LEARNING α-INTEGRATION WITH PARTIALLY-LABELED DATA

Heeyoul Choi[1], *Seungjin Choi*[2], *Anup Katake*[3] *and Yoonsuck Choe*[1]

[1]Dept. of Computer Sci. and Eng., Texas A&M Univ., TX
{*hchoi,choe*}*@cs.tamu.edu*
[2]Dept. of Computer Sci., Pohang Univ. of Sci. and Tech., Korea
seungjin@postech.ac.kr
[3]Starvision Technologies, Inc., TX
akatake@starvisiontech.com

ABSTRACT

Sensory data integration is an important task in human brain for multimodal processing as well as in machine learning for multisensor processing. α-integration was proposed by Amari as a principled way of blending multiple positive measures (e.g., stochastic models in the form of probability distributions), providing an optimal integration in the sense of minimizing the α-divergence. It also encompasses existing integration methods as its special case, e.g., weighted average and exponential mixture. In α-integration, the value of α determines the characteristics of the integration and the weight vector $\boldsymbol{w}$ assigns the degree of importance to each measure. In most of the existing work, however, α and $\boldsymbol{w}$ are given in advance rather than learned. In this paper we present two algorithms, for learning α and $\boldsymbol{w}$ from data when only a few integrated target values are available. Numerical experiments on synthetic as well as real-world data confirm the proposed method's effectiveness.

Index Terms— α-integration, parameter estimation

1. INTRODUCTION

When we make an educated guess in recognition task, we use multiple sensory data (visual, audio, taste, smell and touch) received through the corresponding senses, and then integrate them. Several hypotheses have been proposed to account for the multimodal integration mechanism in the brain at many different levels. Also, in machine learning for pattern recognition, data integration has been an important issue to achieve improved accuracy than that based on a single source of information because one sensor might not be good enough to provide unambiguous information. Some data integration algorithms have been proposed (see [1] and references therein) such as Bayesian inference [2], evidence theory [3, 4], clustering algorithms [5] and neural networks [6]. Kernel-based integration methods are also used for integration [7, 8].

α-integration was proposed for stochastic model integration of multiple positive measures [9]. It 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 [10, 11] 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 with a proper α value for each purpose (see [12, chap 21] and [9]).

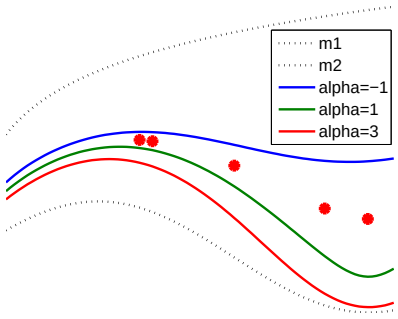

Fig. 1. Given three α values and a weight vector [0.6 0.4], three α-integration curves of two synthetic curves (m_1, m_2) are calculated, respectively. Like red dots, however, there are some cases when a few integration results are available not knowing the α value or the weight vector.

However, there is an unresolved critical issue in α-integration. In most existing works about α-integration [13, 9, 14, 15, 16], the value of α as well as the weight vector $\boldsymbol{w}$ is given in advance rather than learned. Contrary to the existing works, there are some cases when a few integration results are available but not the α value or $\boldsymbol{w}$, as in Fig. 1. Also, we can easily imagine that once a data set is given, that should be all the brain needs for integration. That is, the brain does not need to have any additional parameters specified by some external entity/process. So, the α value or $\boldsymbol{w}$ needs to be learned adaptively from a couple of integration results and corresponding measures. Then they can be applied to integrate the rest of measures.

Also, if α should be given, then even though α-integration theory generalizes some specific stochastic models into α-family, in practice, a specific model is chosen in advance. For example, if we fix α to 1 in advance, we get to use geometric mean of exponential family which is a special case of α-integration. Then there is no actual benefit in generalizing an arbitrary stochastic model. Instead, if we can find out the α value automatically, the size of the model that is considered gets actually larger than a specific stochastic model. Instead of using one model specified by α, the integration gets more accurate in terms of α-divergence since it searches over all the models.

In this paper, we propose new algorithms to learn α-integration from data when the sources and only a few integrated target values are available. There are two kinds of parameters: α and $\boldsymbol{w}$. Given

a couple of training data points, we first define an objective function with respect to α and $\boldsymbol{w}$ and then derive two algorithms to learn the parameters based on gradient descent. Basically the updating procedures consist of two parts: (1) α-integration and (2) parameter updating. These parts are executed iteratively because parameter updating equations include the α-integration equation in themselves.

2. α-INTEGRATION AND α-DIVERGENCE

Here we provide a brief overview of α-integration, more details on which can be found in [9]. Let us consider two positive measures of random variable x, denoted by $m_1(x) > 0$ and $m_2(x) > 0$ for $i = 1, 2$. *α-mean* [9] is a one-parameter family of means, defined by

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

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

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

α-mean includes various commonly used 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. The value of the parameter α (which is usually specified in advance and fixed) reflects the characteristics of the integration. As α increases, α-mean resorts more to the smaller one between $m_1(x)$ and $m_2(x)$ (more optimistic), while as α decreases, the larger one is considered with more weight [9].

α-mean can be generalized to the weighted α-mixture of M positive measures $m_1(x)$, ..., $m_M(x)$ with weights $\boldsymbol{w} = [w_1, w_2, \ldots, w_M]$, which is referred to as *α-integration* of $m_1(x)$, ..., $m_M(x)$ with weights $\boldsymbol{w}$ [9].

Definition 1 (α-integration) *The α-integration of $m_i(x)$, $i = 1, \ldots, M$, with weights w_i is defined by*

$$\widetilde{m}(x) = f_\alpha^{-1}\left(\sum_{i=1}^{M} w_i f_\alpha(m_i(x))\right), \tag{3}$$

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

Given M positive measures, $m_i(x)$, $i = 1, \ldots, M$, 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 how close two positive measures are is evaluated using a divergence measure. It was shown by Amari that α-integration $\widetilde{m}(x)$ is optimal in the sense that the risk function

$$\mathcal{J}_\alpha[\widetilde{m}(x)] = \sum_{i=1}^{M} w_i D_\alpha[m_i(x) \,\|\, \widetilde{m}(x)] \tag{4}$$

is minimized, where $D_\alpha[m_i(x) \,\|\, \widetilde{m}(x)]$ is the *α-divergence* of $\widetilde{m}(x)$ from the measures $m_i(x)$ [9].

3. LEARNING THE PARAMETERS FOR α-INTEGRATION

The problem that we consider in this paper is as follows. Given M positive measurements, $m_i(x)$, where $i = 1, \cdots, M$ and $m_i(x) > 0$ for all x, our task is to determine an α-integration $\widetilde{m}(x)$ when target values for $\widetilde{m}(x)$ are partially available. We assume that either α or $\boldsymbol{w}$ is known in advance. In other words, given w_i's (or fixed in advance), we learn the parameter α such that the optimal α-integration $\widetilde{m}(x)$ is as close to partially available target values as possible. If α is given, then we learn parameters w_i's under the same criterion.

Optimal α-integration has the form

$$\widetilde{m}(x) = \begin{cases} \left\{\sum_i w_i m_i(x)^{\frac{1-\alpha}{2}}\right\}^{\frac{2}{1-\alpha}}, & \alpha \neq 1, \\ \exp\left\{\sum_i w_i \log m_i(x)\right\}, & \alpha = 1, \end{cases} \tag{5}$$

which is derived by applying the calculus of variation to solve $\frac{\partial \mathcal{J}_\alpha[\widetilde{m}(x)]}{\partial \widetilde{m}(x)} = 0$ for $\widetilde{m}(x)$, where $\mathcal{J}_\alpha[\widetilde{m}(x)]$ is of the form of Eq. (4) [9].

Sec. 3.1 describes how to learn the parameter α in optimal α-integration in Eq. (5) when target values for $\widetilde{m}(x)$ are partially available, given w_i's. Sec. 3.2 explains how to determine parameters w_i's in optimal α-integration in Eq. (5) when α is already learned or given in advance.

3.1. Learning α

Given M measures $m_i(x_k)$ where $i = 1, ..., M$ and $k = 1, ..., N$, let S_N be the number of targets ($S_N \ll N$). With true target values t_j (the integrated values) where $j = 1, \cdots, S_N$, our objective function for α, $\mathcal{J}(\alpha)$ is defined by

$$\mathcal{J}(\alpha) = \sum_{j=1}^{S_N} (t_j - \widetilde{m}(x_j))^2, \tag{6}$$

which is expected to be minimized. Then, we take a derivative of Eq. (6), which follows as below.

$$\frac{\partial \mathcal{J}(\alpha)}{\partial \alpha} = -2 \sum_j (t_j - \widetilde{m}(x_j)) \frac{\partial \widetilde{m}(x_j)}{\partial \alpha}, \tag{7}$$

where

$$\begin{aligned} \frac{\partial \widetilde{m}(x)}{\partial \alpha} &= \frac{2\widetilde{m}(x)}{1-\alpha}\left\{\frac{\log(\sum_i w_i f_\alpha(m_i(x)))}{1-\alpha} \right. \\ &\qquad \left. + \frac{\sum_i w_i \frac{\partial f_\alpha(m_i(x))}{\partial \alpha}}{\sum_i w_i f_\alpha(m_i(x))}\right\}, \\ \frac{\partial f_\alpha(u)}{\partial \alpha} &= -\frac{1}{2}\log(u) u^{\frac{1-\alpha}{2}}. \end{aligned}$$

Finally, we use gradient descent to update α given by

$$\Delta\alpha = -\eta_\alpha \frac{\partial \mathcal{J}(\alpha)}{\partial \alpha}, \tag{8}$$

where η_α is the learning rate for α. Note that since $\widetilde{m}(x)$ is a monotonically decreasing function with respect to α, Eq. (6) is a convex function and Eq. (8) converges to the global optimizer always as we can see in some examples in Fig. 2.

3.2. Learning $\boldsymbol{w}$

In order to learn $\boldsymbol{w}$, we can use Eq. (6) as an objective function for $\boldsymbol{w}$ to get a gradient algorithm based on the derivative of the objective function with respect to $\boldsymbol{w}$. Each element of the gradient vector $\frac{\partial \mathcal{J}(\boldsymbol{w})}{\partial \boldsymbol{w}}$ is obtained by

$$\frac{\partial \mathcal{J}(\boldsymbol{w})}{\partial w_i} = -2 \sum_j (t_j - \widetilde{m}(x_j)) \frac{\partial \widetilde{m}(x_j)}{\partial w_i}, \tag{9}$$

where

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

Then, the updating rule is given by

$$\Delta \boldsymbol{w} = -\eta_w \frac{\partial(\mathcal{J}\boldsymbol{w})}{\partial \boldsymbol{w}}, \tag{10}$$

where η_w is the learning rate for $\boldsymbol{w}$.

4. EXPERIMENTS

In order to show the effectiveness of our proposed algorithms, we carried out experiments with two different data sets: (a) synthetic data set with two curves and a few true integrated values and (b) monthly average temperatures of multiple cities from www.cityrating.com.

4.1. Synthetic data set

Given 2 curves and only 5 target values from the true integration, we tried to find an optimal α or $\boldsymbol{w}$ for the target values when one of the parameters is given. We used one weight vector, [0.4 0.6] with different true α values, 3 and -2 (unknown to the algorithm). The α learning procedure is based on two measurements and 5 points randomly selected from the true curve with the fixed weights as in Fig. 1. The learning trajectory is shown in Fig. 2. The sum of the squared errors between the estimated curve with the learned α value and the true curve are 9.85e-11 and 9.91e-11 for $\alpha = 3$ and $\alpha = -2$, respectively. With different weight vectors like [0.9 0.1], our algorithm worked perfectly too (data not shown).

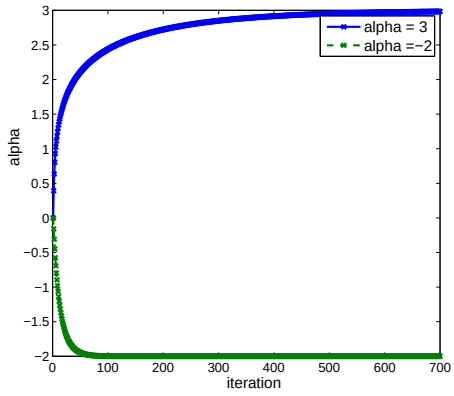

Fig. 2. The trajectories of α values for the α learning with two synthetic curves in the two cases: 3 and -2. Both curves starts from the initial value 0 and are converging to 3 and -2 at around 700 iterations, respectively. The converging speed depends on the learning rate, which is 0.05 here.

Given α, our algorithm found out $\boldsymbol{w}$ perfectly with any given hidden true weights. The squared errors in the estimated curve and in the estimated weight vector are almost 0 as shown in Table 1. In this data set, we did not assume any noise to the true values.

Table 1. Squared errors in learning $\boldsymbol{w}$ ($\times 10^{-29}$)

α	-2	-1	0	1	2	3
$\mathcal{J}$	0.725	0.875	3.116	0.017	1.505	0.932
$\boldsymbol{w}$	0.019	0.024	0.135	0.001	0.228	0.217

4.2. City temperature data set

To test our proposed method in a real world task, we used a monthly average temperature data from several cities in the U.S. First, we used three cities (New York, Chicago, and Houston) as sources and estimated the temperature of Atlanta. Second, two cities (San Antonio and Boston) were used as sources and New York as the target. We assumed that all the temperature information are correct, so we used the same weights for all source cities. For both cases, we used temperatures from 10 randomly selected months to learn α and tested with the other 2 months' temperature.

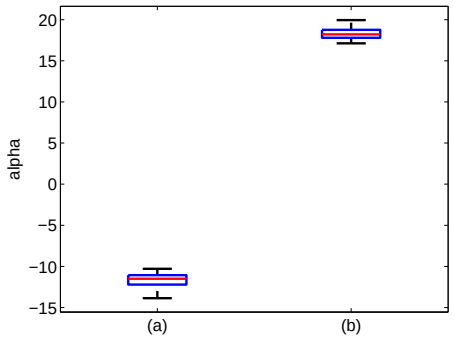

Fig. 3. Boxplot of α values for city temperatures from 50 random experiments. (a) α values for estimating Atlanta's temperature from the three other cities: Chicago, New York and Houston (b) α values for estimating New York's temperature from the two other cities: Boston and San Antonio.

For the first case, Atlanta is roughly equidistant to three cities. Here we have two cooler cities (New York and Chicago) and one warmer city (Houston) than Atlanta. So we expect α to be low to move the estimation to get closer to the higher value (Houston's temperature) because we have only one warmer city. Fig. 3a shows boxplot of α values for 50 experiments and the average α value is -11.62.

In the second case, New York is much closer to Boston than San Antonio. So we expect the α value to be high. Fig. 3b shows boxplot of α values for 50 experiments and the average is 18.37. With this kind of an example, the α value takes on a concrete meaning, based on temperature-based geometry. According to this α value, we can guess how close the city is to some other city in terms of the temperature. A high α value means that the city is close to the cooler city, and a low α value means the city is close to the warmer city.

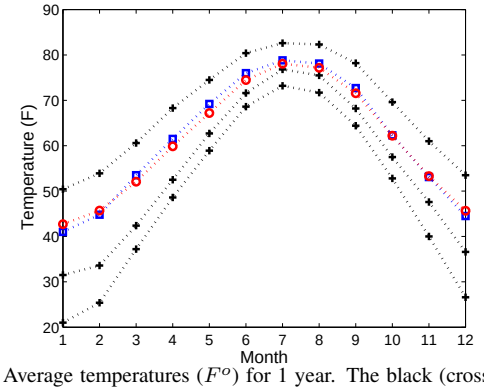

Fig. 4. Average temperatures (F^o) for 1 year. The black (cross) lines are New York, Chicago and Houston. The blue (square) line is the true Atlanta temperature. The red (circle) line is the estimated temperature.

Fig. 4 shows the true temperature of Atlanta and estimated tem-

perature with $\alpha = -11.62$ learned above. Fig. 5 shows a boxplot of errors in the estimated temperature of New York with $\alpha = 18.37$ learned above or $\alpha = -1$ fixed in advance. There are many causes to affect the temperature of a city. Each month of each city might have different causes to the temperature. That can be the reason of the error we cannot overcome simply by averaging even with a very carefully learned α value. 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.

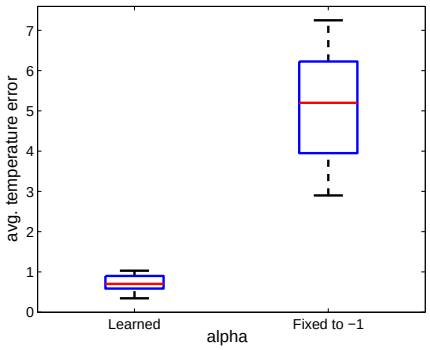

Fig. 5. Errors for New York's temperature (F^o) from the two other cities: Boston and San Antonio. The boxplots are from 50 random experiments. In the 'alpha' axis, 'Learned' means our estimated α and 'Fixed to -1' means the simple arithmetic average.

5. DISCUSSION

The main contribution of this paper is to have proposed a learning algorithm for estimating α-integration parameters α and $\boldsymbol{w}$ from the data, whereas previous works required manually determined, fixed values for those parameters.

In terms of stochastic models, α-family includes some stochastic models like exponential family and mixture family. So learning α can be seen as finding the best family out of all the stochastic families in α-family. In that sense, our proposed algorithm tries to find the best stochastic family model and the best distribution in the model, iteratively. That is, when we learn α, it finds a better model (a set of distributions) than the current model for the current integration and sources. Then α-integration gives us the best distribution out of the current model. As a consequence, α-integration with learning the value of α finds out the best average out of all distributions in α-family.

From a geometrical view point, if we define the distance between any two points in the set, it implies one corresponding metric, which defines the manifold where the points lie on. Here, learning α means defining the manifold of the probability distributions (or nonnegative measurements). When we initialize α, we assume one manifold and when α changes, the shape of the manifold we assume changes. So, α-integration with learning the value of α gives us the best integration with the metric of the manifold which is defined by the optimized α value. The α-integration and the manifold shape are determined iteratively. These two interpretations are very similar because α-integration originated from information geometry [17].

In addition, as we discussed in the previous section, α can take on a concrete meaning depending on the data set. In the temperature experiments, α has some temperature-based geometrical meaning. Likewise, with a specific data set, we can try to interpret the optimized α value after learning.

6. CONCLUSION

We proposed a new method to learn parameters, α and the weight vector $\boldsymbol{w}$, for α-integration to optimize data integration. The updating rules were rigorously derived and the performance was checked in experiments with two data sets. Given only few target values, our method found the best parameters to achieve the best integration. We expect our approach to help further automate the α-integration framework.

7. ACKNOWLEDGMENTS

Portion of this work was supported by NRF Converging Research Center Program (No. 2009-0093714), Korea Research Foundation Grant (KRF-2008-313-D00939), NIPA Software Engineering Technologies Development and Experts Education Program, and WCU Program (Project No. R31-2008-000-10100-0). Heeyoul Choi was supported by StarVision Technologies' student sponsorship program.

8. REFERENCES

[1] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 369–376, 1997.

[2] J. Pearl, *Proabilistic Inference in Intelligent Systems*, Morgan Kaufmann, 1999.

[3] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," *The annals of Statistics*, vol. 28, pp. 325–339, 1967.

[4] G. Shafer, *A Mathematical Theory of Evidence*, Princeton, NJ, Princeton University Press, 1976.

[5] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiely & Sons, 2001.

[6] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[7] G. R. G. Lanckriet, M. Deng, N. Cristianini, M. I. Jordan, and W. S. Noble, "Kernel-based data fusion and its application to protein function prediction in yeast," in *Proc. Pacific Symposium on Biocomputing (PSB)*, Big Island, HI, 2004.

[8] H. Choi, S. Choi, and Y. Choe, "Manifold integration with Markov random walks," in *Proc. Association for the Advancement of Artificial Intelligent (AAAI)*, Chicago, IL, 2008.

[9] S. Amari, "Integration of stochastic models by minimizing α-divergence," *Neural Computation*, vol. 19, pp. 2780–2796, 2007.

[10] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, pp. 79–81, 1991.

[11] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, pp. 1771–1800, 2002.

[12] E. R. Kandel, J. H. Schwarts, and T. M. Jessell, *Principles of Neural Science*, McGraw-Hill, fourth edition, 2000.

[13] T. Minka, "Divergence measures and message passing," Tech. Rep. MSR-TR-2005-173, Microsoft Research, 2005.

[14] A. Cichocki, H. Lee, Y.-D. Kim, and S. Choi, "Nonnegative matrix factorization with α-divergence," vol. 29, no. 9, pp. 1433–1440, Jul. 2008.

[15] Y. -D. Kim, A. Cichocki, and S. Choi, "Nonnegative Tucker decomposition with alpha-divergence," in *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, Las Vegas, Nevada, 2008.

[16] H. Choi, A. Katake, S. Choi, and Y. Choe, "Alpha-integration of multiple evidence," in *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, Dallas, Texas, 2010.

[17] S. Amari and H. Nagaoka, *Methods of Information Geometry*, American Mathematical Society and Oxford University Press, RI and New York, 2000.