Approximate Intrinsic Fourier Analysis of Speech

Frank Tompkins and Patrick J. Wolfe

Statistics and Information Sciences Laboratory
Harvard School of Engineering and Applied Sciences
33 Oxford Street, Cambridge, MA 02138

tompkin@fas.harvard.edu, patrick@seas.harvard.edu

10.21437/Interspeech.2009-28

Abstract

Popular parametric models of speech sounds such as the source-filter model provide a fixed means of describing the variability inherent in speech waveform data. However, nonlinear dimensionality reduction techniques such as the intrinsic Fourier analysis method of Jansen and Niyogi provide a more flexible means of adaptively estimating such structure directly from data. Here we employ this approach to learn a low-dimensional manifold whose geometry is meant to reflect the structure implied by the human speech production system. We derive a novel algorithm to efficiently learn this manifold for the case of many training examples—the setting of both greatest practical interest and computational difficulty. We then demonstrate the utility of our method by way of a proof-of-concept phoneme identification system that operates effectively in the intrinsic Fourier domain.

Index Terms: speech analysis, unsupervised learning, phoneme identification, Nyström method

1. Introduction

The well-known source-filter model treats the speech production system as a convolution of a glottal waveform and a filter response that models the resonances of the vocal tract [1]. This model is modulated by several parameters such as formant frequency, formant bandwidth, and the period of the glottal airflow velocity. Consider a typical windowed segment of speech containing, say, 256 samples. The vector of samples lies in a 256-dimensional space, even though the source-filter model has fewer free parameters. Thus data variability in the ambient 256-dimensional space can be described by variations in a lower dimensional 'intrinsic' space, at least to the extent that the source-filter model accurately describes speech sounds. We then expect the space of speech sounds to roughly exhibit a manifold structure. In other words, the cloud of data points formed by, say, the spectra of windowed speech signals might be close to a low-dimensional submanifold of the ambient space of all possible windowed spectra. This insight has been made previously and boils down to the idea that operating in an intrinsic space can be quite useful; learning a mapping from the ambient space of speech sounds to an intrinsic space provides a sparse representation of the speech information present in the signal. We expect the idea of intrinsic representation of speech might improve the performance of any number of speech algorithms.

How then does one learn an intrinsic space of speech sounds—a speech manifold? The signal processing community has developed myriad manifold learning/dimensionality reduction algorithms in recent years, such as LLE, Hessian eigenmaps, and Laplacian eigenmaps (see, e.g., [2]). Recently, Jansen and Niyogi [3, 4] have described a modified version of Laplacian Eigenmaps, which they dub 'intrinsic Fourier analysis,' that learns an intrinsic speech representation in an unsupervised setting. The authors also develop a mathematical argument supporting the manifold structure of speech to the extent that the vocal tract is well modeled by concatenated tubes. However, they present neither a computational analysis of the algorithm nor a detailed study of its practical utility (though they have carried out extensive testing in the semi-supervised setting). Fully unsupervised learning is of great interest not only because labeling speech signals manually is labor-intensive and prone to inconsistencies among linguists, but also on account of the overwhelming abundance of unlabeled, raw speech data.

The technique of [3] begins with a set of N data points $x_i \in R^H$, where each point is a representation of a windowed segment of speech in some domain, such as the magnitude of the Fourier spectrum, the cepstrum, or LPC coefficients. The implications of choosing a particular domain over the others have not been studied, and are a topic of future research (e.g., can we account for pitch variability by choosing the LPC domain over the Fourier domain?). The goal is to use the points x_i to learn a mapping f that takes a novel point $x \in R^H$ to an intrinsic representation on the speech manifold. Then variations in speech ought to be differentiated by fewer components in this intrinsic manifold basis as compared to the ambient basis of R^H.

To compute f we must solve a generalized eigenvalue problem (GEP), as we will see in Section 2. It turns out that directly solving this GEP using a straightforward approach such as the QZ algorithm can be prohibitively slow. In our experiments, even moderately large data sets (500 to 1000 training samples) presented convergence issues for MATLAB, which often simply gave up on finding a solution. Thus a more robust algorithm is needed if one wishes to study the implications of intrinsic Fourier analysis for speech in detail.

The remainder of this article is organized as follows. We begin with a concise statement of the intrinsic Fourier analysis technique. Then we derive the mathematics behind our algorithm. This is followed by a discussion of computational details involving the Nyström approximation method and its implementation. We then conclude with a presentation of proof-of-concept experimental results and a discussion of their implications for the approximate intrinsic Fourier analysis of speech waveform data.

2. Intrinsic Fourier analysis

Let X be the H by N matrix whose ith column is x_i. We assume $H << N$, since a typical application in speech may easily involve thousands of training samples each having on the order of tens or hundreds of dimensions. Let $K(x, y)$ be a positive semi-definite kernel function. For simplicity we choose the inner product kernel $K(x,y) = \langle x, y\rangle$. Let the kernel matrix $K = X^{\mathrm{T}}X$ be the set of all inner products between the inputs, so that $K_{ij} = x_i^{\mathrm{T}}x_j$. This is an N by N symmetric positive semi-definite matrix, but being a Gram matrix it is of rank at most $H < N$ and thus singular.

As is typical in manifold learning algorithms, one constructs an undirected graph from the data points to serve as a proxy for the manifold. This is generally done via a k nearest neighbors search or ϵ balls. For this graph we define the symmetric adjacency matrix W such that W_{ij} is 1 if x_i is a neighbor of x_j (or vice versa) and 0 otherwise. We then define the graph Laplacian matrix $L = D - W$, where D is symmetric with $D_{ii} = \sum_{j=1}^{N} W_{ij}$. In practice, it is common to adopt the normalized Laplacian, given by $D^{-1/2}LD^{-1/2}$, in place of the Laplacian, as it has some nice theoretical properties [5] that are beyond the scope of this article.

We seek a projection f to an intrinsic basis on the manifold. Once this mapping has been learned from the speech data x_i, a novel data point x can be expressed in the intrinsic basis as $f(x)$. To this end, let H_K denote the reproducing kernel Hilbert space for K. Then one such f is given by the solution to the regularized optimization problem

$$f = \arg\min_{\phi\in H_K} \xi\|\phi\|_K^2 + \sum_{ij} \phi(x_i)L_{ij}\phi(x_j). \qquad (1)$$

The jth component of the solution can be expressed as

$$f_j(x) = \sum_{i=1}^{N} \alpha_{ij}K(x_i, x) = x^{\mathrm{T}} \sum_{i=1}^{N} \alpha_{ij}x_i, \qquad (2)$$

where α_{ij} is the ith component of the eigenvector corresponding to the jth smallest eigenvalue of the GEP

$$(\xi I + LK)\alpha = \lambda K\alpha. \qquad (3)$$

Ordering the eigenvalues leads to a natural ordering of smoothness in the intrinsic spectral components f_j. Note that f is not invariant with respect to arbitrary scaling of the eigenvectors; thus we make the standard assumption that $\|\alpha\| = 1$. To emphasize the fact that choosing a linear kernel results in a linear function f, we can further express f as a matrix equation $f(x) = Bx$, where the jth row of B is given by

$$B_{j:} = \sum_{i=1}^{N} \alpha_{ij}x_i^{\mathrm{T}}. \qquad (4)$$

3. Approximation algorithm

We are now ready to state our central algorithmic observation: since K has rank at most $H < N$, at most H eigenvalues λ will be finite. This occurs because the $N - \mathrm{rank}(K)$ eigenvectors corresponding to $\lambda = \infty$ lie in the null space of K. Thus, even though (3) contains N by N matrices and potentially has N solutions, we know ahead of time that we seek only H solutions. This begs the question: can we solve a smaller problem instead? Indeed, we now present such a method for efficiently computing exact or approximate solutions to (3).

We point out that many contemporary algorithms for efficiently solving large GEPs require one or more of symmetric matrices, positive definite matrices, and structured matrices, though in [6] the authors attempt to develop a more general algorithmic framework. However, we will presently utilize the special structure of (3) to formulate a complete algorithmic solution. These ideas may apply to other GEPs with similar low rank structure.

3.1. Derivation

First we introduce a 'diagonalized' GEP. Let $K = UDU^{\mathrm{T}}$ be the eigendecomposition of the kernel matrix. Letting $\alpha_K = U^{\mathrm{T}}\alpha$ be the expression of the eigenvector α in the eigenbasis of K, (3) can be expressed as

$$(\xi I + U^{\mathrm{T}}LUD)\alpha_K = \lambda D\alpha_K. \qquad (5)$$

Let N be the null space of K (distinction between this N and the number of data points as defined previously should be clear from context). By expressing an arbitrary eigenvector as $\alpha_K = \alpha_{N^\perp} + \alpha_N$, where $\alpha_N \in N$ and $\alpha_{N^\perp} \in N^\perp$, we show that the large GEP in (3) can be solved exactly by first solving a much smaller eigenvalue problem in $N^\perp$ followed by a simple algebraic extension to N. Plugging the expression for α_K into (5) we find

$$\xi\alpha_N + (\xi I + U^{\mathrm{T}}LUD)\alpha_{N^\perp} = \lambda D\alpha_{N^\perp}. \qquad (6)$$

Now let the eigenbasis of K be ordered such that the first H eigenvectors span $N^\perp$. Let $a \in R^H$ be the vector of nonzero elements of $\alpha_{N^\perp}$ and $b \in R^{N-H}$ be the vector of nonzero elements of α_N. Thus, a is the first H elements of $\alpha_{N^\perp}$ (the remaining elements must all be zero by definition) and b is the last $N - H$ elements of α_N. Define $M = (\xi I + U^{\mathrm{T}}LUD)$ and denote the matrix composed of the ith through jth rows and mth through nth columns of a matrix A by $A_{[i,j],[m,n]}$. Plugging a and b into (6) we find two simultaneous matrix equations. First,

$$M_{[1,H],[1,H]}a = \lambda D_{[1,H],[1,H]}a, \qquad (7)$$

from the first H components of (6), and second, from the last $N - H$ components,

$$\xi b + M_{[H+1,N],[1,H]}a = 0. \qquad (8)$$

Thus, we can first solve an H by H standard eigenvalue problem (note D is diagonal) to find a. Then we compute the b corresponding to each eigenvector a as a product

$$b = -\frac{1}{\xi}M_{[H+1,N],[1,H]}a. \qquad (9)$$

Zero-padding a and b appropriately to get $\alpha_{N^\perp}$ and α_N respectively, we then have the solutions to the original problem of (3) as

$$\alpha = \frac{U(\alpha_{N^\perp} + \alpha_N)}{\|U(\alpha_{N^\perp} + \alpha_N)\|}. \qquad (10)$$

We premultiply by $U = U^{-\mathrm{T}}$ to transform α_K back to the original basis of R^N. Normalization ensures the constraint $\|\alpha\| = 1$ of (3) is satisfied.

This procedure requires spectral decomposition of K. While this computation is likely less expensive than solving (3) directly, we can appeal to kernel approximation methods to compute U and D more quickly.

3.2. Nyström approximation

The Nyström method can be used to compute an approximation of the eigendecomposition of $K = X^{\mathrm{T}}X$ by sampling the columns of X appropriately [7]. We suppose the kernel matrix is decomposed according to Nyström as $\tilde{K} = \tilde{U}\tilde{D}\tilde{U}^{\mathrm{T}}$ with $\tilde{D}$ diagonal and the columns of $\tilde{U}$ of unit norm. Since K has rank at most H, we may choose $\tilde{D}$ to be of size H. In this case, the columns of $\tilde{U}$ span $N^\perp$ and the approximation error $\|K - \tilde{K}\|$ is essentially zero. Choosing $\tilde{D}$ to be smaller results in a larger approximation error. The power of the Nyström method to find this 'sparse representation' of K can be leveraged to solve (7) without computing the entire spectrum of K.

3.3. Implementation

Since the only goal of Nyström is the minimization of $\|K-\tilde{K}\|$, the output $\tilde{U}$ generally does not satisfy $\tilde{U}^{\mathrm{T}}\tilde{U}=I$. So in using Nyström to solve (7), we must orthogonalize $\tilde{U}$ as the derivation of (5) hinges on the orthogonality of U. Once we have orthogonalized $\tilde{U}$ to obtain, say, $\tilde{U}_o$, we then maintain $K=\tilde{U}_o\tilde{D}_o\tilde{U}_o^{\mathrm{T}}$ by defining

$$\tilde{D}_o=\tilde{U}_o^{-1}\tilde{U}\tilde{D}(\tilde{U}_o^{-1}\tilde{U})^{\mathrm{T}} \qquad (11)$$

where $()^{-1}$ denotes the pseudoinverse. This decomposition can be made approximate by sampling fewer than H columns from X in computing the Nyström approximation. There exist many suitable orthogonalization algorithms, such as the Gram-Schmidt process, QR factorization, and projection onto the Stiefel manifold [8] via $\tilde{U}(\tilde{U}^{\mathrm{T}}\tilde{U})^{-1/2}$.

Thus computation of $D_{[1,H],[1,H]}=\tilde{D}_o$ and $U_{[1,N],[1,H]}=\tilde{U}_o$ can be realized using the Nyström technique. Now, (7) becomes a GEP, albeit a much smaller one than (3), since $\tilde{D}_o$ is generally not diagonal. We point out that, rather than using Nyström, one could apply any number of iterative algorithms to find the smallest H finite eigenvalues and form a diagonal $\tilde{D}$ and orthogonal $\tilde{U}$. We test both possibilities in the experimental section.

In any case, the full D matrix is just $D_{[1,H],[1,H]}$ padded with zeros (as K has rank H) and the columns of $U_{[1,N],[H+1,N]}$ are a basis for the null space of K such that $U^{\mathrm{T}}U=I$. Such a basis can be found by solving $Kx=0$. Alternatively, we can compute this basis with high probability by simply initializing $U_{[H+1,N],[H+1,N]}=I$ followed by orthogonalization of U such that the last $N-H$ columns span the null space of K in order to maintain the ordering assumption used to derive (7) and (9). One could use the Gram-Schmidt process or QR factorization, but Stiefel projection fails because it mixes up the ordering. This ad hoc technique works because any column of $U_{[1,N],[1,H]}$ has very low probability of having all zeros in the first H components, and thus the initial U constructed with $U_{[H+1,N],[H+1,N]}=I$ is full rank with high probability. In empirical studies of speech signals we have encountered no case where this approach fails.

Having computed D and U, we construct M, solve (7), and compute (9) and (10) for each eigenvector. These are combined with the training data x_i to compute B in (4). The intrinsic spectrum of a new data point x can now be computed as Bx.

4. Experimental results

First, we consider the algorithm itself—the accuracy of the approximate spectra it produces and its execution time. We then quantitatively demonstrate the utility of working in the intrinsic Fourier domain, and the extent to which it compresses speech information, by building a rudimentary phoneme identification system and studying its performance as the number of intrinsic components used varies.

4.1. Accuracy and efficiency

To generate the following two plots, we drew samples from the TIMIT training set consisting of the magnitude of the 256-point Fourier transform ($H=129$) of the central windowed segment of each phoneme in order to learn the speech manifold. We set $\xi=1$, used the k nearest neighbors approach with k equal to half the number of training examples of each phoneme, and used the normalized Laplacian as described in Section 2. We selected QR factorization to find a basis for N according to the method described in Section 3.3. When using Nyström to solve (7) we used Stiefel projection to compute $\tilde{U}_o$. These were found empirically to be the fastest choices for orthogonalization in MATLAB.

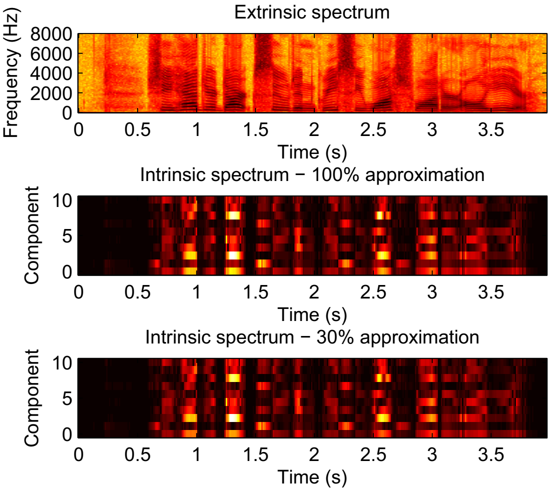

Figure 1: *Comparison of extrinsic, intrinsic, and approximate intrinsic spectra. Note the distinction between vowels and fricatives evident from the first two intrinsic manifold components.*

In Fig. 1 we compare extrinsic, exact intrinsic, and approximate intrinsic spectra of the sentence 'sa1' in TIMIT, which reads, "she had your dark suit in greasy wash water all year." These plots show that even a significantly approximate solution of (3) yields qualitatively similar spectra to an exact solution. One also notices the ability of the intrinsic representation to distinguish phonemes, such as vowels versus fricatives. We employed Nyström with H approximants in the center plot and $.3H$ approximants in the rightmost plot to learn the manifold (that is, B). The intrinsic spectra were computed by windowing the sentence with a Hamming window with 50% overlap and computing Bx for each window's Fourier magnitude spectrum x. The training data x_i were identical in both cases—30 examples of each phoneme from the TIMIT training set.

In Fig. 2 we plot execution time for manifold learning as a function of training set size for

1. directly solving (3) using MATLAB's built-in EIG() function, which uses the QZ algorithm (specifically the DGGEV subroutine in LAPACK),
2. solving (7) using MATLAB's EIGS() function to find only the H finite eigenvalues,
3. solving (7) using Nyström with H approximants, and
4. solving (7) using Nyström with $.3H$ approximants.

The training sets, drawn across all phonemes in TIMIT, were constructed such that the smaller sets are contained in the larger ones. Note that the full QZ algorithm's run time increases more rapidly than the others. That curve is incomplete because MATLAB's EIG() function was unable to solve the GEP. These time tests were performed in MATLAB R2007a on an Intel Core 2 Duo E7200 machine with 2GB of RAM.

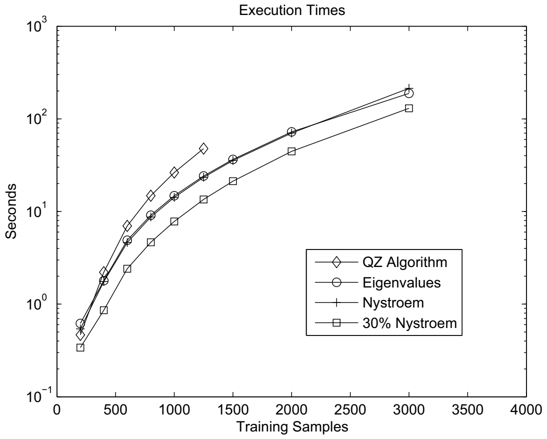

Figure 2: *Log plot of execution times.*

4.2. Phoneme identification

We created a feed-forward neural network to perform rudimentary phoneme identification on three vowel phonemes, *iy*, *ao*, and *ae*, and computed identification error as a function of the number of intrinsic components used. The neural network input is an individual windowed speech segment in the intrinsic domain and the output is a vector of length 3. In training, we set the corresponding output node to 1 for the correct phoneme label and fixed the other 2 nodes to 0. In testing, we took the phoneme label corresponding to the largest of the 3 output values. To learn the speech manifold exactly, we selected 40 examples of each phoneme in TIMIT, so this falls into the large training set regime that is the focus of this article.

In Table 1 we show average classification errors for a neural network with one hidden layer of 10 nodes implemented with MATLAB's Neural Network Toolbox. The ambient space in which the manifold training points x_i lie was taken to be the set of 11ary LPC coefficients (with ten degrees of freedom, as the first coefficient, which we ignore, is fixed at unity). We chose this space in an effort to limit the effect of speaker pitch variation on the manifold structure ab initio. Each experiment consisted of training the neural network via Levenberg-Marquardt back-propagation on intrinsic LPC spectra of 200 examples of each of the three vowel phonemes selected from male speakers in the TIMIT training set, and finally generating phoneme labels for female speakers in the TIMIT test set with 200 examples per phoneme as well. The pitch variation between genders provides a sort of worst case scenario. The error for each experiment was computed as the percentage of incorrect phoneme labels produced. We performed 20 experiments per number of intrinsic components, and the means and standard deviations of the errors were tabulated.

These data show quantitatively that the intrinsic representation indeed compresses important speech information into just a few components. Using 10 components as a baseline, even discarding half the components barely alters the error; thus the higher order components are not useful in differentiating these phonemes. Note that we are interested in how the error changes as a function of intrinsic components used, not the absolute error rate. Any improvement such as imposing an inter-frame correlation metric or simply using more training examples should yield lower overall error rates. Since it is not the primary goal of this article to advance the state of the art in phoneme identification, the method employed here is very simplistic.

5. Discussion

In this article we focused on deriving and testing an algorithm for efficiently solving the regularization GEP of intrinsic Fourier analysis exactly or approximately, for the case of many training samples. We presented results that demonstrate improved computational efficiency, and developed a neural network that learns a speech-to-phoneme mapping via the intrinsic spectrum, illustrating that this unsupervised manifold learning approach compresses distinguishing speech characteristics into a useful sparse representation.

Much work remains to be done in studying how the intrinsic representation of speech might improve the performance of existing speech algorithms or lead to entirely new approaches. We have taken a step in this direction—facilitating future research by improving computational efficiency. Furthermore, we point out that while one can formally argue that the system comprised of the vocal tract and glottal source waveform possesses a manifold structure, such arguments cannot apply to the entire gamut of speech sounds (specifically including stochastic and turbulent phenomena). Our current research aims at studying how the intrinsic setting can be leveraged to study the many variations present in speech data.

Table 1: *Phoneme identification error for three vowels.*

Intrinsic Components	Mean Error
10	14.3% ± 1.8%
7	13.2% ± 3.8%
4	17.9% ± 1.9%
3	23.4% ± 1.0%
2	27.9% ± 0.8%

6. Acknowledgements

This material is based in part upon work supported by the Defense Advanced Research Projects Agency under Grant HR0011-07-1-0007, and by the National Science Foundation under Grants DMS-0631636 and DMS-0652743.

7. References

[1] T. F. Quatieri, *Discrete-Time Speech Signal Processing*. Prentice Hall, 2002.

[2] D. L. Donoho and C. Grimes, "Hessian Eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proc. Natl. Acad. Sci. USA*, vol. 100, pp. 5591–5596, 2003.

[3] A. Jansen and P. Niyogi, "Intrinsic Fourier analysis on the manifold of speech sounds," in *Proc. IEEE Internat. Conf. Acoust. Speech Signal Process.*, 2006, pp. 241–244.

[4] ——, "A geometric perspective on speech sounds," University of Chicago, Tech. Rep. TR-2005-08, 2005.

[5] U. von Luxburg, M. Belkin, and O. Bousquet, "Consistency of spectral clustering," *Ann. Statist.*, vol. 36, pp. 555–586, 2008.

[6] J. G. L. Booten, H. A. van der Vorst, M. P. M., and H. J. J. te Riele, "A preconditioned Jacobi-Davidson method for solving large generalized eigenvalue problems," CWI, Tech. Rep. Dept. Num. Math., 1994.

[7] M.-A. Belabbas and P. J. Wolfe, "Spectral methods in machine learning and new strategies for very large data sets," *Proc. Natl. Acad. Sci. USA*, vol. 106, pp. 369–374, 2009.

[8] Y. Chikuse, *Statistics on Special Manifolds*. Springer-Verlag, 2003.