FITTING SPARSE GAUSSIAN FUNCTION MIXTURES ON POWER SPECTRA

Luis Weruaga

Electrical & Computer Engineering Dept.
Khalifa University, United Arab Emirates
E-mail: weruaga@ieee.org

Javier Vía

Dept. of Communications Engineering
Universidad de Cantabria, Spain
E-mail: jvia@gtas.dicom.unican.es

ABSTRACT

This paper presents a novel method for fitting a sparse Gaussian mixture on a non-negative function of reference. Despite this problem is well known to be highly non-linear, the use of the logarithmic utility function seems to alleviate such an undesired situation. Moreover, sparsity measures related to the Gaussian precision can be integrated in such a way that the numerical algorithm turns out easy to implement, it is numerically efficient, and presents stable convergence. The method has been originally thought for the parameterization of power spectra, but it can also be useful in different scenarios.

Index Terms— Gaussian function mixture, function approximation, sparsity, optimization.

1. INTRODUCTION

The Gaussian function has many appealing properties, such as in universal function approximation [1], minimum time–frequency dispersion [2], and outcome of the *central limit theorem*. Fitting a Gaussian function mixture (GFM) on data has proven very useful in selected problems in signal processing and electrical engineering [3]-[8]. Despite function approximation and data regression with GFM counts with solid analytical foundations [9]-[11], several research questions still remain open, such as the update of the Gaussian variance/width, or the development of mechanisms that promote sparsity in the mixture.

In the so-called radial basis functions (RBF) [9] for instance, updating the Gaussian width with gradient descent yields unstable or slow convergence [10, 12], and in Gaussian "kernel" regression [11], the variance must be chosen in advance. Several efforts have been done to alleviate those limitations, such as Kalman filtering [13], growing and pruning [14, 15], self-organization [16], covariance update on dense regular node tiling [17], and others [18, 19]. In spite of the previous efforts, and several others not included here due to space constraints, Gaussian variance adaptation and sparsity (as will be shown later, both concepts are interrelated) have not been sufficiently explored. In consequence, the current GFM techniques cannot beat the "curse of dimensionality".

This paper presents a novel method that seems to overcome the limitations of previous GFM techniques. The fitting criterion is based on the generalized logarithmic utility function [20], which can move smoothly between mean square error (MSE) and log-error minimization. In this specific scenario, the estimation of all parameters of the Gaussian mixture can be achieved by solving a sequence of least squares problems. Moreover, sparsity-promoting measures can be easily integrated in the original formulation. Although the work tackles one-dimensional functions, it can be easily extended to multidimensional Gaussian mixtures. The paper organization follows: the problem is stated in Sec. 2, Sec. 3 presents the numerical algorithm, comparison with previous methods is brought in Sec. 4, the empirical validation on speech spectra is found in Sec. 5, and the conclusions close the paper.

2. PROBLEM FORMULATION

We aim to approximate a nonnegative univariate function of reference $y(x) \geq 0$ as a non-negative sum of Gaussian activation functions at points x

$$f(x) = \sum_{k=1}^{K} w_k g(x, \mu_k, \eta_k), \tag{1}$$

where $w_k \geq 0$, and $g(x, \mu, \eta)$ is defined for $\eta \geq 0$ as

$$g(x, \mu, \eta) = \exp\left(-\eta(x-\mu)^2\right). \tag{2}$$

In this paper, we propose a cost function based on the generalized logarithmic utility function [20]

$$l_c(z) = \log(z + c), \tag{3}$$

where $c \geq 0$. In particular, we aim to minimize

$$J(\boldsymbol{v}, \boldsymbol{\mu}, \boldsymbol{\eta}) = \frac{1}{2} \sum_x e^2(x), \tag{4}$$

where

$$e(x) = l_c\big(y(x)\big) - l_c\big(f(x)\big). \tag{5}$$

The vectors of log-weights, centers, and precisions, are respectively defined as $\boldsymbol{v} = [v_1, \ldots, v_K]^T$ with $v_k = \log w_k$,

$\boldsymbol{\mu} = [\mu_1, \ldots, \mu_K]^T$, and $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_K]^T$. The parameter c in (5) allows us to smoothly move from the traditional case of MSE minimization ($c \to \infty$) [10], to the minimization of the mean square log error ($c = 0$), which appears as the "perceptually-linear" scale in speech and audio processing.

It is important to point out that the selection of the order K is a challenging problem. Of course, in some particular applications the parameter K could be *a priori* known. However, we will focus on the general case in which K is unknown, and only an upper bound is available. Thus, we propose to find the optimal number of Gaussians by promoting sparsity in the precision vector $\boldsymbol{\eta}$, which will result in several linearly dependent Gaussian shapes (those with $\eta_k = 0$), that can be replaced by only one zero-precision Gaussian.

According to the previous discussion, the optimization problem to solve is

$$\begin{aligned} &\underset{\boldsymbol{v},\boldsymbol{\mu},\boldsymbol{\eta}}{\text{minimize}} \quad J(\boldsymbol{v},\boldsymbol{\mu},\boldsymbol{\eta}) + \lambda \mathbf{1}^T\boldsymbol{\eta}, \\ &\text{subject to} \quad \boldsymbol{\eta} \geq 0 \end{aligned} \tag{6}$$

where $\mathbf{1}^T$ is the all-one row vector, λ is the regularization constant, and $\mathbf{1}^T\boldsymbol{\eta} = \|\boldsymbol{\eta}\|_1$ is the convex envelope of the sparsity-promoting l_0-norm $\|\boldsymbol{\eta}\|_0$.

3. PROPOSED ALGORITHM

Excluding the case of mean square log error ($c = 0$) with only one Gaussian shape ($K = 1$, $\lambda = 0$), which can be easily convexified by means of a simple reparameterization [21], the optimization problem in (6) is non-convex in general, and therefore difficult to solve. Here, we aim to find a local solution by solving the Karush-Kuhn-Tucker (KKT) conditions [22]. In order to do that, we write the Lagrangian of (6) as

$$\mathcal{L}(\boldsymbol{v},\boldsymbol{\mu},\boldsymbol{\eta},\boldsymbol{\gamma}) = J(\boldsymbol{v},\boldsymbol{\mu},\boldsymbol{\eta}) + \lambda \mathbf{1}^T\boldsymbol{\eta} - \boldsymbol{\gamma}^T\boldsymbol{\eta}, \tag{7}$$

where $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_K]$ is a vector of Lagrange multipliers. Thus, the KKT conditions can be written as

$$\begin{aligned} &\text{Gradient of the Lagrangian:} && \nabla_k \mathcal{L}(\boldsymbol{v},\boldsymbol{\mu},\boldsymbol{\eta},\boldsymbol{\gamma}) = 0 \quad \forall k \\ &\text{Primal feasibility:} && \boldsymbol{\eta} \geq 0, \\ &\text{Dual feasibility:} && \boldsymbol{\gamma} \geq 0, \\ &\text{Complementary Slackness:} && \boldsymbol{\gamma}^T\boldsymbol{\eta} = 0, \end{aligned}$$

where $\nabla_k = \big[\partial/\partial v_k, \partial/\partial \mu_k, \partial/\partial \eta_k\big]^T$.

3.1. Implementation Details

The gradient of the Lagrangian (7) results in

$$\nabla_k \mathcal{L} = -\sum_x \rho_k(x)\boldsymbol{a}_k(x)e(x) + \big[0,\, 0,\, \lambda - \gamma_k\big]^T, \tag{8}$$

where

$$\boldsymbol{a}_k(x) = \big[1,\, 2\eta_k(x-\mu_k),\, -(x-\mu_k)^2\big]^T, \tag{9}$$

and

$$\rho_k(x) = \frac{w_k g(x,\mu_k,\eta_k)}{f(x) + c} \tag{10}$$

is the "relevance" of the kth activation function at x, which satisfies $0 < \rho_k(x) < 1$. We can rewrite the error term around the kth activation function as

$$e(x) = e_k(x) + \eta_k(x-\mu_k)^2 - v_k, \tag{11}$$

where

$$e_k(x) = \log\big[y(x) + c\big] + \log\big[\rho_k(x)\big] \tag{12}$$

provides a measure of the residual error when the kth Gaussian shape is not included in the model. With the previous simplifications, the gradient (8) can be rewritten as

$$\nabla_k \mathcal{L} = \boldsymbol{y}_k - \boldsymbol{H}_k\boldsymbol{z}_k - \big[0,\, 0,\, \gamma_k\big]^T, \tag{13}$$

for $k = 1, \cdots, K$, where

$$\boldsymbol{y}_k = -\sum_x \rho_k(x)\boldsymbol{a}_k(x)e_k(x) + \big[0,\, 0,\, \lambda\big]^T, \tag{14a}$$

$$\boldsymbol{H}_k = \sum_x \rho_k(x)\boldsymbol{a}_k(x)\big[x^2, x, 1\big], \tag{14b}$$

and

$$\boldsymbol{z}_k = \big[\eta_k,\, -2\eta_k\mu_k,\, \eta_k\mu_k^2 - v_k\big]^T. \tag{15}$$

At this point, one can realize that the KKT conditions result in a complicated system of nonlinear equations: the terms $\boldsymbol{a}_k(x)$ and $\rho_k(x)$ depend on all the parameters to be found. In order to obtain the unknowns v_k, μ_k, η_k and γ_k from the null of the gradient (13), the terms $\boldsymbol{H}_k$ and $\boldsymbol{y}_k$ are considered constant, such that the vector $\boldsymbol{z}_k$ is easily obtained by solving a least squares problem.[1] This way of proceeding resembles a quasi-Newton recursion, in which the gradient is approximated by a linear function of the parameters, and the solution is accomplished by reevaluating the constant terms with the partial solution. In each iteration, we need however to take into account the complementary slackness condition, which yields one of the following two cases: 1) when the non-negativity constraint is inactive ($\gamma_k = 0$), the problem reduces to minimize $\|\boldsymbol{y}_k - \boldsymbol{H}_k\boldsymbol{z}_k\|$, which can be easily done by means of the generalized inverse of $\boldsymbol{H}_k$; once the solution is obtained we check whether $\eta_k \geq 0$ and $\|\boldsymbol{y}_k - \boldsymbol{H}_k\boldsymbol{z}_k\| = 0$, if any of these conditions is not satisfied, we need to consider the second scenario; 2) if the non-negativity constraint is active ($\eta_k = 0$), we need to solve the least squares problem

$$\underset{v_k,\gamma_k}{\text{minimize}} \; \Big\| \boldsymbol{y}_k - \boldsymbol{H}_k\big[0,\, 0,\, -v_k\big]^T - \big[0,\, 0,\, \gamma_k\big]^T \Big\|^2, \tag{16}$$

and, if the solution provides a $\gamma_k < 0$, we will just set $\gamma_k = 0$ and $v_k = -\boldsymbol{y}_k^T\boldsymbol{h}_k^{(3)}/\|\boldsymbol{h}_k^{(3)}\|^2$, where $\boldsymbol{h}_k^{(3)}$ denotes the third column of $\boldsymbol{H}_k$.

[1] Obviously, once the vector $\boldsymbol{z}_k$ is obtained, the original parameters v_k, μ_k, η_k can be easily recovered.

3.2. Additional Remarks

The Hessian matrix that derives from the Lagrangian (7) allows us to better understand the complexity of the problem. From the formula of the gradient (8), it is easy to deduce that the Hessian elements that correspond to different activation functions are weighted by the product of the respective relevance functions (10). Given the local character of the relevance function, the resulting Hessian matrix is sparse (and nearly block-diagonal). Thus, splitting the problem into K parallel subproblems (13) is a reasonable efficient approach. On the other hand, the properties of the proposed iterative algorithm can be identified from the exact Hessian submatrix for the kth activation function, which results in

$$\begin{aligned}\nabla^2_k \mathcal{L} = &\sum_x \rho_k^2(x)\, \boldsymbol{a}_k(x)\boldsymbol{a}_k^T(x) \\ &- \sum_x \rho_k(x)\big(1-\rho_k(x)\big)\, \boldsymbol{a}_k(x)\boldsymbol{a}_k^T(x)\, e(x) \\ &- \sum_x \rho_k(x) \begin{bmatrix} 0 & 0 & 0 \\ 0 & -2\eta_k\mu_k & 2(x-\mu_k) \\ 0 & 2(x-\mu_k) & 0 \end{bmatrix} e(x).\end{aligned} \tag{17}$$

The first term in the Hessian represents its main positive semidefinite contribution, while the other two, that cannot be guaranteed such, depend on the error $e(x)$. Hence, assuming a small-noise scenario, both terms nearly vanish in the vicinity of the solution. It is interesting to mention that these Hessian terms correspond to the parametric elements kept constant during the quasi-Newton iteration and reevaluated thereafter. Finally, the positive semidefinite Hessian term reveals an interesting aspect, namely, each point x is weighted by the *square* of the relevance function (10). This fact invites us to redefine the approximate Hessian matrix $\boldsymbol{H}_k$ and reference vector $\boldsymbol{y}_k$ in (14) accordingly. This alternative has been proven experimentally to yield faster speed of convergence. The overall method is summarized in Algorithm 1.

Algorithm 1 Proposed Weighted Sum of Gaussian Algorithm

Input:x, $y(x)$, c, λ and K.
Output: Parameters $\boldsymbol{v}$, $\boldsymbol{\eta}$, $\boldsymbol{\mu}$ of the model.
Initialize $\boldsymbol{v}$, $\boldsymbol{\eta}$ and $\boldsymbol{\mu}$.
repeat
  Obtain $f(x)$
  for $k = 1, \ldots, K$ **do**
    Obtain $\rho_k(x)$, $\boldsymbol{a}_k(x)$ and $e_k(x)$.
    Obtain $\boldsymbol{y}_k$ and $\boldsymbol{H}_k$ as
      $\boldsymbol{y}_k = -\sum_x \rho_k(x)^2 \boldsymbol{a}_k(x) e_k(x) + \big[0,\, 0,\, \lambda\big]^T$
      $\boldsymbol{H}_k = \sum_x \rho_k(x)^2 \boldsymbol{a}_k(x) \big[x^2, x, 1\big]$
    Minimize $\|\boldsymbol{y}_k - \boldsymbol{H}_k \boldsymbol{z}_k\|$ to obtain $\boldsymbol{z}_k$.
    if $\boldsymbol{z}_k(1) < 0$ or $\|\boldsymbol{y}_k - \boldsymbol{H}_k \boldsymbol{z}_k\| > 0$ **then**
      Set $\boldsymbol{z}_k(1) = \boldsymbol{z}_k(2) = 0$.
      Obtain $\boldsymbol{z}_k(3) = -v_k$ and γ_k by solving (16).
      if $\gamma_k < 0$ **then**
        Set $\gamma_k = 0$.
        Obtain $-\boldsymbol{z}_k(3) = v_k = -\boldsymbol{y}_k^T \boldsymbol{h}_k^{(3)} / \|\boldsymbol{h}_k^{(3)}\|^2$.
      end if
    end if
    Recover the original parameters v_k, μ_k, η_k from $\boldsymbol{z}_k$.
  end for
until Convergence

4. COMPARATIVE ANALYSIS

The most popular methods for regression with a Gaussian function mixture are the so-called radial basis functions (RBF) [10] and the support-vector "kernel" regression (SVR) [11]. In RBFs [9], the estimation of the parametric mixture is attempted with a MSE-driven gradient descent [10, 12]. Its main drawback however lies on the update of the precision η, as the gradient turns out highly non-linear. Assuring a stable convergence is thus a challenging task, and, apart from exhaustive regular tiling [17], in practice only weights w and centers μ are updated [15, 16]. SVR [11] presents comparable drawbacks, as the variance of the Gaussian kernel must be selected in advance, before the training process starts. Despite both methods are not limited to non-negative functions, the previous drawback prevents them to reach the type of sparse solutions that our method promotes.

Given that function $y(x)$ is non-negative and summable, it can be treated as a probability density and be decomposed as a Gaussian mixture model (GMM) by means of the expectation maximization (EM) algorithm [23]. This popular iterative method can be rewritten as

$$\rho_k(x) = \frac{w_k^{(\xi)} g\big(x, \mu_k^{(\xi)}, \eta_k^{(\xi)}\big)}{\sum_{\ell=1}^K w_\ell^{(\xi)} g\big(x, \mu_\ell^{(\xi)}, \eta_\ell^{(\xi)}\big)} \tag{18a}$$

$$p_k(x) = \frac{y(x)\rho_k(x)}{\sum_{x'} y(x')\rho_k(x')} \tag{18b}$$

$$w_k^{(\xi+1)} = \textstyle\sum_x y(x) \sum_x p_k(x) \tag{18c}$$

$$\mu_k^{(\xi+1)} = \textstyle\sum_x x\, p_k(x) \tag{18d}$$

$$2/\eta_k^{(\xi+1)} = \textstyle\sum_x \big(x - \mu_k^{(\xi+1)}\big)^2 p_k(x) \tag{18e}$$

where ξ denotes iteration. Note that the amplitude w_k (instead of its log counterpart v_k) and the variance (inverse of the precision η_k) are direct outcome of the algorithm iteration. Interestingly, both the EM-GMM fitting and our method are built around the relevance of each activation function (10) and (18a). The EM-GMM however treats the data as probabilistic "mass", while the proposed method simply performs a regression. This conceptual difference has a profound impact in the event of "missing data": the EM-GMM implicitly considers $y(x) = 0$ for values x not available during the training, while in our algorithm the missing data is implicitly ignored, hence keeping intact the generalization capabilities.

5. SIMULATION RESULTS

In order to validate qualitatively the proposed method we bring a toy experiment consisting of a non-uniform sampling of a certain smooth function. Figure 1 illustrates this scenario in vertical logarithmic scale. The EM-GMM method (18) and two variants of our technique for $c = 0$ are considered: the basic one ($\lambda = 0$) and that with sparsity measures ($\lambda = 0.01$). All techniques undergo the same initialization and have the same size ($K = 40$): such a number of activation functions exceeds by far the size required for this scenario. From the results shown in Fig. 1 one can conclude that the proposed method efficiently selects a reduced number of Gaussians (about 14 here) while achieving excellent fitting performance; the EM-GMM falters with non-uniformly spaced data (as this case resembles a missing-data situation).

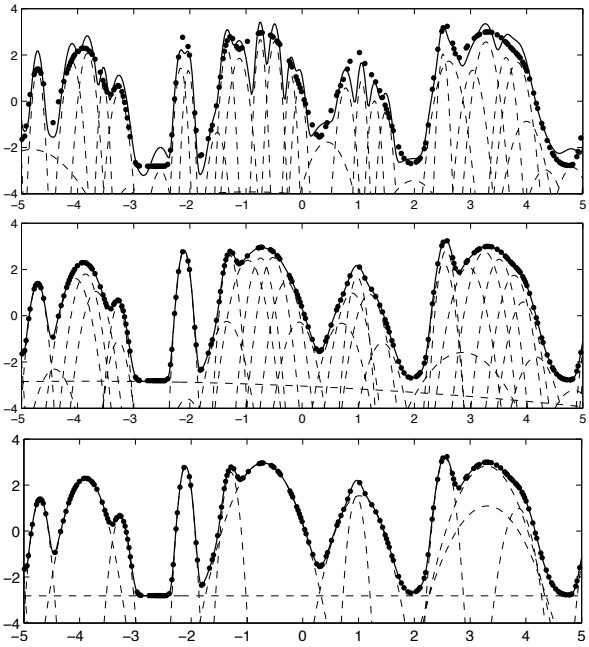

Fig. 1. EM-GMM (top), proposed method with $\lambda = 0$ (middle) and $\lambda = 0.01$ (bottom). Data $y(x)$ (dotted), fitting function $f(x)$ (solid) and activation functions (dashed).

The second scenario selected corresponds to the parameterization of the power spectrum of short segments of speech, shown in Figure 2. The data was obtained from the discrete Fourier transform of the Gaussian-windowed segment (this procedure gives rise to Gaussian-shaped spectral components). The parameters are $K = 40$, $\lambda = 0.01$, and $c = 0$. In the voiced speech segment, the proposed method converges in few iterations (about 10), while the resulting solution approximates faithfully the actual power spectrum. The width of the resulting Gaussian units give an indication of the tonal components in the spectrum. In case of the unvoiced segment,

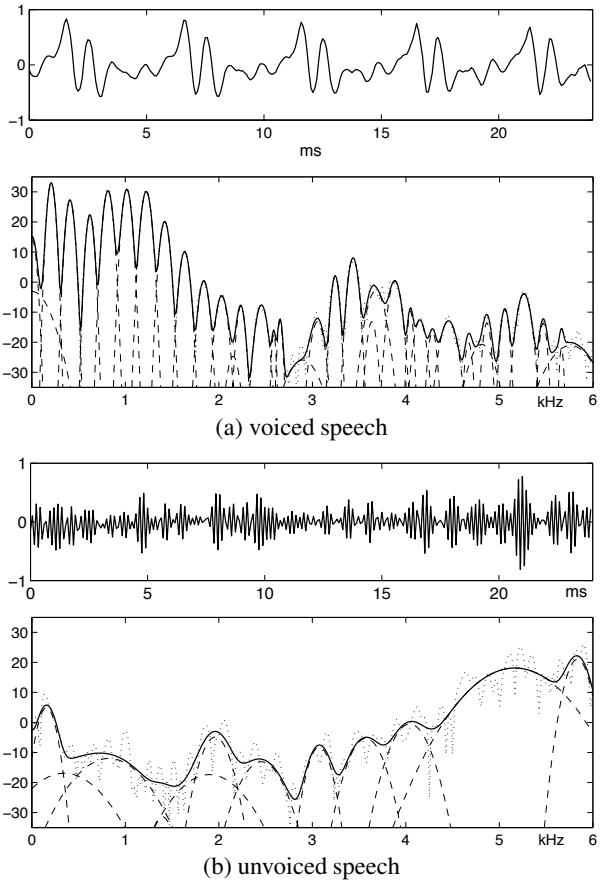

Fig. 2. For each case: top — signal segment, bottom — GFM approximation (solid) of its power spectrum (dotted) and GFM (dashed). Vertical axis in logarithmic scale.

the final mixture is more sparse, this indicating the presence of non-tonal components. The proposed technique is planned to be used within the method [24], proposed recently by one of the authors, to detect and to quantify chirp-tonal/stochastic signal components for audio analysis and coding.

6. CONCLUSIONS

The logarithmic utility function has allowed us to revamp the highly non-linear problem of Gaussian function mixture (GFM) fitting into a sequence of least squares problems. Simulation results on the power spectrum of short speech segments illustrate the qualitative performance of the method. Our current efforts are directed to prove and to improve the convergence of the method, in especial when the sparsity-promoting measures are active, the integration of model-growing strategies, and the extension to multidimensional spaces and to positive/negative functions.

7. REFERENCES

[1] J. Park and I. Sandberg, "Universal approximation using radial-basis function networks," *Neural Computation*, vol. 3, pp. 246–257, Jun. 1991.

[2] L. Cohen, *Time–frequency analysis*. Prentice Hall, 1995.

[3] P. Yee and S. Haykin, "A dynamic regularized radial basis function network for nonlinear, nonstationary time series prediction," *IEEE Trans. Signal Processing*, vol. 47, pp. 2503–2521, Sep. 1999.

[4] S. Seshagiri and H. Khalil, "Output feedback control of nonlinear systems using RBF neural networks," *IEEE Trans. Neural Networks*, vol. 11, pp. 69–79, Jan. 2000.

[5] T. Blu and M. Unser, "Wavelets, fractals, and radial basis functions," *IEEE Trans. Signal Processing*, vol. 50, pp. 543–553, Mar. 2002.

[6] M. Yee, B. Yeap, and L. Hanzo, "Radial basis function-assisted turbo equalization" *IEEE Trans. Communications*, vol. 51, pp. 664–675, Apr. 2003.

[7] C. Tseng and S. Lee, "Design of fractional order digital differentiator using radial basis function," *IEEE Trans. Circuits Syst.*, vol. 57, pp. 1708–1718, Jul. 2010.

[8] J. Jacobs, "Bayesian support vector regression with automatic relevance determination kernel for modelling of antenna input characteristics," *IEEE Trans. Antennas Propagat.*, vol. 60, pp. 2114–2118, Apr. 2012.

[9] S. Chen, C. Cowan, and P. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, vol. 2, pp. 302–309, Mar. 1991.

[10] P. Yee and S. Haykin. *Regularized radial basis function networks: Theory and applications*. Wiley, 2001.

[11] B. Schölkopf and A. Smola. *Learning with kernels*. MIT Press, 2002.

[12] N. Karayiannis, "Reformulated radial basis neural networks trained by gradient descent," *IEEE Trans. Neural Networks*, vol. 10, pp. 657–671, Aug. 2002.

[13] D. Simon, "Training radial basis neural networks with the extended Kalman filter," *Neurocomputing*, vol. 48, pp. 455–475, 2002.

[14] G. Huang, P. Saratchandran, and N. Sundararajan, "A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation," *IEEE Trans. Neural Networks*, vol. 16, pp. 57–67, Jan. 2005.

[15] M. Bortman and M. Aladjem, "A growing and pruning method for radial basis function networks," *IEEE Trans. Neural Networks*, vol. 20, pp. 1039–1045, Jun. 2009.

[16] J. Lian, Y. Lee, S. Sudhoff, and S. Zak, "Self-organizing radial basis function network for real-time approximation of continuous-time dynamical systems," *IEEE Trans. Neural Networks*, vol. 19, pp. 460–474, Mar. 2008.

[17] H. Huan, D. Hien, and H. Tue, "Efficient algorithm for training interpolation RBF networks with equally spaced nodes," *IEEE Trans. Neural Networks*, vol. 22, pp. 982–988, Jun. 2011.

[18] S. Chen, X. Hong, B. Luk, and C. Harris, "Construction of tuneable radial basis function networks using orthogonal forward selection," *IEEE Syst., Man, Cybern.*, vol. 39, pp. 457–466, Apr. 2009.

[19] T. Xie, H. Yu, J. Hewlett, P. Rózycki, and B. Wilamowski, "Fast and efficient second-order method for training radial basis function networks," *IEEE Trans. Neural Networks, Learn. Syst.*, vol. 23, pp. 609–619, Apr. 2012.

[20] C. Friedman, *Utility-based learning from data,* CRC Press, 2010.

[21] H. Guo, "A simple algorithm for fitting a Gaussian function," *IEEE Signal Processing Mag.*, vol. 28, pp. 134–137, Sep. 2011.

[22] S. Boyd and L. Vandenberghe, *Convex optimization,* Cambridge University Press, 2004.

[23] J. A. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," Tech. Rep. 97-021, *Intl. Comp. Sci. Inst.*, Berkeley CA, 1997.

[24] L. Weruaga, "Redundant time–frequency marginals for chirplet decomposition," *IEEE Wksp. Machine Learning for Signal Processing*, 2012, pp. 1–5.