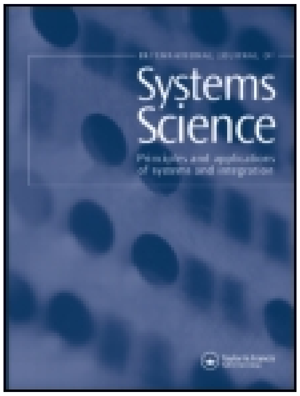
This article was downloaded by: [National Chiao Tung University 國立交通大學]
On: 25 April 2014, At: 19:13
Publisher: Taylor & Francis
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



International Journal of Systems Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tsys20

Weighted bootstrap for neural model selection

Shun-Chin Chuang [a], Wen-Liang Hung [b] & Hsin-Chia Fu [a]

[a] Department of Computer Science and Information Engineering, National Chiao-Tung University, Hsin-Chu, Taiwan

[b] Graduate Institute of Computer Science, National Hsinchu University of Education, Hsin-Chu, Taiwan

Published online: 17 Mar 2008.

To cite this article: Shun-Chin Chuang, Wen-Liang Hung & Hsin-Chia Fu (2008) Weighted bootstrap for neural model selection, International Journal of Systems Science, 39:5, 557-562, DOI: 10.1080/00207720701847711

To link to this article: http://dx.doi.org/10.1080/00207720701847711

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the “Content”) contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms & Conditions of access and use can be found at http://www.tandfonline.com/page/terms-and-conditions

International Journal of Systems Science
Vol. 39, No. 5, May 2008, 557–562

Downloaded by [National Chiao Tung University] at 19:13 25 April 2014

Weighted bootstrap for neural model selection

SHUN-CHIN CHUANG†, WEN-LIANG HUNG*‡ and HSIN-CHIA FU†

†Department of Computer Science and Information Engineering, National Chiao-Tung University, Hsin-Chu, Taiwan
‡Graduate Institute of Computer Science, National Hsinchu University of Education, Hsin-Chu, Taiwan

(*Received 28 November 2006; in final form 5 December 2007*)

This article proposes a weighted bootstrap procedure, which is an efficient bootstrap technique for neural model selection. Our primary interest in reducing computer effort is to not resample (in the original bootstrap procedure) *uniformly* from the original sample, but to modify this distribution in order to obtain variance reduction. The performance of the weighted bootstrap is demonstrated on two artificial data sets and one real dataset. Experimental results show that the weighted bootstrap procedure permits an approximately 2 to 1 reduction in replication size.

Keywords: Bayesian bootstrap; Bayesian bootstrap clones; Bootstrap; Model selection; Weighted bootstrap

1. Introduction

Backpropagation multilayer perceptrons (MLPs) are by far the most commonly used neural network structures for applications in a wide range of areas, such as pattern recognition, signal processing, data compression and automatic control. A backpropagation MLP is an adaptive network whose nodes (or neurons) perform the same function on incoming signals; and this node function is usually a composite of the weight sum and a differentiable non-linear activation function. Three of the most commonly used activation functions in backpropagation MLPs are logistic, hyperbolic tangent and identity functions. For simplicity, we assume that the backpropagation MLP in this article uses the logistic function as its activation function.

Next, we consider a set of independent observations of a continuous variable y that it has to explain from a set of p explanatory variables $(x_1, x_2, \ldots, x_p)$. We use an MLP with p inputs, one hidden layer with H hidden units and one output layer to model these data as follows:

$$y = w_0 + \sum_{h=1}^{H} w_h \phi\Big(b_h + \sum_{j=1}^{p} w_{jh} x_j\Big) + \epsilon, \qquad (1)$$

where ϵ is the residual term, with zero mean, variance σ^2 (with normal distribution or not), and ϕ is the logistic function. Let $y(x;\theta)$ be the computed value for an input $x = (x_1, \ldots, x_p)$ and a parameter $\theta = (w_0, w_1, \ldots, w_H, w_{11}, \ldots, w_{pH})$. For MLPs, the choice of an appropriate model is an important problem. Recently, Kallel *et al.* (2002) applied the bootstrap method for neural model selection, since it is more effective than the leave-one-out method.

The bootstrap method was originally proposed by Efron (1979) for use in setting independent and identically distributed (i.i.d.) random variables. The typical bootstrap method can be described as follows. Consider an i.i.d. sample $\mathbf{z} = \{z_1, \ldots, z_n\}$ from a distribution function F and a statistic of interest $s(\mathbf{z})$. The ideal bootstrap estimate of the expectation of $s(\mathbf{z})$ is

$$\hat{e} = E_{\hat{F}} s(\mathbf{z}^*), \qquad (2)$$

where $\hat{F}$ is the empirical distribution function, $E_{\hat{F}}$ is the expectation under $\hat{F}$, and $\mathbf{z}^* = \{z_1^*, \ldots, z_n^*\}$ is drawn randomly from $\mathbf{z}$ with replacement. Unless $s(\mathbf{z})$ is the mean or some other simple statistic, it is not easy to compute ê exactly, so we approximate the ideal bootstrap estimate by

$$\hat{e}_B = \frac{1}{B} \sum_{b=1}^{B} s(\mathbf{z}^{*b}), \qquad (3)$$

*Corresponding author. Email: wlhung@mail.nhcue.edu.tw

International Journal of Systems Science
ISSN 0020–7721 print/ISSN 1464–5319 online © 2008 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00207720701847711

Downloaded by [National Chiao Tung University] at 19:13 25 April 2014

where each $\mathbf{z}^{*b}$ is a sample of size n drawn with a replacement from $\mathbf{z}$, B is the number of Monte Carlo simulations, and $s(\mathbf{z}^{*b})$ is the value of the statistic s evaluated at $\mathbf{z}^{*b}$.

Formula (3) is an example of a Monte Carlo estimate of the expectation $E_{\hat{F}}s(\mathbf{z}^*)$. Note that $\hat{\mathrm{e}}_B \to \hat{\mathrm{e}}$ as $B \to \infty$ according to the law of large numbers; furthermore $E(\hat{\mathrm{e}}_B) = \hat{\mathrm{e}}$ and $var(\hat{\mathrm{e}}_B - \hat{\mathrm{e}}) = c/B$ so that the error (standard deviation of $\hat{\mathrm{e}}_B - \hat{\mathrm{e}}$) goes to zero at the rate $1/\sqrt{B}$.

The remaining part of this article is organised as follows. In section 2, to make the model estimation robust against outliers, we propose a weighted bootstrap algorithm to selection model for MLPs, where the resampling probability distribution varies inversely with the absolute value of residual. Some numerical simulation results are given in section 3 and conclusions are presented in section 4.

2. Weighted bootstrap applied to selection model for MLPs

Let $\mathcal{B}_0$ be a dataset of size n, that is, $n = \mathrm{card}\{\mathcal{B}_0\}$,

$$\mathcal{B}_0 = \{(x_1; y_1), \ldots, (x_n; y_n)\},$$

where x_i is the ith value of a p-vector of explanatory variables and y_i is the response to x_i. First, we use the dataset $\mathcal{B}_0$ to estimate the parameter θ of the model (1) and the resulting least-squares estimator of θ is denoted by $\hat{\theta}$. Thus, the residual for the ith observation is denoted by e_i and is defined as follows:

$$e_i = y_i - y(x_i; \hat{\theta}).$$

Frequently, in applications, the dataset contains some cases that are extreme; that is, the observations for these cases are well separated from the remainder of the data. These extreme cases may involve large residuals and often have dramatic effects on the fitted model. It is therefore important to study the extreme cases carefully and their influence should be reduced in the fitting process. However, in uniform resampling, that is, random resampling with replacement from $\mathcal{B}_0$, each sample value is drawn with the same probability $1/n$. This resampling technique discards the influence of these extreme cases. An alternative to discarding extreme cases that is less severe is to dampen the influence of these cases. This is the purpose of our proposed weighted bootstrap approach.

Under weighted bootstrap (sampling is conducted with replacement), each data point $(x_i;y_i)$ is assigned a probability q_i of being selected on any given draw, where $\sum q_i = 1$. Taking $q_i = 1/n$ for each i, we obtain the original bootstrap method. To determine what q_i's should be used, we intuitively want to reduce the influence of extreme cases so that the q_i varies inversely with the size of absolute value $|e_i|$. It is well-known that an exponential operation is highly useful in dealing with a similarity relation (cf. Zadeh 1971), Shannon entropy (cf. Pal and Pal 1991, 1992) and in cluster analysis (cf. Wu and Yang 2002; Yang and Wu 2004). We therefore choose

$$q_i \propto \exp(-|e_i|).$$

That is,

$$q_i = \frac{\exp(-|e_i|)}{\sum_{j=1}^{n} \exp(-|e_j|)}, \; i = 1, \ldots, n.$$

Based on the above discussion, we give the weighted bootstrap algorithm as follows:

Weighted Bootstrap Algorithm:

(S1) Define the resampling probability distribution $\{q_i | i = 1, \ldots, n\}$ of $\mathcal{B}_0 = \{(x_i;y_i) | i = 1, \ldots, n\}$ to be

$$q_i = \frac{\exp(-|e_i|)}{\sum_{j=1}^{n} \exp(-|e_j|)}, \; i = 1, \cdots, n.$$

(S2) With the original sample $\mathcal{B}_0 = \{(x_i; y_i) | i = 1, \ldots, n\}$ fixed, draw a 'bootstrap sample' of size n called $\mathcal{B}^\dagger = \{(x_i^\dagger; y_i^\dagger) | i = 1, \ldots, n\}$, under resampling probability distribution $\{q_i | \; i = 1, \ldots, n\}$. For this bootstrap sample to estimate θ by minimizing $\sum_{i=1}^{n} (y_i^\dagger - y(x_i^\dagger; \theta))^2$, we get $\hat{\theta}^\dagger$. Then we have the mean of the squares of the residuals on the test base $\mathcal{B}_0$:

$$TMSE = \frac{1}{n} \sum_{i=1}^{n} \left(y_i^\dagger - y(x_i^\dagger; \hat{\theta}^\dagger) \right)^2.$$

(S3) Repeat S2 B times, we obtain B bootstrap replications corresponding to each bootstrap sample:

$$TMSE(1), \ldots, TMSE(B).$$

Thus, we get the mean value and the standard deviation of the B bootstrap replications:

$$\mu_{\mathrm{boot}} = \frac{1}{B} \sum_{b=1}^{B} TMSE(b),$$

$$\sigma_{\mathrm{boot}} = \left(\frac{1}{B-1} \sum_{b=1}^{B} (TMSE(b) - \mu_{\mathrm{boot}})^2 \right)^{1/2}.$$

Downloaded by [National Chiao Tung University] at 19:13 25 April 2014

The question is here: is there another resampling probability distribution of $\mathcal{B}_0$? The following is an investigation into this question. The answer is affirmative. The generalised bootstrap uses a different heuristic. It consists mainly of generating a random probability measure $\tilde{F}$ from the empirical one. The probability measure $\tilde{F}$ puts some random weight W_i on each data point (x_i, y_i), where $\sum_{i=1}^{n} W_i = 1$. The choices for the weights may be found in Praestgaard and Wellner (1993). The recent work of Barbe and Bertail (1995) involving Edgeworth expansions for a class of weighted bootstrap versions of general von Mises differentiable functionals provides the basis for this study. Prior to their work, most of the results on Edgeworth expansions for the weighted bootstrap has been relegated to the case of the sample mean. Related works include papers by Weng (1989), Haeusler *et al.* (1992), Lo (1993), Hall and Mammen (1994) and Guillou (1995). Their results indicated that one could indeed choose weights which are as accurate as the classical bootstrap in approximating the sampling distribution of the sample mean. Shao and Tu (1995) gave a survey of weighted bootstrap methods in other settings as well. The random weighted bootstrap algorithm is as follows:

Random Weighted Bootstrap Algorithm:

(S1) Generate i.i.d. random variables Z_ℓ, $\ell = 1, \ldots, n$ according to a given distribution G with $G(0) = 0$. Define the random weighted bootstrap empirical measures $\tilde{F}$, where $\tilde{F}$ puts the random weight

$$W_i = \frac{Z_i}{\sum_{\ell=1}^{n} Z_\ell}$$

on each data point (x_i, y_i), $i = 1, \ldots, n$.

The remainder follows the steps (S2) and (S3) of the weighted bootstrap algorithm.

According to equation (3), we have

$$\mu_{\text{boot}} = \frac{1}{B}\sum_{b=1}^{B} TMSE(b) \to \frac{1}{n}\sum_{i=1}^{n} e_i^2 \,(\text{say } MSE), \text{ as } B \to \infty.$$

Usually, MSE is an estimate of σ^2 and, by using the law of large number, we have

$$MSE = \frac{1}{n}\sum_{i=1}^{n} e_i^2 \to \sigma^2, \text{ as } n \to \infty.$$

It is natural to pose the question: 'how accurate is MSE?'. σ_{boot} is the bootstrap procedure for estimating the standard error of MSE from the observed dataset $\mathcal{B}_0$. Notice that a good model should have the small μ_{boot} and σ_{boot}. Therefore, to choose between several models $M_1, M_2, \ldots$, the best one will be the one that has the best compromise to simultaneously minimise μ_{boot} and σ_{boot}.

3. Examples

In this section, we present several examples to compare the original bootstrap (OB), weighted bootstrap (WB) and random weighted bootstrap (RWB) algorithms. In RWB algorithm, we choose Z_ℓ to be uniform $U(0, 1)$ or $\Gamma(4, 2)$ random variables, which corresponds to the Bayesian bootstrap (BB) (cf. Rubin 1981) or Bayesian bootstrap clones (BBC) (cf. Lo 1991). According to Efron and Tibshirani's (1993) experience, there are two rules of thumb to evaluate how large B should be to evaluate μ_{boot} and σ_{boot}: (i) even a small number of bootstrap replications, say $B = 25$, is usually informative, while $B = 50$ is often enough to use; (ii) only very seldom are more than $B = 200$ replications needed for computing μ_{boot} and σ_{boot}. Therefore, we consider $B = 50$ and 100 for OB algorithm. However, WB is an efficient bootstrap computation. In some cases, it can achieve a small error for a given number of function evaluations B, or equivalently, require a smaller value of B to achieve a specified accuracy. Thus, $B = 25$ and 50 are considered in WB, BB and BBC algorithms.

Example 1 Consider the problem of fitting a polynomial model:

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_p x^p + \epsilon.$$

A dataset $\mathcal{B}_0$ is generated by putting

$$x_i = i^{1/3}, \;\; y_i = 4 + x_i + 2x_i^2 + 3x_i^3 + \epsilon_i, i = 1, \ldots, 500,$$

where ϵ_i is a random error term which has the standard normal distribution. We consider three models

Model M_1 : $p = 1, \; y = \theta_0 + \theta_1 x + \epsilon.$

Model M_2 : $p = 2, \; y = \theta_0 + \theta_1 x + \theta_2 x^2 + \epsilon.$

Model M_3 : $p = 3, \; y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \epsilon,$ (true model).

For each model, we also compute μ_{boot} and σ_{boot} based on OB, WB, BB and BBC algorithms. The results are listed in table 1, indicating that the best model for WB algorithm is M_3 with $B = 25, 50$. It is natural to pose the question: 'which one is appropriate?'. Since the difference between 1.0872 and 1.0698 is negligible,

Table 1. Comparison results of OB, BB, BBC and WB algorithms with different bootstrap replications B.

	B	M_1		M_2		M_3	
		μ_{boot}	σ_{boot}	μ_{boot}	σ_{boot}	μ_{boot}	σ_{boot}
WB	25	21.3598	1.8513	1.1032	0.0574	**1.0872**	**0.0554**
	50	21.3293	1.8046	1.0901	0.0561	1.0698	0.0518
BB	25	25.0765	2.3519	1.2424	0.0935	1.2620	0.0940
	50	22.4747	2.1096	**1.1135**	**0.0839**	1.1311	0.0843
BBC	25	24.9511	2.3425	1.2374	0.0931	1.2557	0.0935
	50	22.3623	2.0991	**1.1090**	**0.0835**	1.1266	0.0839
OB	50	22.4298	2.1074	**1.1113**	**0.0838**	1.1288	0.0842
	100	22.3245	1.8893	1.1032	0.0801	1.1144	0.0811

Note: Bold values represent the smallest value in each row.

Downloaded by [National Chiao Tung University] at 19:13 25 April 2014

we choose $B=25$ for the WB algorithm. However, the best model for OB algorithm is M_2, and the difference between 1.1113 and 1.1032 is also negligible. Thus, we choose $B=50$ for the OB algorithm. On the other hand, comparing $B=25$ with $B=50$ in BB and BBC algorithms, we find that the values of μ_{boot} and σ_{boot} are larger with $B=25$. Hence, we choose $B=50$ for BB and BBC algorithms and the best model is M_2. But M_2 is not a true model. Therefore, the WB algorithm not only permits a reduction in replication size but also selects the correct model.

Example 2 Kallel *et al*. (2002) used equation (1) with sigmoid transfer function ϕ to simulate a dataset

$$\mathcal{B}_0 = (x_1^{(i)}, x_2^{(i)}, y_i), \ i = 1, \ldots, 500$$

by computing y_i as a noisy output of a multilayer perceptron, defined by $p=2$ input variables, $x_1 \sim N(0.2, 4)$, $x_2 \sim N(-0.1, 0.25)$, there is one hidden layer and 4 neurons on the hidden layer, $\theta = (0.5, -0.1, 0.2, 0.5, -0.4, 0.2, 0.1, 3, 0.3, 2, 0.5, 0.1, 0.2, 2, 0.2, 3, 0.1)$, as defined in section 1, $\in \sim N(0,0.04)$. They considered three models:

Model M_2: two inputs, one hidden layer with 2 hidden neurons.

Model M_4: two inputs, one hidden layer with 4 hidden neurons: true model.

Model M_6: two inputs, one hidden layer with 6 hidden neurons.

For each model, we compute μ_{boot} and σ_{boot} based on OB, BB, BBC and WB algorithms. Table 2 shows that the best model for these algorithms is M_2. However, the number of bootstrap replications B equals 50 for OB, BB and BBC algorithms. This is because the values of μ_{boot} and σ_{boot} are larger with $B=25$ in BB and BBC algorithms. But the WB algorithm only needs $B=25$. Although M_2 is not a true model, it is the best because the multilayer perceptrons are generally overparameterised. Therefore, it is not surprising that M_2 is selected as the best.

Example 3 In this example, we discuss a real dataset from Draper and Smith (1998, p. 296). A proposed model for this dataset, based on theoretical considerations, is

$$\log_{10} Y = \log_{10} \alpha + \beta \log_{10} X_1 + \gamma \log_{10} X_2 + \delta \log_{10} X_3 + \epsilon,$$

where $\alpha = 0.05$. In the following, we consider four models

Model M_{123}: $\log_{10} Y = \log_{10} \alpha + \beta \log_{10} X_1 + \gamma \log_{10} X_2 + \delta \log_{10} X_3 + \epsilon$ (true model).

Model M_{12}: $\log_{10} Y = \log_{10} \alpha + \beta \log_{10} X_1 + \gamma \log_{10} X_2 + \epsilon$.

Model M_{13}: $\log_{10} Y = \log_{10} \alpha + \beta \log_{10} X_1 + \delta \log_{10} X_3 + \epsilon$.

Model M_{23}: $\log_{10} Y = \log_{10} \alpha + \gamma \log_{10} X_2 + \delta \log_{10} X_3 + \epsilon$.

For each model, we compute μ_{boot} and σ_{boot} based on OB, BB, BBC and WB algorithms. Table 3 shows that the best model for these algorithms is M_{123}. Besides, the values of μ_{boot} and σ_{boot} are larger with $B=25$ in BB and BBC algorithms. This means that $B=25$ is not suitable in BB and BBC algorithms. Therefore, we choose $B=50$ for OB, BB and BBC algorithms. But the WB algorithm only needs $B=25$. It indicates that the WB algorithm does permit a reduction in replication size.

4. Conclusions

To reduce computer effort of the bootstrap method for neural model selection, we propose the WB algorithm based on exponential operation on the absolute value of residuals. Compared with OB, BB and BBC algorithms,

Table 2. Comparison results of OB, BB, BBC and WB algorithms with different bootstrap replications *B*.

	B	M_2 μ_{boot}	M_2 σ_{boot}	M_4 μ_{boot}	M_4 σ_{boot}	M_6 μ_{boot}	M_6 σ_{boot}
WB	25	**0.0422**	**0.0027**	0.0433	0.0028	0.0440	0.0029
	50	0.0420	0.0026	0.0424	0.0027	0.0426	0.0029
BB	25	0.0480	0.0030	0.0489	0.0032	0.0499	0.0034
	50	**0.0429**	**0.0027**	0.0439	0.0029	0.0447	0.0031
BBC	25	0.0482	0.0030	0.0490	0.0032	0.0500	0.0034
	50	**0.0430**	**0.0027**	0.0440	0.0029	0.0449	0.0031
OB	50	**0.0431**	**0.0027**	0.0439	0.0029	0.0448	0.0031
	100	0.0430	0.0026	0.0435	0.0028	0.0447	0.0030

Note: Bold values represent the smallest value in each row.

Table 3. Comparison results of OB, BB, BBC and WB algorithms with different bootstrap replications *B*.

	B	M_{123} μ_{boot}	M_{123} σ_{boot}	M_{12} μ_{boot}	M_{12} σ_{boot}	M_{13} μ_{boot}	M_{13} σ_{boot}	M_{23} μ_{boot}	M_{23} σ_{boot}
WB	25	**0.0242**	**0.0050**	0.0275	0.0057	0.3070	0.0530	0.0285	0.0052
	50	0.0237	0.0040	0.0273	0.0053	0.3053	0.0490	0.0277	0.0047
BB	25	0.0281	0.0059	0.0308	0.0077	0.3572	0.0657	0.0319	0.0064
	50	**0.0250**	**0.0053**	0.0275	0.0070	0.3179	0.0585	0.0286	0.0057
BBC	25	0.0283	0.0058	0.0309	0.0076	0.3504	0.0659	0.0320	0.0063
	50	**0.0251**	**0.0052**	0.0275	0.0070	0.3122	0.0586	0.0287	0.0057
OB	50	**0.0250**	**0.0053**	0.0275	0.0069	0.3192	0.0587	0.0285	0.0057
	100	0.0244	0.0050	0.0273	0.0061	0.3101	0.0554	0.0278	0.0053

Note: Bold values represent the smallest value in each row.

the numerical results show that the WB algorithm not only is an effective means of reducing the bootstrap replications but also selects the correct model. Therefore, the proposed algorithm should be considered in the neural model selection.

References

P. Barbe and P. Bertail, "*The Weighted Bootstrap*," Lecture Notes in Statist, 98, New York: Spinger, 1995.

N.R. Draper and H. Smith, *Applied Regression Analysis*, New York: John Wiley, 1998.

B. Efron, "Bootstrap methods: another look at the jackknife", *Ann. Statist*., 7, pp. 1–26, 1979.

B. Efron and R. Tibshirani, *An Introduction to Bootstrap*, New York: Chapman and Hall, 1993.

A. Guillou, "Weighted bootstraps for studentized statistics", *C.R. Acad. Sci. Paris Sér. I Math*., 320, pp. 1379–1384, 1995.

E. Haeusler, D. Mason and N.A. Newton, "Weighted bootstrapping of means", *CWI Quarterly*, 5, pp. 213–228, 1992.

P. Hall and P. Mammen, "On general resampling algorithms and their performance in distribution estimation", *Ann. Statist*., 22, pp. 2011–2030, 1994.

A.Y Lo, "Bayesian bootstrap clones and a biometry function", *Sankhya Ser, A*, 53, pp. 320–333, 1991.

A.Y Lo, "A Bayesian method for weighted sampling", *Ann. Stat*., 21, pp. 2138–2148, 1993.

R. Kallel, M. Cottrell and V. Vigneron, "Bootstrap for neural model selection", *Neurocomputing*, 48, pp. 175–183, 2002.

N.R. Pal and S.K. Pal, "Entropy, a new definition and its applications", *IEEE. Trans. Syst. Man Cybernet*, 21, pp. 1260–1270, 1991.

N.R. Pal and S.K. Pal, "Some properties of the exponential entropy", *Inform. Sci*., 66, pp. 119–137, 1992.

J.T. Praestgaard and J.A. Wellner, "Exchangeably weighted bootstraps of the general empirical process", *Ann. Probab*., 21, pp. 2053–2086, 1993.

D.R. Rubin, "The Bayesian bootstrap", *Ann. Statist*., 9, pp. 130–134, 1981.

J. Shao and D. Tu, *The Jacknife and Bootstrap*, New York: Springer, 1995.

C.S. Weng, "On a second-order property of the Bayesian bootstrap mean", *Ann. Stat*., 17, pp. 705–710, 1989.

K.L. Wu and M.S. Yang, "Alternative c-means clustering algorithms", *Pattern Recognition*, 27, pp. 2267–2278, 2002.

M.S. Yang and K.L. Wu, "A similarity-based robust clustering method", *IEEE Trans. Patt. Anal. Machine Intelli*., 26, pp. 434–448, 2004.

Z.A. Zadeh, "Similarity relations and fuzzy orderings", *Inform. Sci*., 3, pp. 177–200, 1971.

Downloaded by [National Chiao Tung University] at 19:13 25 April 2014

Shun-Chin Chuang received his B.S. degree in Department of Computer Science and Information Engineering in 1991 from Tamkang University, Taiwan, and his M.S. degree in Computer Science and Information Engineering in 1996 from National Chiao-Tung University, Taiwan. He is now a PhD candidate in Computer Science and Information Engineering Department at National Chiao-Tung University. His research interests include pattern recognition, neural network, content-based image/video retrival.

Dr Wen-Liang Hung Professor of Computer Science, received his PhD degree in Statistics from National Central University, Chungli, Taiwan, in June 1997. His research interests include clustering analysis, fuzzy set theory, soft computing and pattern recognition. Recently he is very interested in applying clustering analysis to astrophysics data system.

Hsin-Chia Fu received his B.S. degree from National Chiao-Tung university in electrical and Communication engineering in 1972, and his M.S. and PhD degrees from New Mexico State University, both in Electrical and Computer Engineering in 1975 and 1981, respectively. His research interests include digital signal/image processing, Multimedia information processing, and neural networks.

Downloaded by [National Chiao Tung University] at 19:13 25 April 2014