VIRTUAL AUTOENCODER BASED RECOMMENDATION SYSTEM FOR INDIVIDUALIZING HEAD-RELATED TRANSFER FUNCTIONS

Yuancheng Luo [1] *Dmitry N. Zotkin*[2] *Ramani Duraiswami*[1] *

[1] University of Maryland, Department of Computer Science, yluo1@umd.edu, ramani@umiacs.umd.edu
[2] University of Maryland, Institute for Advanced Computer Studies, dz@umiacs.umd.edu

ABSTRACT

We propose a virtual autoencoder based recommendation system for learning a user's Head-related Transfer Functions (HRTFs) without subjecting a listener to impulse response or anthropometric measurements. Autoencoder neural-networks generalize principal component analysis (PCA) and learn non-linear feature spaces that supports both out-of-sample embedding and reconstruction; this may be applied to developing a more expressive low-dimensional HRTF representation. One application is to individualize HRTFs by tuning along the autoencoder feature spaces. We demonstrate this new approach by developing a virtual (black-box) user that can localize sound from query HRTFs reconstructed from those spaces. Standard optimization methods tune the autoencoder features based on the virtual user's feedback. Experiments with CIPIC HRTFs show that the virtual user can localize along out-of-sample directions and that optimization in the autoencoder feature space improves upon initial non-individualized HRTFs. Other applications of the representation are also discussed.

Index Terms— Head-related Transfer Function, Autoencoder, Gaussian Process Regression

1. INTRODUCTION

Autoencoders are auto-associative neural networks that learn low-dimensional non-linear features which can reconstruct the original inputs [1]. This form of dimensionality reduction generalizes PCA given that trained linear-autoencoder weights form a non-orthogonal basis that capture the same total variance as leading PCs of the same dimension. Non-linear autoencoders are a form of kernel-PCA where inputs outside the training set can be embedded into the feature spaces and projected back to the original domain. Multiple autoencoders can be connected layer-wise or *stacked* to magnify expressive power and denosing autoencoder variants have also been shown to learn more representative features [2].

Low-dimensional PCA representations of HRTFs are often used as targets for regression/interpolation and personalization from predictors such as anthropometry [3, 4]. While PCA captures maximal variance along linear bases, non-linear relationships that are visible in HRTFs such as shifted spectral cues (notches/peaks) and smoothness assumptions along frequency are not represented in the versions synthesized using the linear principal components. Non-linear autoencoders provide a means of learning these properties in an unsupervised fashion, while at the same time achieving superior data compression.

*Partial support by the National Science Foundation (NSF grant IIS-1117716), the Office of Naval Research (MURI grant N00014-08-10638), and a MIPS award award from VisiSonics Corp. and MTECH.

We use the autoencoder derived feature-spaces for HRTF personalization. Most virtual auditory display (VAD) systems use non-individualized HRTFs to render 3D sounds due to the prohibitive costs of physically measuring HRTFs for each user. Non-individualized HRTFs are often per-direction sample-means of existing HRTF collections, or might just use the measurement of a single user. While this may preserve some of the common spectral features in the low frequency ranges, many subject-specific features due to anthropometric variations are lost, leading to localization errors [5]. Many works have sought individualized HRTFs by learning their relationships to subject's anthropometry [6, 3, 4]. A recent idea achieves some progress by granting listeners full-access to non-individualized HRTF PCA weights for interactive tuning [7]; while no anthropometric measurements are needed, the user must learn to tune PCA weights w.r.t. effects on the perceived sound direction.

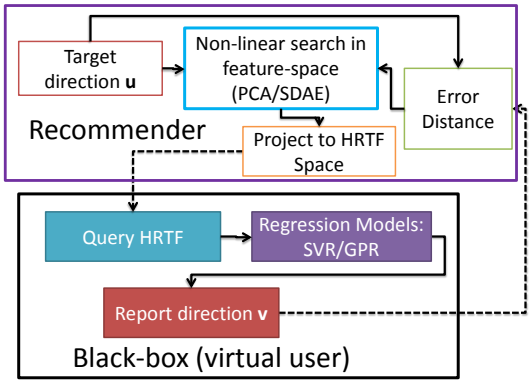

Figure 1: Recommender system based on an autoencoder learned HRTF representation for input direction **u**, query HRTF, and feedback direction **v** by a virtual user.

We use interactive tuning over feature spaces learned from a stacked denoising autoencoder (SDAE) [2] in section 2. This procedure, especially during development, would have been tedious for a real user. The method is implemented and tested on a novel virtual user concept shown in Fig. 1; a simulated listener localizes and reports the sound direction of noise samples convolved with a query HRTF. This virtual user is posed as a multiple regression problem between HRTF and direction, which we solve via Gaussian process regression (GPR) models [8] in section 3. The *best* HRTF recommendation is posed as an optimization problem between a query HRTF generated from the autoencoder feature space that minimizes an objective distance measure between a target and the virtual user's reported direction of sound in section 4. Experiments demonstrate

that our virtual user trained on CIPIC [9] HRTFs can localize directions outside a small training subset and that the recommender system improves upon initial non-individuated HRTFs along the horizontal and median plane directions.

2. AUTOENCODERS

The basic autoencoder is a three layer neural network composed of an *encoder* that transforms input layer vector $x \in \mathbb{R}^d$ via a deterministic function $f_\Theta(x)$ into a hidden layer vector $y \in \mathbb{R}^{d'}$ and a *decoder* that transforms vector y into the output layer vector $z \in \mathbb{R}^d$ via a transformation $g_{\Theta'}(y)$ [2]. The aim is to reconstruct $z \approx x$ from the lower-dimensional representation vector y where $d' < d$. The typical neural-network transformation function is given by

$$f_\Theta(x) = s\left(Wx + b\right), \quad g_{\Theta'}(y) = s\left(W'y + b'\right), \qquad (1)$$

where non-linearity is introduced via the sigmoid activation function $s(x) = \frac{1}{1+e^{-x}}$. Parameters $\Theta = \{W, b\}, \Theta' = \{W', b'\}$ are the weight matrices $W \in \mathbb{R}^{d' \times d}, W' \in \mathbb{R}^{d \times d'}$ and bias vectors $b \in \mathbb{R}^{d'}, b' \in \mathbb{R}^d$. They are trained via gradient descent of the reconstruction (mean-squared) error on the training set $X = \{x^{(1)}, \ldots, x^{(N)}\}$ w.r.t. parameters Θ and Θ'. We train an autoencoder to find a low-dimensional representation y that has mappings from input HRTF measurements[1] belonging to one or more subjects $H_{\theta,\phi} \in X$ to themselves for spherical coordinates (θ, ϕ).

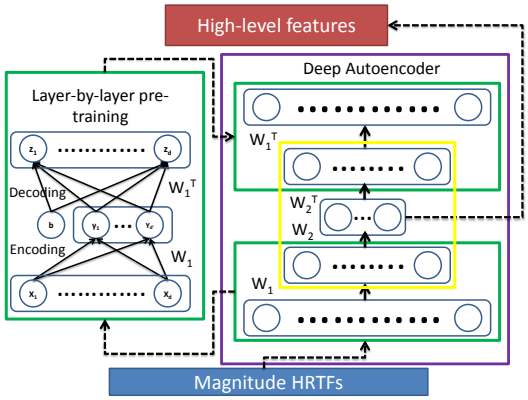

Figure 2: Two autoencoders are pre-trained and unrolled into a single deep autoencoder. Samples of non-linear high-level features can decode original HRTFs.

2.1. Stacking Autoencoders and Learned Filters

Stacked autoencoders [1] are a series of K autoencoders where the hidden layer of the k^{th} autoencoder feeds into the input of the $(k+1)^{th}$ autoencoder shown in Fig. 2. The dimensionality of the hidden layer (number of neurons) for successive autoencoders decreases to avoid learning the identity function. Layer-by-layer pre-training serializes parameter optimization for successive autoencoders where their weight matrices between input-hidden-output layers are constrained to their transposes $W' = W^T$. This has been shown to avoid getting stuck in local minimums (sub-optimal weights) that *unrolled* autoencoders (w/o pre-training) tend to fall into; unrolling the autoencoders results in a single *deep* autoencoder of $2K + 1$ layers. Further training of the deep autoencoder usually results in smaller reconstruction errors.

To visualize the types of features that autoencoders produce from HRTF measurements, one can examine the rows of weight matrix W that are fed into corresponding neurons in vector y. Weights between the input and first hidden-layer in Fig. 3 have interpretable filters; peak and notch shaped filters along the HRTF spectrum compete to activate or deactivate a neuron. Dirac-shaped filters correspond to the identity mapping along a single frequency.

2.2. Denosing Autoencoders and Dimensionality Reduction

The denosing autoencoder [2] is a variant of the basic autoencoder that reconstructs the original inputs from a corrupted version. A common stochastic corruption is to randomly zero-out elements in training data X. This forces the autoencoder to learn hidden representation vectors y that are stable under large perturbations of inputs x, which implicitly encodes a smoothness assumption w.r.t. frequency in the case of HRTF measurement inputs; reconstructed outputs z are therefore smooth curves. This property is useful for HRTF dimensionality reduction where some of the variance due to noise can be ignored to yield better reconstruction errors in Fig. 3.

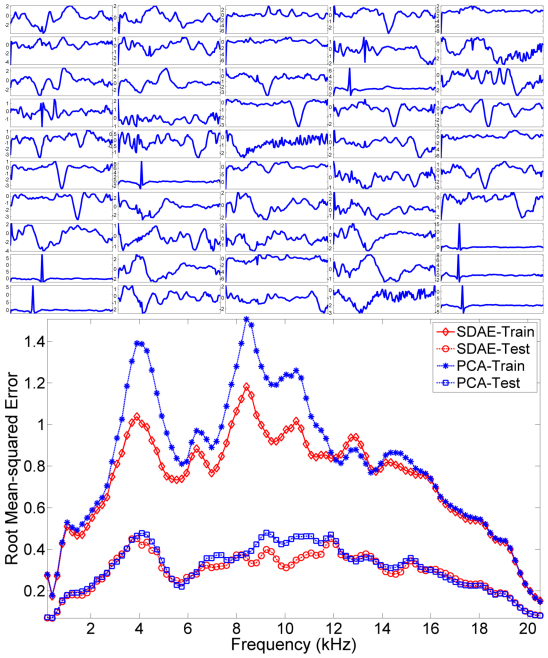

Figure 3: Layer 1 filters of SDAE-$\{100, 50, 25, 2\}$ (inputs-per-autoencoder) are learned from $(30/35)$ CIPIC subjects HRTFs (top). Autoencoder features and PCA weights (2D) reconstruct training and out-of-sample HRTF measurements (bot).

3. VIRTUAL BLACK-BOX USER

The hypothetical virtual user must be able to localize sound as a real user would with binaural sound presented over head-phones.

[1] HRTF measurements are preprocessed by taking the magnitude of the discrete Fourier transform, truncating to $100/200$ bins, and scaling the magnitude range to $(0, 1)$ (1 is maximum magnitude for all HRTFs). We use the term *HRTF measurement* to refer exclusively to HRTF magnitude.

In practice, real users are given snippets of white noise convolved with separate HRTFs along left and right channels. For simplicity, the recommender directly supplies the virtual user with a query magnitude HRTF H_* for sound localization along a single ear; the power spectrum of white noise is flat and we do not wish to disentangle binaural results. Installing a virtual user in place of a real user also automates sound tests for future recommender query strategies.

Thus, we model the virtual user as a regression problem from HRTF measurements in Fig. 1 as predictors of spherical direction in the form of a unit vector $v \in \mathbb{R}^3$. This non-linear multiple regression problem has the probabilistic interpretation $P(v|H_*)$, which can be accurately modeled by multiple Gaussian processes (GPs) with automatic hyperparameter training. Other non-linear regression models such as support vector regression (SVR) [10] do not have probabilistic interpretations and flexible parameter training.

3.1. Gaussian Process Regression

In the virtual user multiple regression problem, we independently train 3 GPs that predict the Cartesian direction cosines $y = v_i$ from d-dimensional predictor variables $x = H_{\theta,\phi} \in \mathbb{R}^d$ given by HRTF measurements of the virtual user. In this Bayesian nonparametric approach to regression, it is assumed that the observation y is generated from an unknown latent function $f(x)$ and is corrupted by additive (Gaussian) noise

$$y = f(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2), \tag{2}$$

where the noise term ϵ is zero-centered with constant variance σ^2. Placing a GP prior distribution on the latent function $f(x)$ enables inference and enforces several useful priors such as local smoothness, stationarity, and periodicity. For any subset of inputs $X = [x_1, \ldots, x_N]$, the corresponding vector of function values $\mathbf{f} = [f(x_1), f(x_2), \ldots, f(x_N)]$ has a joint N-dimensional Gaussian distribution that is specified by the prior mean $m(x)$ and covariance $K(x_i, x_j)$ functions given by

$$\begin{aligned} f(x) \sim \text{GP}\left(m(x), K(x_i, x_j)\right), \quad m(x) = 0, \\ K(x_i, x_j) = \textbf{cov}\left(f(x_i), f(x_j)\right). \end{aligned} \tag{3}$$

For N training outputs y and N_* test outputs f_*, we define the Gram matrix $\hat{K} = K_{ff} + \sigma^2 I$ as the pair-wise covariance evaluations between training and test predictors given by matrices $K_{ff} = K(X, X) \in \mathbb{R}^{N \times N}$, $K_{f*} = K(X, X_*) \in \mathbb{R}^{N \times N_*}$, and $K_{**} = K(X_*, X_*) \in \mathbb{R}^{N_* \times N_*}$. GP inference is a marginalization over the function space $\mathbf{f}$, which expresses the set of test outputs conditioned on the test inputs, training data, and training inputs as a normal distribution $P(f_*|X, y, X_*) \sim \mathcal{N}(\bar{f}_*, \textbf{cov}\left(f_*\right))$ given by

$$\begin{aligned} \bar{f}_* = E[f_*|X, y, X_*] = K_{f*}^T \hat{K}^{-1} y, \\ \textbf{cov}\left(f_*\right) = K_{**} - K_{f*}^T \hat{K}^{-1} K_{f*}. \end{aligned} \tag{4}$$

For simplicity, the virtual user reports only the predicted mean $\bar{f}_*$ from inputs X_* in Eq. 4 as the predicted direction and ignores the predicted variance which measures confidence. Model-selection is an O(N^3) runtime task of minimizing the gradient of the negative log-marginal likelihood function w.r.t. hyperparameters Θ_i:

$$\begin{aligned} \log p(y|X) &= -\frac{1}{2}\left(\log|\hat{K}| + y^T \hat{K}^{-1} y + N \log(2\pi)\right), \\ \frac{\partial \log p(y|X)}{\partial \Theta_i} &= -\frac{1}{2}\left(\text{tr}\left(\hat{K}^{-1} P\right) - y^T \hat{K}^{-1} P \hat{K}^{-1} y\right), \end{aligned} \tag{5}$$

where $P = \partial \hat{K} / \partial \Theta_i$ is the matrix of partial derivatives.

For the choice of covariance, we consider the product of stationary Matérn ($\nu = 3/2$) functions for each of the d independent variables $r_{ijk} = |x_{ik} - x_{jk}|$ given by

$$K(x_i, x_j) = \prod_{k=1}^{d} \left(1 + \frac{\sqrt{3} r_{ijk}}{\ell_k}\right) e^{-\frac{\sqrt{3} r_{ijk}}{\ell_k}}, \tag{6}$$

where ℓ_k is the characteristic length-scale hyperparameter for the k^{th} predictor variable. This covariance function outperforms other Matérn classes $\nu = \{1/2, 5/2, \infty\}$ in terms of data marginal-likelihood and prediction error in experiments.

3.2. Experimental Localization

To evaluate the virtual user's localization of sound directions outside the database, we specify its GPs over a random subset of available HRTF-direction pairs (1250/3) belonging to CIPIC subject 154's right ear and jointly train all hyperparameters and noise term σ for 50 iterations via gradient descent of the log-marginal likelihood in Eq. 5. The prediction error is the cosine distance metric between predicted direction v and test direction u given by

$$\text{dist}\left(\mathbf{u}, \mathbf{v}\right) = 1 - \frac{< \mathbf{u}, \mathbf{v} >}{||\mathbf{u}||||\mathbf{v}||}, \quad \mathbf{u}, \mathbf{v} \in \mathbb{R}^3. \tag{7}$$

The mean prediction errors of GPs trained on random and full measurements are shown in Table. 1. Results indicate better localization near the ipsilateral right-ear directions than in the contralateral direction where clusterings are seen in Fig. 4. Compared to nu-SVR [10] with radial basis function kernel and tuned parameter options (-s 4 -t 2 -g 7 -c 5), GPR is more accurate because of more expressive parameters and automatic model-selection.

SVR-Random	GPR-Random	SVR-Full	GPR-Full
8.19°	**5.37°**	1.34°	**0.44°**

Table 1: Mean cosine distances Eq. 7 (in degrees) between predicted and test directions for trained models (random and full data).

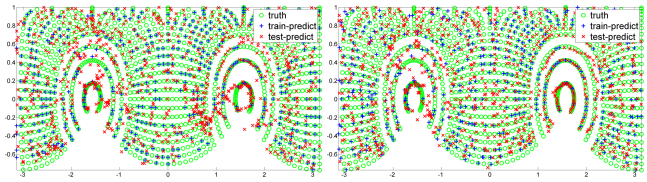

Figure 4: Virtual user predicted directions; plots are Mercator sphere-to-cylinder projections of spherical coordinates θ, ϕ. SVR (left) and GPR (right) models are trained on identical randomized subset of HRTF-direction pairs and predicted over test (remaining) pairs. SVR test predictions exhibit clustering behavior that indicate poor generalization. GPR performs better overall than SVR; it accurately localizes ipsilateral (direct-ear) directions better than contralateral (head-shadow) directions.

4. AUTOMATIC RECOMMENDER

The immediate goal of the automatic recommender is to submit a HRTF for a target direction that matches the virtual user's reported

direction. In practice, we wish to maximize localization accuracy and minimize the number of probes. It is clear that sampling from the full HRTF space is too costly so the recommender narrows the search space to lower dimensional PCA weight and SDAEs feature spaces learned from the training HRTF measurement set ($30/35$ CIPIC subjects, right-ears). The PCs span the log-scaled HRTF magnitude space where the both positive and negative weights reconstruct positive magnitude HRTFs after taking the exponential. We train[2] the SDAE using the Deep Learning Toolbox [11] with the layer structure $\{100, 100, 50, 25, 12\}$ (inputs-per-autoencoder). Decoding from the highest-level feature space gives non-negative magnitude HRTFs due to the sigmoid activation function in Eq. 1.

The search task is posed as an unconstrained optimization problem. The recommender uses the quasi-newton method with cubic line-search and initial guesses at non-individualized (training sample-directional-means) HRTFs projected or encoded into the low-dimensional search space for PCA and SDAE respectively; probes to the virtual user are HRTF measurements projected or decoded from the search space. The objective function is the Euclidean distance between target and reported directions u and v respectively; the cosine distance is not used because the magnitude of the reported direction v may contain information about spurious localization phenomenons such as *within-head*.

The accuracy of the learned HRTFs for both horizontal and median plane directions are compared to the virtual user's actual HRTFs via the signal-to-distortion ratio (SDR) given by

$$\text{SDR}_{\theta,\phi} = 10\log_{10}\frac{\sum_{k=1}^{d} H_{\theta,\phi,k}^2}{\sum_{k=1}^{d}(H_{\theta,\phi,k} - \hat{H}_{\theta,\phi,k})^2}, \qquad (8)$$

where $\hat{H}_{\theta,\phi,k}$ is the recommender's HRTF probe to the virtual user. The recommender that searches the SDAE search space achieves better localization than that of PCA along front-back and ipsilateral directions to the listening ear along the horizontal plane in Fig. 5. For the median plane[3], SDAE space localizes better along directions closer to the horizontal plane. Listening tests by the authors' showed no perceptual differences when either original or learned versions of CIPIC subject 154 were used in a VAD.

5. CONCLUSIONS

This paper presents a first application of deep-learning autoencoders to HRTFs. We developed low-dimensional feature representations that are applied to novel techniques for obtaining a user's HRTFs using listening tests; a front-to-end system for simulating sound localization by a virtual user and a recommender system is built. Experimental results quantified both aspects of localizing sound in unknown directions and personalizing non-individualized HRTFs from latent feature spaces. This is a first step towards the larger goal of learning a parsimonious representation of the listener's HRTFs with minimum probes. Future work will pursue developments along bounded derivative-free global optimization in feature space, autoencoder variants, and real listening tests.

6. REFERENCES

[1] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[2]Pre-training iterations 30, fine-tune iterations 15, batch-size 1, pre-train learning rate 1, fine-tune rate .5, momentum 0

[3]Positive θ refers to directions in front of listener

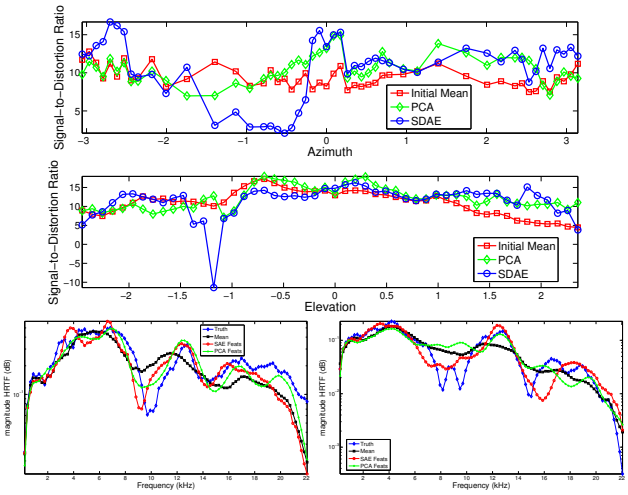

Figure 5: SDRs for initial mean HRTF guesses and localized HRTFs after searching PCA and SDAE spaces for targeted directions along horizontal and median planes (top). Sample HRTFs along horizontal plane, $\phi = 80°$ (left). Median plane HRTF $\theta = 84°$ (right).

[2] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning*, vol. 11, pp. 3371–3408, Dec. 2010.

[3] Q. Huang and Y. Fang, "Modeling personalized head-related impulse response using support vector regression," *J Shanghai Univ (Engl Ed)*, vol. 13, no. 6, pp. 428–432, 2009.

[4] H. Hu, L. Zhou, H. Ma, and Z. Wu, "HRTF personalization based on artificial neural network in individual virtual auditory space," *Applied Acoustics*, vol. 69, no. 2, pp. 163–172, 2008.

[5] E. M. Wenzel, M. Arruda, D. J. Kistler, and F. L. Wightman, "Localization using nonindividualized head-related transfer functions," *JASA*, vol. 94, p. 111, 1993.

[6] D. Zotkin, J. Hwang, R. Duraiswaini, and L. S. Davis, "HRTF personalization using anthropometric measurements," in *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.* Ieee, 2003, pp. 157–160.

[7] K. Fink and L. Ray, "Tuning principal component weights to individualize HRTFs," in *ICASSP*, 2012.

[8] C. E. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. Cambridge, Massachusettes: MIT Press, 2006.

[9] V. R. Algazi, R. O. Duda, and C. Avendano, "The CIPIC HRTF Database," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 2001, pp. 99–102.

[10] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[11] R. B. Palm, "Prediction as a candidate for learning deep hierarchical models of data," Master's thesis, Technical University of Denmark, DTU Informatics, 2012.