experts model (Jacobs, Jordan, Nowlan, & Hinton, 1991; Hinton, 2002), can be considered as special cases of α-integration. Some psychophysical laws such as Weber's law and Steven's law support that our brain could use something like α-representation when proper α values are used (see Kandel, Schwarts, & Jessell, 2000, and Amari, 2007).

However, there is an unresolved critical issue in α-integration. In most existing works on α-integration (Minka, 2005; Amari, 2007; Cichocki, Lee, Kim, & Choi, 2008; Kim, Cichocki, & Choi, 2008; Choi, Katake, Choi, & Choe, 2010), the value of α as well as the weight vector $\boldsymbol{w}$ are given in advance rather than learned. However, these existing α-integration approaches cannot effectively handle cases where only a number of integrated (mixed) results are available while the α value and $\boldsymbol{w}$ are unknown, as in Figure 1. When we have a very sparse sampling of integrated observations plus the original measurements, we should be able to deduce the α and the weight so that we can obtain the full integration result. So, the α value or $\boldsymbol{w}$ needs to be learned adaptively from a small number of integrated observations and the corresponding measurements (the raw data). The full integration result can then be estimated using the deduced α and $\boldsymbol{w}$.

We can also choose a specific α value if we have an underlying stochastic model or assumption. For example, we can set α to 1 so that the resulting integration is a geometric mean. In this case, the underlying stochastic model is the exponential family, a specific case of α-family. However, if the α value is fixed like that, the benefit of generalizing to an arbitrary stochastic model is lost. So we should automatically infer the α value, which amounts to a search through a space of stochastic models, which will lead to a more accurate integration that optimizes on α-divergence.

In this letter, we propose a new algorithm to learn α-integration parameters α and $\boldsymbol{w}$ from the data sources and a small number of integrated target values. We first define an objective function with respect to α and $\boldsymbol{w}$ and then derive two update rules to learn the parameters based on gradient

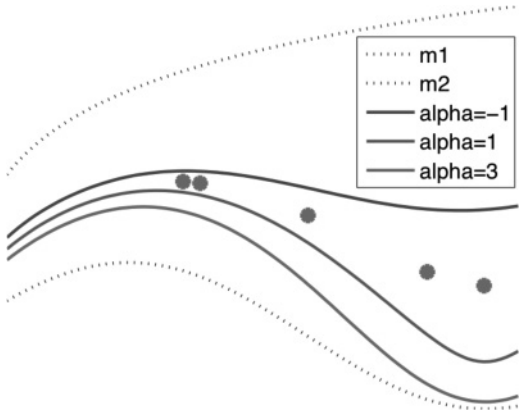


Figure 1: Given three α values and a weight vector [0.6 0.4], three α-integration curves of two data measures (m_1, m_2) can be calculated, respectively. In many real-world situations, only a sparse sampling of integrated results may be available (dots).

descent. The procedure consists of two steps: (1) an α-integration step and (2) a parameter update step (our main contribution). These steps are executed iteratively.

The rest of this letter is organized as follows. First, we briefly review α-integration and α-divergence in section 2. Then in section 3, we propose a new parameter learning algorithm. Section 4 includes experimental results and analysis with toy and real-world data sets. In section 5, we discuss the proposed algorithm in terms of stochastic models and geometry, followed by the conclusion in section 6.

2 Previous Work

In this section, we provide a brief overview of α-mean, α-integration, and α-divergence. More details can be found in Amari (2007).

Let us consider two positive measures of random variable x, denoted by $m_1(x) > 0$ and $m_2(x) > 0$, satisfying

$$\int m_i(x)\, dx > 0,$$

for $i = 1, 2$. The simplest integration of these two positive measures is to compute the arithmetic mean, $\widetilde{m}_{\mathrm{a}}(x) = \frac{1}{2}\{m_1(x) + m_2(x)\}$, or the geometric mean, $\widetilde{m}_{\mathrm{g}}(x) = \sqrt{m_1(x)m_2(x)}$.

2.1 α-Mean. *α-mean* is a one-parameter family of means defined by

$$\widetilde{m}_\alpha(x) = f_\alpha^{-1}\left(\frac{1}{2}\{f_\alpha(m_1(x)) + f_\alpha(m_2(x))\}\right), \tag{2.1}$$

where $f_\alpha(\cdot)$ is a differentiable monotone function given by

$$f_\alpha(z) = \begin{cases} z^{\frac{1-\alpha}{2}}, & \alpha \neq 1, \\ \log z, & \alpha = 1. \end{cases} \tag{2.2}$$

The function $f_\alpha(\cdot)$ in equation 2.2 is the only function that makes the α-mean to be linear scale free for $c > 0$, that is, the α-mean of $c\,m_1(x)$ and $c\,m_2(x)$ is $c\,\widetilde{m}_\alpha(x)$ (Hardy, Littlewood, & Polya, 1994; Amari & Nagaoka, 2000).

α-mean includes various means as its special case. For $\alpha = -1, 1, 3, \infty$ or $-\infty$, α-mean becomes arithmetic mean, geometric mean, harmonic mean, minimum, or maximum, respectively. Figure 1 shows an example of α-mean with two source measures. The value of the parameter α (which is usually specified in advance) reflects the characteristics of the integration. As α increases, the α-mean resorts more to the smaller of $m_1(x)$ or $m_2(x)$, while as α decreases, the larger of the two is weighed more heavily, as shown in Figure 2 (Amari, 2007).

2.2 α-Integration. *α-integration* is a generalization of α-mean to multiple positive sources, $m_1(x), \ldots, m_M(x)$ with different weights, w_i (Amari, 2007). The α-integration equation of $m_i(x)$, $i = 1, \ldots, M$, with w_i is defined by

$$\widetilde{m}(x) = f_\alpha^{-1}\left(\sum_{i=1}^{M} w_i f_\alpha(m_i(x))\right), \tag{2.3}$$

where $w_i > 0$ for $i = 1, \ldots, M$ and $\sum_{i=1}^{M} w_i = 1$.

Given M positive measures, the goal of integration is to seek their weighted average $\widetilde{m}(x)$ that is as close to each of the measures as possible, while the closeness of two positive measures is evaluated by divergence. Amari (2007) showed that α-integration $\widetilde{m}(x)$ is optimal in the sense that

$$\mathcal{J}_\alpha[\widetilde{m}(x)] = \sum_{i=1}^{M} w_i D_\alpha[m_i(x) \parallel \widetilde{m}(x)] \tag{2.4}$$

is minimized, where $D_\alpha[m_i(x) \parallel \widetilde{m}(x)]$ is the α-divergence of $\widetilde{m}(x)$ from the measures $m_i(x)$.

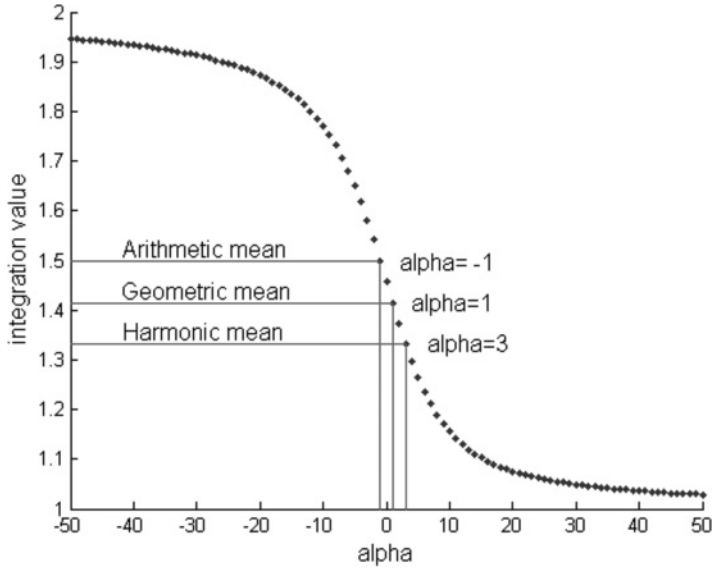


Figure 2: α-integration includes various types of means as its special case depending on the α value. Here, the values of two measures are 1 and 2, respectively. Note that as α increases or decreases, the α-integration value decreases or increases monotonically and converges with 1 or 2, respectively.

2.3 α-Divergence. α-divergence belongs to a family of convex divergence measures known as Csiszár's f-divergence or Ali-Silvey divergence (Ali & Silvey, 1966; Csiszár, 1974). The α-divergence $D_\alpha[m_1\|m_2]$ is derived from f-divergence $D_f[m_1\|m_2]$, making use of $f(\cdot)$ given by

$$f(z)=\begin{cases}\dfrac{4}{1-\alpha^2}\left\{\dfrac{1-\alpha}{2}+\dfrac{1+\alpha}{2}z-z^{\frac{1+\alpha}{2}}\right\}, & \alpha\neq\pm1,\\ z-1-\log z, & \alpha=-1,\\ z-1+z\log z, & \alpha=1.\end{cases}\tag{2.5}$$

That is, for $\alpha\neq\pm1$,

$$D_\alpha[m_1\|m_2]=\frac{4}{1-\alpha^2}\left\{\int\frac{1-\alpha}{2}m_1(x)+\frac{1+\alpha}{2}m_2(x)-m_1(x)^{\frac{1-\alpha}{2}}m_2(x)^{\frac{1+\alpha}{2}}dx\right\}.$$

For $\alpha = -1$, $D_\alpha[m_1 \| m_2] = D_{\mathrm{KL}}[m_1 \| m_2]$, and for $\alpha = 1$, $D_\alpha[m_1 \| m_2] = D_{\mathrm{KL}}[m_2 \| m_1]$, where $D_{\mathrm{KL}}[m_i \| m_j]$ is the Kullback-Leibler divergence between m_i and m_j.

For applications where α-integration is applied to real problems, see Kim et al. (2008), and Choi, Choi, Katake, Kang, and Choe (2010), and Choi, Katake et al. (2010), where α and $\boldsymbol{w}$ are given and fixed (i.e., specified by the user). However, these values are unknown a priori unless we have a specific data set and understand it. In the next section, we propose a new objective function and update rules for learning α and $\boldsymbol{w}$.

3 Learning the Parameters for α-Integration

The problem that we consider in this paper is as follows. Given M positive measurements, $m_i(x)$, our task is to determine an α-integration $\widetilde{m}(x)$ when target values for $\widetilde{m}(x)$ are sparsely observed. In other words, we learn the parameters α and $\boldsymbol{w}$ such that the optimal α-integration $\widetilde{m}(x)$ is as close as possible to the observed target values.

Optimal α-integration has the form

$$\widetilde{m}(x) = \begin{cases} \left\{ \sum_i w_i m_i(x)^{\frac{1-\alpha}{2}} \right\}^{\frac{2}{1-\alpha}}, & \alpha \neq 1, \\ \exp\left\{ \sum_i w_i \log m_i(x) \right\}, & \alpha = 1, \end{cases} \tag{3.1}$$

which is derived by applying the calculus of variation to solve

$$\frac{\partial \mathcal{J}_\alpha[\widetilde{m}(x)]}{\partial \widetilde{m}(x)} = 0 \tag{3.2}$$

for $\widetilde{m}(x)$, where $\mathcal{J}_\alpha[\widetilde{m}(x)]$ is defined in equation 2.4 (Amari, 2007).

Let $m_i(x_k)$ be ith measurement for x_k, where $i = 1, \ldots, M$ and $k = 1, \ldots, N$, with the number of targets, S_N, ($S_N \ll N$). Given true target values t_j (the integrated values) where $j = 1, \cdots, S_N$, our objective function to be minimized for α and $\boldsymbol{w}$, $\mathcal{J}(\alpha, \boldsymbol{w})$, is simply defined as

$$\mathcal{J}(\alpha, \boldsymbol{w}) = \frac{1}{S_N} \sum_{j=1}^{S_N} (t_j - \widetilde{m}(x_j))^2. \tag{3.3}$$

One example of the error surface using the objective function is shown in Figure 3. The error surface was generated with 30 randomly selected target values from the synthetic data set in the experiment section. Here, we can

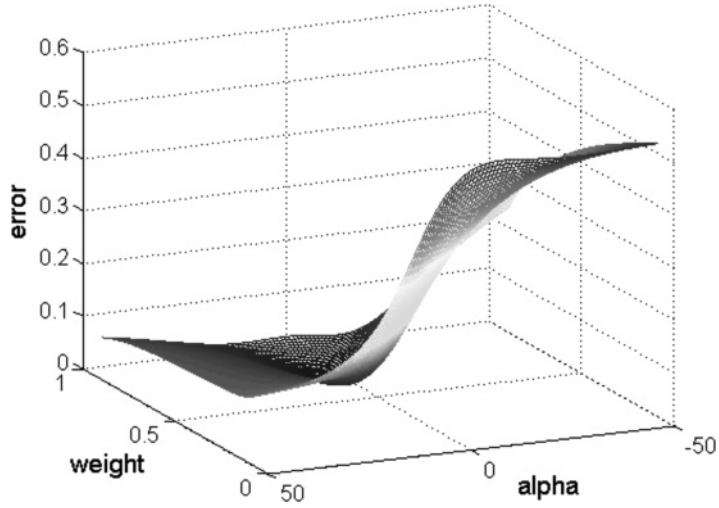


Figure 3: An example of the error surface from equation 3.3, with 30 randomly selected target values (integrated result) of the synthetic data set in Figure 5. Here only one weight for the first measurement, w_1, is shown against α.

see that the optimal parameters, $\alpha = 3$, $\boldsymbol{w} = [0.7, 0.3]$, are found at the global minimum of the error surface.

Two update rules for α and $\boldsymbol{w}$ are described in sections 3.1 and 3.2, respectively. Section 3.3 shows how to learn both parameters simultaneously from the given data set and discusses some related issues.

3.1 Update Rule for α. To learn α, we take a derivative of equation 3.3 with respect to α,

$$\frac{\partial \mathcal{J}(\alpha, \boldsymbol{w})}{\partial \alpha} = -\frac{2}{S_N} \sum_j (t_j - \widetilde{m}(x_j)) \frac{\partial \widetilde{m}(x_j)}{\partial \alpha}, \tag{3.4}$$

where

$$\frac{\partial \widetilde{m}(x)}{\partial \alpha} = \frac{2\widetilde{m}(x)}{1-\alpha} \left\{ \frac{\log\left(\sum_i w_i f_\alpha(m_i(x))\right)}{1-\alpha} + \frac{\sum_i w_i \frac{\partial f_\alpha(m_i(x))}{\partial \alpha}}{\sum_i w_i f_\alpha(m_i(x))} \right\}, \tag{3.5}$$

and

$$\frac{\partial f_\alpha(u)}{\partial \alpha} = -\frac{1}{2} \log(u) u^{\frac{1-\alpha}{2}}. \tag{3.6}$$

Finally, we use gradient descent to update α as

$$\Delta\alpha = -\eta_\alpha \frac{\partial \mathcal{J}(\alpha, \boldsymbol{w})}{\partial \alpha}, \tag{3.7}$$

where η_α is the learning rate for α. By applying equation 3.7 repeatedly until it converges to a stable point, we can find the optimal value for α.

3.2 Update Rule for $\boldsymbol{w}$. In order to learn $\boldsymbol{w}$, we take a derivative of equation 3.3 with respect to $\boldsymbol{w}$. As in the update rule for α, each element of the gradient vector $\frac{\partial \mathcal{J}(\alpha, \boldsymbol{w})}{\partial \boldsymbol{w}}$ is obtained by

$$\frac{\partial \mathcal{J}(\alpha, \boldsymbol{w})}{\partial w_i} = -\frac{2}{S_N} \sum_j (t_j - \widetilde{m}(x_j)) \frac{\partial \widetilde{m}(x_j)}{\partial w_i}, \tag{3.8}$$

where

$$\frac{\partial \widetilde{m}(x)}{\partial w_i} = \begin{cases} \dfrac{2}{1-\alpha} \left(\dfrac{\widetilde{m}(x) f_\alpha((m_i(x))}{\sum_k w_k f_\alpha(m_k(x))} \right), & \alpha \neq 1 \\ \widetilde{m}(x) \log m_i(x), & \alpha = 1 \end{cases}.$$

Then, as before, the update rule is

$$\Delta \boldsymbol{w} = -\eta_w \frac{\partial \mathcal{J}(\alpha, \boldsymbol{w})}{\partial \boldsymbol{w}}, \tag{3.9}$$

where η_w is the learning rate for $\boldsymbol{w}$. After each iteration, the weight vector should be renormalized so that its sum is equal to 1. As above, we can apply equation 3.9 until it converges.

Moreover, for $\boldsymbol{w}$, when the α value is around 1, we can have a simple algebraic solution that is approximately the same as the solution from equation 3.9. If α is exactly 1, the unique algebraic solution is exactly the same as the solution from equation 3.9. To obtain the one-shot solution, we start from rewriting the objective function in equation 3.3 to look like

$$\mathcal{J}_2(\alpha, \boldsymbol{w}) = \frac{1}{S_N} \sum_{j=1}^{S_N} (f_\alpha(t_j) - \sum_i w_i f_\alpha(m_i(x_j))^2. \tag{3.10}$$

Note that this objective function does not have $\widetilde{m}(x)$ in the equation in contrast to $\mathcal{J}$, so $\boldsymbol{w}$ can be obtained without any iterative interaction with the update rule for $\widetilde{m}(x)$. Actually, $\mathcal{J}_2$ is a variation of $\mathcal{J}$ where f was taken

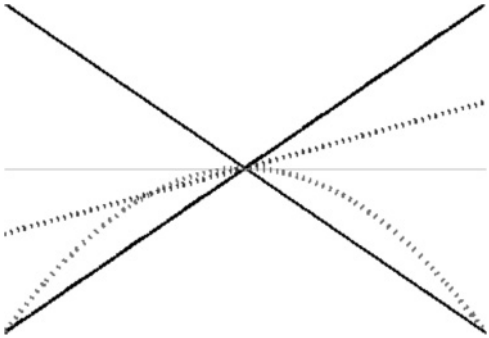


Figure 4: Different role of α and $\boldsymbol{w}$ in integrating two sources. The two diagonal lines are the two data sources. The horizontal line is an integration of the two sources with $\alpha = -1$ and $\boldsymbol{w} = [0.5, 0.5]$. The dotted line is an integration with $\alpha = -1$ and $\boldsymbol{w} = [0.7, 0.3]$, which is a weighted sum of the two sources. The dotted curve is another integration, with $\alpha = 3$ and $\boldsymbol{w} = [0.5, 0.5]$.

off, where f has nothing to do with $\boldsymbol{w}$ (only when $\alpha = 1$). Then we can use the least-square method for optimization. In this letter, however, we use equation 3.9 as the update rule for $\boldsymbol{w}$ to keep the method working with arbitrary α values including $\alpha = 1$.

3.3 Learning α and $\boldsymbol{w}$ at the Same Time. In this section, we show how we can learn both α and $\boldsymbol{w}$ at the same time and discuss some related issues. As in Figure 4, α and $\boldsymbol{w}$ usually play different roles during integration. For example, in the figure, the two diagonal lines are sources, and the horizontal line is an integration of the two sources with $\{\alpha, \boldsymbol{w}\} = \{-1, [0.5, 0.5]\}$, which is the arithmetic mean. The dotted line and the dotted curve are integrations with $\{\alpha, \boldsymbol{w}\} = \{-1, [0.7, 0.3]\}$, and $\{\alpha, \boldsymbol{w}\} = \{3, [0.5, 0.5]\}$, respectively. These integrated lines and the curve are generated by changing only one of the two parameters compared to the case of arithmetic mean to show the different roles they play. Obviously, the dotted line is a linearly weighted sum of the two sources, which does not care about the magnitude of the sources. However, the dotted curve follows the magnitude, not the sources. Since the two parameters play different roles during integration, we can estimate the parameters by applying the two update rules simultaneously starting from a random initial point. Equations 3.7 and 3.9 can be applied sequentially in each iteration of the learning procedure.

However, α and $\boldsymbol{w}$ could serve the same role in some special cases like the point of intersection of the dotted line and curve in Figure 4. Here, with the same sources and the same integrated value, we can have two different sets of parameters. This kind of case could happen easily, as can be seen in

the integration equation:

$$\widetilde{m}(x) = \left\{ \sum_i w_i m_i(x)^{\frac{1-\alpha}{2}} \right\}^{\frac{2}{1-\alpha}}. \tag{3.11}$$

For example, given two source points $m_1(x_1)$ and $m_2(x_1)$ and the integration point $\widetilde{m}(x_1)$, we have only one equation with two parameters: α and $\boldsymbol{w}$. While either of the two parameters with the other one fixed can solve the equation, we may not be able to optimize the two parameters at the same time. So in order to avoid such cases, we need enough labeled data points to learn both parameters. That is, if we have more data points, we can have more stable solutions for both parameters even in noisy cases.

Another issue is to determine the appropriate learning rates: η_α and η_w. Suppose there are two positive numbers and we want to find the maximum of them. Then there are two ways to get the maximum: (1) take $\alpha = -\infty$ and (2) $\boldsymbol{w} = [1, 0]$ if the first one is bigger, or $\boldsymbol{w} = [0, 1]$ otherwise. That is, to obtain the maximum, both $\alpha = -\infty$, and $\boldsymbol{w} = [0, 1]$ (or $[1, 0]$) could be the solution. And to find the minimum, both $\alpha = \infty$ and $\boldsymbol{w} = [1, 0]$ (or $[0, 1]$) could be the solution. Here, to get the same effect, the range of α is $[-\infty, \infty]$, while for $\boldsymbol{w}$, it is $[0, 1]$. So this different scale should be reflected on the learning rates. That is, the learning rate for α, η_α should be much greater than the one for $\boldsymbol{w}$, η_w. We will see some examples in section 4.

4 Experiments

In order to show the effectiveness of our proposed algorithm, we carried out experiments with three data sets: (1) a synthetic data set with two curves and a few true integrated values as in Figure 5; (2) monthly average temperatures of multiple cities from www.cityrating.com, as in Table 1; and (3) a color image in Figure 12, which is converted to gray images in different ways.

4.1 Synthetic Data. In this section, we learn the parameters from a synthetic data set where we know the true parameter values (but hidden to the algorithm) so that we can measure the accuracy of the learning algorithms. We show the results of learning the parameters α and $\boldsymbol{w}$ simultaneously. Based on repeated experiments with randomly generated target values, we confirmed that there is no significant difference on the rate of convergence when different parameters were used.

Figure 5 shows two data sources—the true integrated curves and the target values. We generated 30 labeled data points. In addition, the learning rates for α and $\boldsymbol{w}$ should have different scales, which are 0.5 and 0.00005, respectively. The true values for α and $\boldsymbol{w}$ are 3 and $[0.7, 0.3]$, both of which are hidden to the learning algorithm.

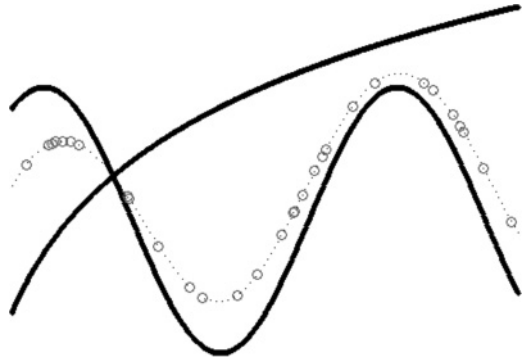
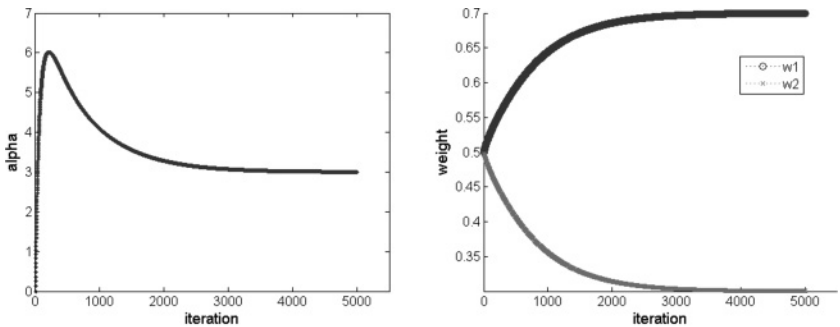


Figure 5: Example data to show how to learn both parameters. The black sine curve has 0.7 weight, and the log curve has 0.3. The dotted curve is the integrated curve with $\alpha = 3$. The circles are target values that are labeled and observed by the learning algorithm.



(a) α (b) $\boldsymbol{w}$

Figure 6: Learning α and $\boldsymbol{w}$ starting from 0 and [0.5, 0.5]. They converge to the true values, 3 and [0.7, 0.3], respectively.

Figure 6 shows the evolution of the parameter values over time. The parameters start from 0 and [0.5, 0.5]. As the learning algorithm updates the parameters, they converge to the true values, 3 and [0.7, 0.3], respectively. Note that α increases up to 6 and then goes back to converge to 3. This is because of the interaction between α and $\boldsymbol{w}$, which may be sharing the same role under certain situations. However, the error decreases monotonically and converges to zero (see Figure 7).

To check the interaction between α and $\boldsymbol{w}$, we decreased the number of labeled data points. In Figure 8, the error surface is obtained from five randomly selected target values. In Figure 8b, the tangent space at the bottom of the surface seems to meet the surface on a straight line that includes

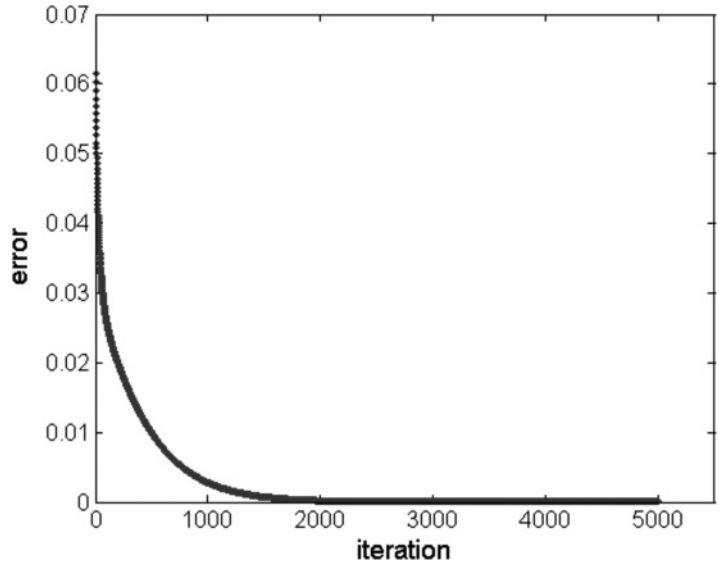


Figure 7: Error decreases as iteration proceeds. Finally, the error converges to almost zero.

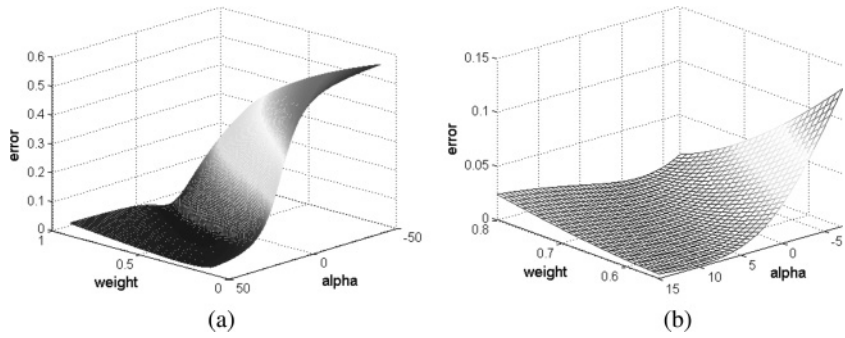


Figure 8: An example of the error surface with five randomly selected target values of the synthetic data set. (a) A big picture of the surface and (b) a zoomed-in surface around the true parameter: $\alpha = 3$ and $\boldsymbol{w} = [0.7, 0.3]$.

the error value at the true parameters. It means that the learning procedure could stop at any parameter set corresponding to the line. This line is a function of α and $\boldsymbol{w}$, indicating the interaction between them. If we increase the number of labeled data points, this line becomes curved so that the algorithm could learn the true parameters. Compare this error surface to the one in Figure 3, where 30 randomly selected target values are given. Note

Table 1: Monthly Average Temperature data (F°).

City	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.
Chicago	21.0	25.4	37.2	48.6	58.9	68.6	73.2	71.7	64.4	52.8	40.0	26.6
Boston	28.6	30.3	38.6	48.1	58.2	67.7	73.5	71.9	64.8	54.8	45.3	33.6
New York	31.5	33.6	42.4	52.5	62.7	71.6	76.8	75.5	68.2	57.5	47.6	36.6
Atlanta	41.0	44.8	53.5	61.5	69.2	76.0	78.8	78.1	72.7	62.3	53.1	44.5
Houston	50.4	53.9	60.6	68.3	74.5	80.4	82.6	82.3	78.2	69.6	61.0	53.5
San Antonio	49.3	53.5	61.7	69.3	75.5	82.2	85.0	84.9	79.3	70.2	60.4	52.2

Source: www.cityrating.com.

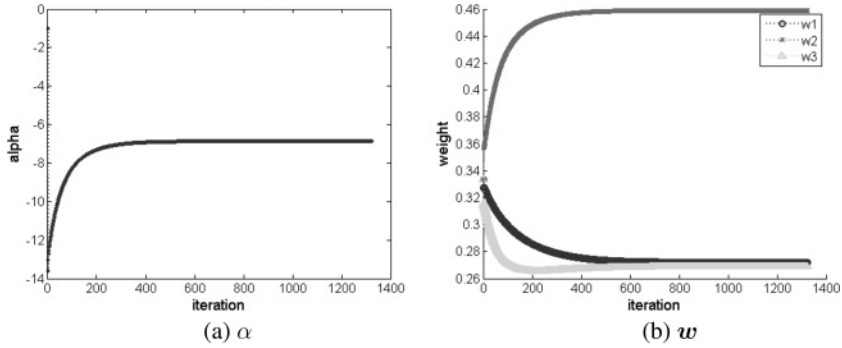
that these surfaces were generated from randomly selected target values so that sometimes it could learn the parameter almost perfectly, even with just five target values.

4.2 Temperature Data. As a second data set, we used a monthly average temperature data from several cities in the United States (see Table 1). We used three cities (New York, New York; Chicago, Illinois; and Houston, Texas) as sources and estimated the temperature of Atlanta, Georgia. We used temperatures from 10 randomly selected months in Atlanta to learn the parameters and tested with the 2 remaining months' temperature. The temperature scale is Fahrenheit (F°).

The parameters α and $\boldsymbol{w}$ were initialized to 0 and $[1/3, 1/3, 1/3]$, respectively. We expected α to decrease so much that the integration approaches the higher value, and also expected Houston to have more weight than others, since Atlanta's temperature is more similar to Houston's than the other cities. In the experiments, as the learning procedure proceeded, the parameters converged to -6.86 and $[0.27, 0.46, 0.27]$, as shown in Figure 9. In this example, the learned values of the parameters take on a real-world meaning, based on temperature-based geometry. This was confirmed on similar experiments with other cities as shown in Table 1.

The error decreases to 0.604 as shown in Figure 10. Figure 11 shows the true temperature of Atlanta, along with the estimated temperature using the optimal parameters. There are many causes that affect the temperature of a city. Each month of each city might have various unique factors that affect its temperature, which can be the source of error we cannot overcome simply by averaging, even with accurately learned α and $\boldsymbol{w}$ values. However, our proposed method is better than the simple arithmetic (or geometric or harmonic) average. It theoretically achieves the minimum error among all linear-scale-free averaging methods.

4.3 Color-to-Grayscale Image Conversion. Our method can be applied to approximately discover a proper color-to-grayscale image conversion strategy, given a color image and an example grayscale version of the image.



(a) α (b) $\boldsymbol{w}$

Figure 9: Learning α and $\boldsymbol{w}$. Time evolution of (a) α converging to -6.862 and (b) $\boldsymbol{w}$ converging to [0.272, 0.459, 0.269] for New York, Houston, and Chicago.

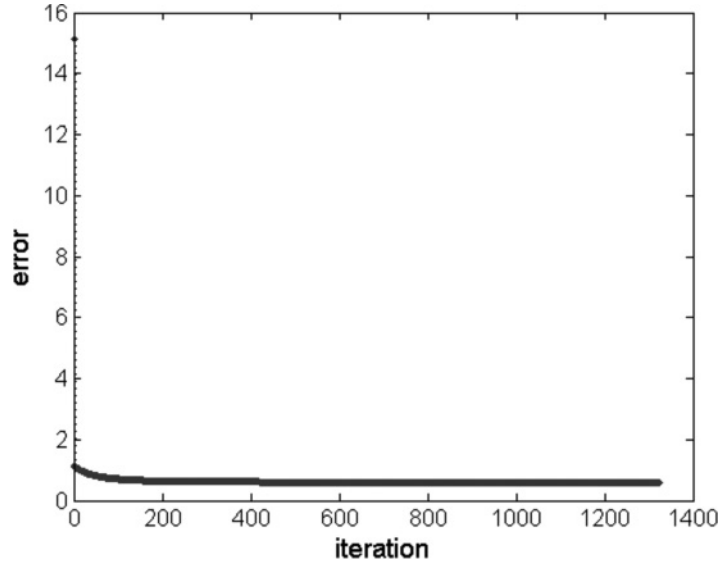


Figure 10: Error curve when both parameters are updated, converging to 0.604.

An $m \times n$ pixel-sized color image consists of three different $m \times n$ sources: red, green, and blue (RGB). To convert the color image to grayscale, simple strategies like the luminosity method find a weighted sum of the RGB sources. Among many complex strategies, we took five popular ones: Rasche05, Grundland07, Smith08, Color2gray, and CIE_Y. (For details of these strategies, see Cadík, 2008.)

A 198×200 pixel color image used in the experiment is shown in Figure 12, and the five converted grayscale images are shown in Figure 13a.

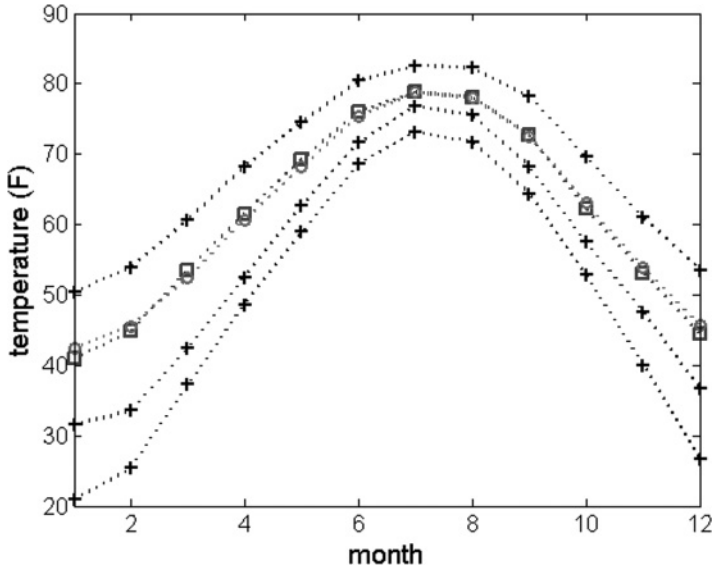
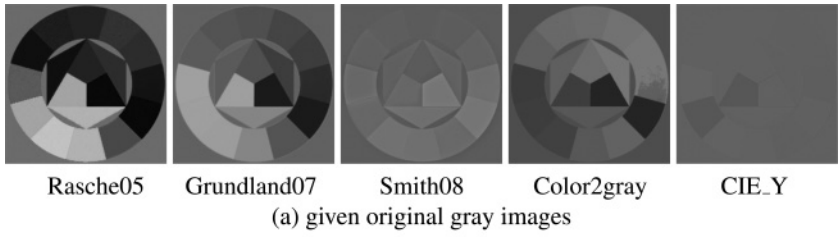


Figure 11: The results of learning both α and $\boldsymbol{w}$. The lines with crosses are for are New York, Chicago, and Houston. The lines with squares are the true Atlanta temperature. The lines with circles are the estimated temperature.

Figure 12: A color image. A color image can be converted to grayscale using different methods. Adapted from http://cadik.posvete.cz/color_to_gray_evaluation/



(a) given original gray images

(b) α-integrated gray images with the estimated parameters

Figure 13: Original grayscale images and the α-integrated images. From left to right, the conversion strategies are Rasche05, Grundland07, Smith08, Color2gray, and CIE_Y.

Table 2: Parameters for Conversion Strategies.

Algorithm	α	$\boldsymbol{w}$
Rasche05	−0.1973	[0.2722 , 0.2758, 0.4520]
Grundland07	−3.5940	[0.0188, 0.5766, 0.4046]
Smith08	−7.9890	[0.0966, 0.8617, 0.0417]
Color2gray	−0.5416	[0.3490, 0.6510, 0.0000]
CIE_Y	−3.6599	[0.1970, 0.7347, 0.0683]

Given the color image and the target gray image, we learned the parameters with 500 randomly selected pixel samples from the pair of images. Then we applied α-integration to the color image to convert it into grayscale with the learned parameters. The target grayscale images are not exactly α-integration of RGB, which means our learning would be somewhat of an approximation of the original conversion strategies. Figure 13b shows the five α-integrated gray images, which are qualitatively similar to the original grayscale images in Figure 13a.

The learned parameters are summarized in Table 2. As we found real-world meanings of the learned parameter values in the previous section, we can understand some characteristics of the conversion strategies based on the parameters learned for α-integration. For example, the learned α value for the Smith08 strategy is much lower compared to other strategies, which could mean that the grayscale result of the Smith08 strategy may be brighter than other strategies. Also, from the weight vector, we can tell that

the blue region in the color image would be mapped to a brighter gray in the first two strategies (Rasche05 and Grundland07) relative to those of the other strategies (w_3 is 0.4520 and 0.4046 for the first two; and 0.0417, 0.0000 and 0.0683 for the rest). These interpretations are confirmed in Figure 13. Also note that we can find infinitely many new strategies by changing the parameters.

An earlier version of our work reported here has also been used in a real-world application. Based on our previous conference paper (Choi, Choi, Katake, & Choe, 2010), Wang et al. (2011) have successfully applied our method to colonic polyp detection in CT colonography, where they integrated decisions from the CAD system (machine intelligence) and the MTurk workers (human intelligence) based on learning only α.

5 Discussion

In this paper, we propose a learning algorithm for estimating α-integration parameters α and $\boldsymbol{w}$ from the data; previous work required manually determined, fixed values for those parameters. In this section, we discuss the proposed algorithms in terms of stochastic models and geometry.

In terms of stochastic models, the α-family includes stochastic models like the exponential family and the mixture family, so learning α can be seen as finding the best family out of all the stochastic families in the α-family. In that sense, our proposed algorithm can be seen as trying to find the best stochastic family model and the best distribution to fit the model. That is, when we learn the parameters, they find a better model (a set of distributions) given the current integration samples and the data sources. As a consequence, α-integration with learning the value of α and $\boldsymbol{w}$ identifies, arguably, the best average out of all possible distributions in the α-family. Since we have not theoretically proved the objective function to be convex, there is the possibility that the learning finds local optimal parameters.

From a geometrical point of view, defining the distance between any two points in a data set leads to a single corresponding metric, which can be used to determine the manifold on which the data points lie. Thus, learning the parameters corresponds to defining the manifold of the probability distributions (or nonnegative measurements). When we initialize the parameters, we assume one manifold, and when the parameters are updated, the shape of the manifold we assumed will change. The α-integration and the manifold shape are updated iteratively. In the end, α-integration with learning the value of α and $\boldsymbol{w}$ gives us the best integration with a metric for the manifold. These two interpretations are strongly related since α-integration originated from information geometry (Amari & Nagaoka, 2000).

Another advantage of our approach is that the learned parameter values can take on a concrete real-world meaning depending on the data set. In

the temperature experiments, the parameters have some temperature-based geometrical meaning, and in the color-to-grayscale conversion strategies, the parameter values can tell the relative brightness and bias in color composition. Likewise, with an arbitrary data set, we can try to interpret the optimized parameter values after learning is complete.

In addition to some engineering data sets, our approach can be used in neuroscientific models based on specific parameter values. When part of the brain gets multiple inputs and generates one output by any kind of mixture, we can apply α-integration with learning and estimate the functional role of the input sources based on the optimized parameter values.

6 Conclusion

In this paper, we proposed a new method for learning α-integration parameters from sparse integration samples (target values): the characteristic value α and the weight vector $\boldsymbol{w}$, for α-integration to optimize data integration. The update rules were rigorously derived, and the performance was validated in experiments with several synthetic and real-world data sets. Given only a few target values, our method found the best parameters to achieve the best integration in terms of α-divergence. The estimated parameter values have domain-specific meaning (semantics); thus the resulting parameters can be used to infer the functional nature of the mixture.

We expect our approach to help automate the α-integration framework. Furthermore, α-integration parameter optimization may be possible in different domains, such as reinforcement learning and unsupervised learning.

Acknowledgments

This work is partly based on our previous conference presentation (Choi, Choi, Katake, & Choe, 2010). This publication is based in part on work supported by Award KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST). S.C. was supported by National Research Foundation (NRF) of Korea (2012-0005785 and 2012-0005786), the Converging Research Center Program funded by the Ministry of Education, Science, and Technology (2012K001343), Korea MKE and NIPA IT Consilience Creative Program (C1515-1121-0003), and NRF World Class University Program (R31-10100).

References

Ali, S. M., & Silvey, S. D. (1966). A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society B*, *28*, 131–142.

Amari, S. (2007). Integration of stochastic models by minimizing α-divergence. *Neural Computation*, *19*, 2780–2796.

Amari, S., & Nagaoka, H. (2000). *Methods of information geometry*. Providence, RI: American Mathematical Society, and New York: Oxford University Press.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. New York: Oxford University Press.

Cadík, M. (2008). Perceptual evaluation of color-to-grayscale image conversions. *Comput. Graph. Forum*, *27*, 1745–1754.

Choi, H., Choi, S., & Choe, Y. (2008). Manifold integration with Markov random walks. In *Proc. Association for the Advancement of Artificial Intelligent (AAAI)* (Vol. 1, pp. 424–429). Palo Alto, CA: AAAI.

Choi, H., Choi, S., Katake, A., & Choe, Y. (2010). Learning alpha-integration with partially-labeled data. In *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing* (pp. 2058–2061). Piscataway, NJ: IEEE.

Choi, H., Choi, S., Katake, A., Kang, Y., & Choe, Y. (2010). Manifold alpha-integration. In *Proc. Pacific Rim Int. Conf. on Artificial Intelligence (PRICAI)*, Korea. LNCS 2010, Vol. 6230 (pp. 397–408). New York: Springer-Verlag.

Choi, H., Katake, A., Choi, S., & Choe, Y. (2010). Alpha-integration of multiple evidence. In *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing* (pp. 2210–2213). Piscataway, NJ: IEEE.

Cichocki, A., Lee, H., Kim, Y.-D., & Choi, S. (2008). Nonnegative matrix factorization with α-divergence. *Pattern Recognition Letters*, *29*(9), 1433–1440.

Csiszár, I. (1974). Information measures: A critical survey. In *Trans. 7th Prague Conference on Information Theory* (Vol. A, pp. 73–86). Berlin: Springer-Verlag.

Dempster, A. P. (1967). Upper and lower probabilities induced by a multivalued mapping. *Annals of Statistics*, *28*, 325–339.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. New York: Wiley.

Hall, D. L., & Llinas, J. (1997). An introduction to multisensor data fusion. *Proceedings of the IEEE*, *85*(1), 369–376.

Hardy, G. H., Littlewood, J. E., & Polya, G. (1994). *Inequalities* (2nd ed.). Cambridge: Cambridge University Press.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, *14*, 1771–1800.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, *3*, 79–81.

Kandel, E. R., Schwarts, J. H., & Jessell, T. M. (2000). *Principles of neural science* (4th ed.). New York: McGraw-Hill.

Kim, Y. D., Cichocki, A., & Choi, S. (2008). Nonnegative Tucker decomposition with alpha-divergence. In *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing* (pp. 1829–1832). Piscataway, NJ: IEEE.

Lanckriet, G. R. G., Deng, M., Cristianini, N., Jordan, M. I., & Noble, W. S. (2004). Kernel-based data fusion and its application to protein function prediction in yeast. In *Proc. Pacific Symposium on Biocomputing (PSB)* (Vol. 9, pp. 300–311). Singapore: World Scientific.

Minka, T. (2005). *Divergence measures and message passing* (Tech. Rep. MSR-TR-2005-173). Redmond, WA: Microsoft Research.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference* (2nd ed.) San Francisco: Morgan Kaufmann.

Shafer, G. (1976). *A mathematical theory of evidence*. Princeton, NJ: Princeton University Press.

Wang, S., Anugu, V., Nguyen, T., Rose, N., Burns, J., McKenna, M., Petrick, N., & Summers, R. M. (2011). Fusion of machine intelligence and human intelligence for colonic polyp detection in CT colonography. In *Proceedings of the International Symposium on Biomedical Imaging: From Nano to Macro* (pp. 160–164). Piscataway, NJ: IEEE.

Wolpert, D., & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, *11*, 1317–1329.

Received September 18, 2012; accepted December 21, 2012.