# Auto-Imputing Radial Basis Functions for Neural-Network Turn-Taking Models

*Kornel Laskowski*[1,2]

[1]Carnegie Mellon University, Pittsburgh PA, USA
[2]Voci Technologies Inc., Pittsburgh PA, USA
kornel@cs.cmu.edu

## Abstract

A stochastic turn-taking (STT) model is a per-frame predictor of incipient speech activity. Its ability to make predictions at any instant in time makes it particularly well-suited to the analysis and synthesis of interactive conversation. At the current time, however, STT models are limited by their inability to accept features which may frequently be undefined. Rather than attempting to impute such features, this work proposes and evaluates a mechanism which implicitly conditions Gaussian-distributed features on Bernoulli-distributed indicator features, making prior imputation unnecessary. Experiments indicate that the proposed mechanisms achieve predictive parity with standard model structures, while at the same time offering more direct interpretability and the desired insensitivity to missing feature values.

**Index Terms**: Turn-taking, neural networks, prediction, radial basis functions, imputation.

## 1. Introduction

A stochastic turn-taking (STT) model [1] predicts the probability that a specific conversant will be speaking (versus not speaking) at an instant $t$, given their and their interolocutors' speech activity history prior to $t$. Since their inception [2, 3, 4, 1], these models have come a long way in terms of number of modeled speakers [5], online adaptation [6], and duration of modeled history [7]. Their main strength is the ability to make predictions *at every instant* of a conversation. This renders them suitable for the analysis of truly interactive human behavior in unlabeled — and therefore vast — amounts of conversational data.

One application of STT models is the quantitative study of the role of prosody in shaping the timing of participation to dialogue. The probability that a conversant speaks at instant $t$ may be conditioned not only on past (discrete) speech activity, but also on continuous-valued frame-level features. This was done in [8], where the heretofore standard $N$-gram model was replaced by a neural network with tanh activation functions in the hidden layer. A problem that arises organically in this case is that a number of allegedly interesting articulatory-acoustic and prosodic features may — although continuous-valued — be undefined for individual frames or intervals of frames. Examples are pitch or formant bandwidth. Currently, before a regressor such as a neural network can be applied, undefined values must be "filled in", or *imputed* [9], as depicted in Figure 1.

One solution is to simply replace "missing values" by a default numerical value, such as in the case of pitch trackers which return zero for unvoiced frames. This can be problematic as it disturbs the dynamics and statistics of the analyzed feature. A more principled approach is to impute values using

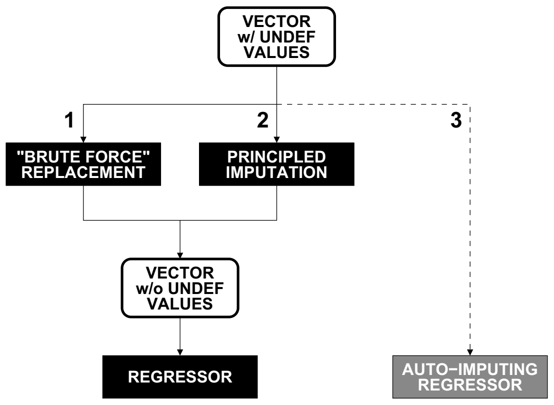

Figure 1: Three alternatives for performing regression using a feature vector which potentially contains undefined values; alternative "3" is the subject of the current article.

extrapolation or model matching, imposing continuity; this in turn prevents regressors from exploiting the fact that the values were actually undefined to begin with. Both approaches (depicted as "1" and "2" in Figure 1, respectively) embody an extra imputation step in processing, whose implementation requires additional expertise and effort and whose execution requires additional computing cycles. The goal of the current work is to sidestep the need for this extra step, by rendering a neural-network regressor capable of auto-imputing its input space as necessary, while retaining the prediction performance of the manually imputed alternative.

The article begins by reviewing the STT methodology, and then defines a feature with potentially missing values for the purposes of demonstration. As a baseline, a method for manually imputing the missing values is borrowed from [8]. Two novel hidden-layer activation functions for neural-network STT models are then proposed: a Bernoulli radial basis function (BRBF) for binary-valued input, and a joint radial basis function (JRBF) which combines the BRBF with a gated form of the more standard Gaussian radial basis function (GRBF). Speaker-independent results on held-out conversations drawn from a large telephony dialogue corpus indicate that the proposed auto-imputing neural networks, which use JRBF units, provide prediction performance parity with a standard neural network implementation requiring prior manual imputation. It is argued that the new method provides a means for quantitatively studying vast amounts of conversational material, for which only a reference speech/non-speech segmentation is required.

# 2. Methods

## 2.1. Turn-Taking Framework

The essential elements of the turn-taking framework (cf. [2, 3, 4, 1, 6, 7]) are shown in Figure 2. The goal is to predict the binary speech activity $\mathbf{q}_t[k] \in \{\square, \blacksquare\}$ for a speaker engaged in dialogue ($K \equiv 2$) at an instant $t$; $k \in \{1, \ldots, K\}$ identifies the party for whom the prediction is being made, referred to as the *target speaker*.

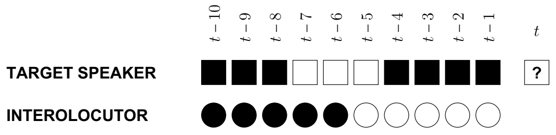

Figure 2: A stochastic turn-taking model predicts the binary speech activity (marked with a "?") at an arbitrary instant $t$ for a target speaker. Squares and circles denote past speech activity for the target speaker and their interlocutor, respectively.

In the current work, all predictions are conditioned on a history consisting of the 10 most recent frames; the frame step is 100 ms. Predictions may be conditioned on past speech activity $\mathbf{q}_{t-10}^{t-1}[k] \in \{\square, \blacksquare\}^{1\times 10}$ of only the target speaker (squares in Figure 2), implying an unconditionally independent (UI) model. They may alternately be conditioned on past speech activity $\mathbf{q}_{t-10}^{t-1} \in \{\square, \blacksquare\}^{2\times 10}$ of both parties (squares *and* circles in Figure 2), implying a conditionally independent (CI) model. The $\mathbf{q}_t$ are the columns of a speech activity matrix $\mathbf{Q}$, of 2 rows and $T$ columns, with $T$ giving the number of frames.

Like a language model in speech recognition, an STT model provides the *prior* probability of conversant speech activity states at instant $t$, *before* any acoustic data for instant $t$ has been observed. STT models are therefore trained using reference state sequences, in this case factored speech activity states (as depicted in Figure 2) rather than words [1]. Because $\mathbf{Q}$ is discrete, STT models have most commonly been implemented as smoothed $N$-gram models [5]. By mapping $\square$ (non-speech) and $\blacksquare$ (speech) to zero and unity, respectively, non-categorical-input regressors can also be used; [8] showed that a standard one-hidden-layer neural network, with tanh activations in the hidden layer, achieves prediction parity with an $N$-gram model, with far fewer parameters but far longer training times.

## 2.2. A Continuous-Valued Feature with Missing Values

To assess the performance of an auto-imputing STT model, the current work explores loudness, whose per-frame values are collected in $\mathbf{F}$, a matrix of dimensions identical to $\mathbf{Q}$. For simplicity, each value of $\mathbf{F}$ is defined as

$$\mathbf{f}_t[k] = \begin{cases} \mathbf{e}_t[k] & \text{if } \mathbf{q}_t[k] = \blacksquare \\ \text{undefined} & \text{otherwise} \end{cases} \tag{1}$$

where $\mathbf{e}_t[k]$ is the signal energy as defined in [8]. The contrast with performance achieved by manually imputing features will use $\tilde{\mathbf{F}}$, the imputed form of $\mathbf{F}$, whose entries are given by

$$\tilde{\mathbf{f}}_t[k] = \begin{cases} 5 + (\mathbf{f}_t[k] - \mu)/\sigma & \text{if } \mathbf{f}_t[k] \text{ defined} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

[1]When coupled with a suitable acoustic model of observations at instant $t$, an STT model could form part of a speech activity detector. In the current work, STT models are studied independently of detection, to learn the "grammar" of interaction sequences.

This $Z$-normalizes $\mathbf{f}_t[k]$ using statistics $\mu$ and $\sigma$ obtained from all of the speech frames in row $k$ of $\mathbf{F}$, first shifting the mean upwards by five standard deviations, leaving all speech frames with positive $\tilde{\mathbf{f}}_t[k]$ values. The non-speech frames are then assigned to a "loudness" of zero. This formula was proposed in [8]; it is noteworthy that while predictions using $\tilde{\mathbf{F}}$ remain causal, the normalization relies on global (and therefore future) values of $\mathbf{F}$. This actually makes this baseline unfairly strong in the present context [2].

## 2.3. Neural Network Architecture and Training

The application of neural networks to stochastic turn-taking models is a relatively new development; prior to [8], turn-taking models were implemented as $N$-grams, owing to the discrete enumerability of conditioning histories, which contained only binary speech activity features. In the current work, as in [8], the probability that pariticipant $k$ speaks at instant $t$ is given by the single output unit of a two-layer feed-forward neural network. Accordingly, the output activation function is the sigmoid, and the error function is the cross entropy error [10]. Training is accomplished using scaled conjugate gradient search (SCG) [11], a second-order technique.

To further accelerate the learning process, networks are trained using SCG for up to 100 iterations using every 1024th exemplar, then for up to 100 iterations using every 512th exemplar, etc., and finally for up to 1000 iterations using every exemplar. For each experiment, a network with $J \in \{2, 4, 8, 16, 32, 64\}$ hidden units is trained using TRAINSET; the best $J$ is selected by minimizing the cross entropy error on DEVSET (cf. Section 3 for a description of data sets). Only that network is applied to TESTSET.

## 2.4. Hidden-Layer Activation Functions

The baseline neural network architecture consists of hidden units whose activation function is the standard [10]

$$h_j = \tanh z_j \tag{3}$$

$$z_j = b_j + \sum_{i=1}^{I} w_{ji} \cdot x_i \ . \tag{4}$$

Each exemplar $\mathbf{x} \equiv \{x_i\}$ leads to updates of the first-layer biases $b_j$ and first-layer weights $w_{ji}$. This function is unsuitable for feature vectors $\mathbf{x}$ which may contain undefined values.

As will be shown, auto-imputation can be achieved by extending a type of activation function known as a radial basis function (RBF) [12, 13]. The most commonly used form is

$$h_j = \exp(-z_j) \tag{5}$$

$$z_j = \sum_{i=1}^{I} b_{ji} \cdot (w_{ji} - x_i)^2 \ . \tag{6}$$

This formulation assumes that the feature density is a weighted sum of diagonal Gaussians, and will henceforth be denoted by "GRBF" [3]. To ensure that $z_j$ remain strictly non-negative during training, Equation 6 was replaced with

$$z_j = r(b_j) \cdot \sum_{i=1}^{I} (w_{ji} - x_i)^2 \ . \tag{7}$$

[2]To see that this is so, note that row 4 in Figure 3 is identical to row 32 in Figure 2 of [8]. In the latter, it had been shown that not using $Z$-normalization results in a hit of approximately 0.01 bits per 100 ms (row 23 in Figure 2 of [8]).

[3]Spherical Gaussian RBFs were also explored, and are a specialized form of the diagonal case in which $b_{ji} = b_i$ for all $j$.

This permits implicating a barrier (cf. [14]); for example, the adopted $r(b) \doteq b_{min}^2 + b^2$ is positive and analytic $\forall b \in \Re$. $b_{min} = 0.0001$ was selected empirically, by observing the evolution of the error for a subset of TRAINSET. These modifications allow for SCG training to proceed without intervention[4].

Gaussian RBFs are appropriate for feature vectors whose elements are continuous-valued, but may be inappropriate for binary-valued elements such as those of $\mathbf{Q}$. For such features, the current work proposes a corresponding *Bernoulli radial basis function* (BRBF),

$$h_j = \sqrt[I]{z_j} \tag{8}$$

$$z_j = \prod_{i=1}^{I} (w_{ji})^{(x_i)} (1 - w_{ji})^{(1-x_i)} \quad . \tag{9}$$

Neural networks with this activation function are closely related to Bernoulli mixture models (BMMs) [15], and in fact RBF neural networks are often trained in two stages; the first stage is an unsupervised maximization of the likelihood of the training data, while the second stage adapts only the second-layer weights. When both sets of weights are adapted in a single stage, as in this work, the first-layer effectively learns "traits" [16]. Since the weights $w_{ji}$ must lie on the unit interval, Equation 9 was replaced with

$$z_j = \prod_{i=1}^{I} (s(w_{ji}))^{(x_i)} (1 - s(w_{ji}))^{(1-x_i)} \quad , \tag{10}$$

where $s(w) \doteq 1/\left(1 + \exp\left(-\beta\left(w - \frac{1}{2}\right)\right)\right)$.

Neither the GRBF nor the BRBF are applicable for feature vectors with potentially undefined values. To extend their operation into auto-imputing territory, each potentially undefined continuous-valued-feature value (such as $x_i \equiv \mathbf{f}_t[k]$) must be accompanied by a defined binary-valued-feature value (such as $\xi_i \equiv \mathbf{q}_t[k]$). Feature $x_i$ is conceived of as being *gated* by the feature $\xi_i$. The proposed Joint Radial Basis Function (JRBF) contains elements of both the GRBF and the BRBF:

$$h_j = \sqrt[I]{z_j} \tag{11}$$

$$z_j = \prod_{i=1}^{I} (s(u_{ji}))^{(\xi_i)} (1 - s(u_{ji}))^{(1-\xi_i)} \cdot \left(\exp\left(-r(b_j) \cdot (w_{ji} - x_i)^2\right)\right)^{(\xi_i)} \quad , \tag{12}$$

where the $u_{ji}$ are the weights applied to feature $\xi$ and the $w_{ji}$ are the weights applied to feature $x$ as gated by feature $\xi$. As can be seen, the $j$th hidden unit is sensitive to both values of $\xi_i$ (zero or unity) — as in the BRBF — but is sensitive to $x_i$ (in a way borrowed from the GRBF) only if $\xi_i$ is unity, as desired.

## 3. Data

Experiments were conducted using the Switchboard-1 Corpus, as re-released in 1997 [17]. It consists of 2435 telephone conversations, each approximately 10 minutes in duration. The corpus was divided into three speaker-disjoint sets in [7], such that TRAINSET, DEVSET, and TESTSET consist of 762, 227, and 199 conversations, respectively. During that process, it was not possible to allocate 1247 conversations because their two speakers had already been placed in different sets. In summary, the numbers of exemplars were 2.6M for TRAINSET, 0.9M for DEVSET, and 0.8M for TESTSET.

Reference speech/non-speech segmentations were used, as elsewhere in speech technology when training prior probability models. $\mathbf{Q}$ was constructed using the forced-alignment-mediated references [18]. $\mathbf{E}$ was computed from channel-separated audio files, as described in Subsection 2.2.

[4]Ad-hoc intervention would otherwise be necessary to ensure that $b_j > 0$ in Equation 6. At each training epoch requiring such intervention, the conjugate gradient direction would need to be reset to the true gradient, undermining the fast convergence of the SCG algorithm.

## 4. Results

### 4.1. Replacement of DOT-TANH by RBFs for Everywhere-defined Binary-Valued Features

In a first experiment, the DOT-TANH activation function is replaced by several alternative RBF functions. This was done to determine whether the RBF formalism, present also in the JRBF context, leads to any reduction in predictive power for feature vectors which do not contain undefined values. The results are shown in rows 1 through 3 of Figure 3, where the feature space consists entirely of past binary-valued speech activity.

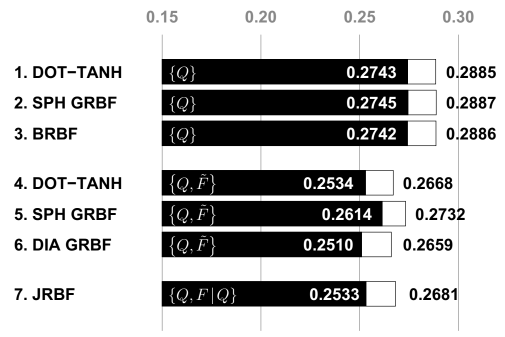

Figure 3: DEVSET cross entropy rates (in bits per 100 ms, along the horizontal axis) for speech activity ($Q$), imputed loudness ($\tilde{F}$), and loudness containing undefined values ($F$), using several types of hidden unit activation functions. White and black bars denote UI and CI models, respectively. "SPH" depicts "spherical covariance" while "DIA" depicts "diagonal covariance"; all other acronyms as in the text.

As can be seen, the spherical GRBF exhibits identical performance to the DOT-TANH baseline from [8], and diagonal covariances are not necessary. More importantly, the BRBF — more appropriate for binary-valued features — also achieves predictive parity with the DOT-TANH network. These findings are somewhat surprising, since DOT-TANH units can form arbitrary linear combinations of the input feature vector elements, while RBF units cannot. Evidently, the large number of hidden units ($J = 64$) compensates for this lack; it is expected that when $J$ is smaller, DOT-TANH-based neural networks would outperform RBF-based systems, unless difference features (for example) were explicitly included in the input space.

### 4.2. Application of Neural Networks to Manually-Imputed Continuous-Valued Features

Rows 4 through 6 in Figure 3 depict the performance of neural-network STT models whose input space consists of both everywhere-defined binary-valued features $\mathbf{Q}$ and everywhere-defined continuous-valued features $\tilde{\mathbf{F}}$. The latter were produced by manually imputing the $\mathbf{F}$ features with potentially undefined values, using Equation 2. It can be seen that while the spherical GRBF does not achieve parity with the DOT-TANH activation function when applied to the concatenated feature vectors, the diagonal GRBF does. Together with the observations from the previous section, it appears that dimension-specific precision values $b_{ji}$ (cf. Equation 6) are necessary to fully exploit the information contained in continuous-valued features.

### 4.3. Application of Auto-Imputing Neural Networks to Unimputed Continuous-Valued Features

Finally, the last experiment — depicted in row 7 of Figure 3 — applied the auto-imputing JRBF activation function to the everywhere-defined binary-valued features $\mathbf{Q}$ and the continuous-valued features $\mathbf{F}$ with potentially undefined values. The latter were not previously manually imputed, as in the preceding section. Nevertheless, the achieved UI value of 0.2681 bits per 100-ms is negligibly different from the value of 0.2668 bits per 100-ms, obtained using DOT-TANH functions after manual imputation (row 4). Similarly, the achieved CI value of 0.2533 bits per 100-ms is also negligibly different from the value of 0.2534 bits per 100-ms. In numerical terms, the perturbation is $+0.5\%$ and $-0.04\%$ for the UI and CI models, respectively. This demonstrates that auto-imputing STT models achieve predictive parity with standard neural network STT models, without the need for manual imputation (which in this case relies on acausally available information, cf. Equation 2).

## 5. Discussion

### 5.1. Generalization to Unseen Data

The three sets of experiments described in the precedings sections for DEVSET, using which the number of hidden units was optimized, were repeated for the completely unseen TESTSET. The results are depicted in Figure 4.

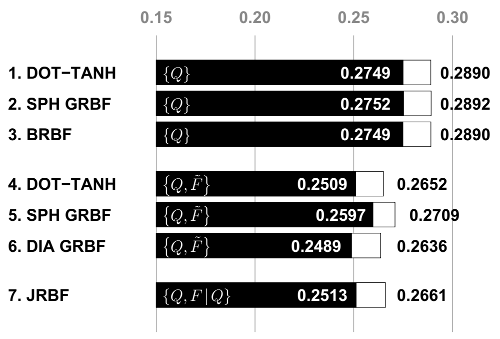

Figure 4: TESTSET cross entropy rates (in bits per 100-ms, along the horizontal axis). Symbols as for Figure 3.

As can be seen, the trends for TESTSET in Figure 4 are very similar to those for DEVSET in Figure 3. The auto-imputing models yield cross-entropy perturbations from their standard non-imputing neural network counterparts of $+0.4\%$ and $+0.2\%$, for the UI and CI variants, respectively. This suggests that the predictive parity achieved by auto-imputing neural networks can be expected for other also-unseen data sets.

### 5.2. Potential Impact

The gating implicit in the proposed JRBF activation function appears to have negligible impact on prediction entropies. This suggests that it is suitable for the large-scale analysis of the impact of any feature with missing values on speech activity prediction, and therefore on turn-taking as implied by STTs. The current work has relied on a feature which is relatively easy to manually impute — it is meaningful to state that non-speech frames have a loudness of zero, provided that care is taken to ensure that speech frames exhibit larger values. This makes loudness an appropriate feature for which to *compare* unimputed and auto-imputing behavior; for a feature such as pitch, which is arguably more interesting, the comparison cannot be made since an unimputed baseline cannot be formulated (e.g. what values of pitch should be assigned to long stretches of non-speech?). The value of the proposed approach lies in its rendering such features possible to evaluate at all. As such, it opens up STT analysis to previously untreatable features.

### 5.3. Interpretability

In contrast to the standard DOT-TANH activation function, which can arbitrarily rotate feature vectors, diagonal-covariance JRBF activation functions facilitate exploratory data analysis. The parameter values have spatial meaning relative to the geometry of the matrix $\mathbf{Q}$. Each hidden layer therefore represents a pattern, against which the actual history during every prediction is matched; the proposed RBFs are a formalization of non-linear factor analysis [16]. Hidden layer parameter values can therefore be inspected visually, with much greater ease than provided by Hinton diagrams [19], facilitating the study of prosodic cues to speakers' decisions to speak. The precision parameters ($b_{ji}$ in Equation 6) also implicitly yield a form of automatic relevance determination [20].

## 6. Conclusions

This article has explored radial basis functions (RBFs) as the hidden-unit activations in neural-network implementations of stochastic turn-taking models. It was shown that spherical Gaussian RBFs and the novel Bernoulli RBFs provide performance which is equivalent to DOT-TANH units when the input space consists of everywhere-defined binary-valued feature vectors. These RBFs were then extended to also accept continuous-valued feature vectors which may contain missing values, yielding the novel Joint RBF (JRBF) activations. Experiments revealed that for input vectors with potentially missing values, JRBFs achieve prediction cross entropies which only negligibly deviate from those achieved by standard neural networks preceded by prior manual imputation. Neural networks using JRBFs can therefore be justifiably referred to as "auto-imputing". It is reasonable to assume that they will be instrumental in the efficient, large-scale analysis of the impact on various features — including those with potentially missing values — on human conversational behavior.

## 7. Acknowledgments

This work was funded in part by the Riksbankens Jubileumsfond (RJ) project *Samtalets Prosodi*. Computing resources at Carnegie Mellon University were made accessible courtesy of Qin Jin.

## 8. References

[1] T. Wilson, J. Wiemann, and D. Zimmerman, “Models of turn-taking in conversational interaction,” *Journal of Language and Social Psychology*, vol. 3, no. 3, pp. 159–183, 1984, doi:10.1177/0261927X8400300301.

[2] E. Chapple, “The interaction chronograph: Its evolution and present application,” *Personnel*, vol. 25, pp. 295–307, 1949.

[3] J. Jaffe, S. Feldstein, and L. Cassotta, “Markovian models of dialogic time patterns,” *Nature*, vol. 216, pp. 93–94, 1967, doi:10.1038/216093a0.

[4] P. Brady, “A model for generating on-off speech patterns in two-way conversation,” *Bell Systems Technical Journal*, vol. 48, no. 9, pp. 2445–2472, 1969.

[5] K. Laskowski, J. Edlund, and M. Heldner, “A single-port non-parametric model of multi-party conversation,” in *Proc. ICASSP*, Praha, Czech Republic, May 2011, IEEE, pp. 5600–5603.

[6] K. Laskowski, M. Heldner, and J. Edlund, “Incremental learning and forgetting in stochastic turn-taking models,” in *Proc. INTERSPEECH*, Firenze, Italy, August 2011, ISCA, pp. 2069–2072.

[7] K. Laskowski and E. Shriberg, “Corpus-independent history compression for stochastic turn-taking models,” in *Proc. ICASSP*, Kyoto, Japan, March 2012, IEEE, pp. 4937–4940, doi:10.1109/ICASSP.2012.6289027.

[8] K. Laskowski, “Exploiting loudness dynamics in stochastic models of turn-taking,” in *Proc. SLT*, Miami Beach FL, USA, December 2012, IEEE, pp. 79–84, doi:10.1109/SLT.2012.6424201.

[9] S. van Buuren, *Flexible Imputation of Missing Data*, Chapman & Hall Interdisciplinary Statistics. CRC Press, Boca Raton FL, USA, 2012.

[10] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Ney York NY, USA, 1995.

[11] M. Møller, “A scaled conjugate gradient algorithm for fast supervised learning,” *Neural Networks*, vol. 6, no. 4, pp. 525–533, 1993, doi:10.1016/S0893-6080(05)80056-5.

[12] M. Buhmann, *Radial Basis Functions: Theory and Implementation*, Cambridge University Press, 2003.

[13] T. Poggio and F. Girosi, “Networks for approximation and learning,” *Proc. IEEE*, vol. 78, no. 9, pp. 1484–1487, September 1990, doi:10.1109/5.58326.

[14] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, New York NY, USA, 1999.

[15] C. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics. Springer Science + Business Media, 2006.

[16] D. Bartholemew, M. Knott, and I. Moustaki, *Latent Variable Models and Factor Analysis*, Probability and Statistics. John Wiley and Sons, 2011.

[17] J. Godfrey and E. Holliman, “Switchboard-1 release 2,” Philadelphia PA, USA, 1997, Linguistic Data Consortium, vol. LDC97S62.

[18] N. Deshmukh, A. Ganapathiraju, A. Gleeson, J. Hamaker, and J. Picone, “Resegmentation of SWITCHBOARD,” in *Proc. ICSLP*, Sydney, Australia, 1998.

[19] G. Hinton, J. McClelland, and D. Rumelhart, *Parallel Distributed Representations: Explorations in the Microstructure of Cognition*, chapter 3: Distributed representations, pp. 77–109, MIT Press, Cambridge MA, USA, 1986.

[20] Y. Qi, T. Minka, R. Picard, and Z. Ghahramani, “Predictive automatic relevance determination by expectation propagation,” in *Proc. ICML*, Banff AB, Canada, 2004, ACM, pp. 85–, doi:10.1145/1015330.1015418.