

EXPLORATORY ANALYSIS AND VISUALIZATION OF SPEECH AND MUSIC BY LOCALLY LINEAR EMBEDDING

Viren Jain and Lawrence K. Saul

Department of Computer and Information Science
University of Pennsylvania, Philadelphia, PA 19104
{viren,lsaul}@seas.upenn.edu

ABSTRACT

Many problems in voice recognition and audio processing involve feature extraction from raw waveforms. The goal of feature extraction is to reduce the dimensionality of the audio signal while preserving the informative signatures that, for example, distinguish different phonemes in speech or identify particular instruments in music. If the acoustic variability of a data set is described by a small number of continuous features, then we can imagine the data as lying on a low dimensional manifold in the high dimensional space of all possible waveforms. Locally linear embedding (LLE) is an unsupervised learning algorithm for feature extraction in this setting. In this paper, we present results from the exploratory analysis and visualization of speech and music by LLE.

1. INTRODUCTION

Many systems for pattern recognition depend on a preprocessing front end for dimensionality reduction. The goal of this front end is to provide a compact representation of high dimensional data that supports subsequent operations such as classification or clustering. For speech and audio, many traditional operations in signal processing (e.g., FFTs, smoothing) can be viewed as attempts to extract a set of pre-defined, hand-crafted features that capture information about the energy and power spectra of the original signal. An interesting question is whether automatic methods, driven by the statistics of large unlabeled data sets, can provide similarly useful features for classification and visualization of high dimensional data, specifically audio. Recent advances in unsupervised learning have led to automatic methods for dimensionality reduction that seem worth exploring for this purpose.

The input signals to many information processing systems can be viewed, in their native format, as high dimensional streams. For example, we can view the images in a video as points in a high dimensional vector space whose dimensionality is equal to the number of pixels [1]. Similarly, in speech and audio, we can view the power spectra from windowed FFTs as points in a high dimensional vector space, of dimensionality equal to the window size. Data sets of such high dimensionality present fundamental challenges for pattern recognition algorithms that must generalize from small training samples and perform robustly in the presence of noise. This is one manifestation of the so-called “curse of dimensionality” [2].

How can we overcome the challenges posed by high dimensional data sets? One answer is provided by recent work in unsupervised learning. Often, the underlying variability of a data set is parameterized by a small number of continuous features; in

this case, we can imagine the data as lying on a low dimensional manifold [3] in the high dimensional input space. Such features might include, for example, the angle and lighting of objects in images, or the timbre and pitch of instruments in music. Unsupervised algorithms for dimensionality reduction are designed to discover these features and to compute a faithful low dimensional embedding of high dimensional inputs.

There are linear and nonlinear methods for dimensionality reduction. Principal component analysis (PCA) [4] is a linear method for dimensionality reduction that projects the data into the subspace with the minimum reconstruction error. Though widely used for its simplicity, PCA is limited by its underlying assumption that the data lies in a linear subspace. Recently, several algorithms for nonlinear dimensionality reduction [5, 6, 7, 8] have been proposed that overcome this limitation of PCA. Like PCA, these algorithms are simple to implement, but they compute nonlinear embeddings of high dimensional data. So far, these algorithms have mainly been applied to data sets of images and video, where they have revealed low dimensional manifolds not detected by purely linear methods. In this paper, we apply one of these algorithms—Locally Linear Embedding (LLE) [6, 7]—to the exploratory analysis and visualization of data sets in speech and music.

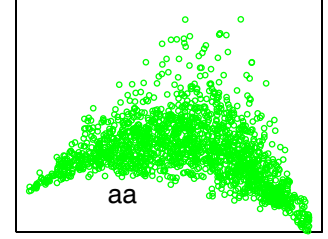
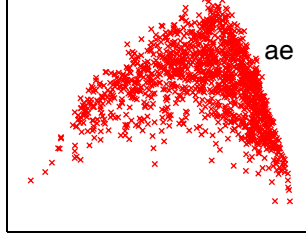
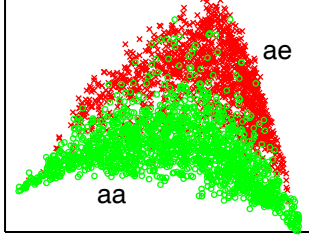
2. LOCALLY LINEAR EMBEDDING

We begin by reviewing the algorithm for LLE; more details can be found in previous work [7]. Given high dimensional inputs $\{\vec{X}_i\}_{i=1}^N$ in D dimensions, LLE attempts to discover low dimensional outputs $\{\vec{Y}_i\}_{i=1}^N$ in $d \ll D$ dimensions that are similarly co-located with respect to their neighbors. The focus on *local* geometric properties distinguishes LLE from other algorithms for dimensionality reduction, such as multidimensional scaling [9] and Isomap [5], which attempt to preserve *global* properties such as the pairwise distances between all inputs. The focus on local properties in LLE also has the benefit of yielding sparse matrices, whose operations and optimizations scale better to large data sets.

The algorithm has three steps. The first step of the algorithm is to compute neighbors for each high dimensional input $\vec{X}_i$. Although neighbors may be computed by arbitrarily sophisticated criteria (for example, incorporating prior knowledge), in this paper we adopt the simplest possible convention—computing the K nearest neighbors for each input based on Euclidean distance.

The second step of LLE appeals to the idea that neighboring inputs lie on (or near) a locally linear patch of the manifold from which they are sampled. For each input $\vec{X}_i$, a set of K linear coefficients are computed that reconstruct the input from its neighbors.

LLE



PCA

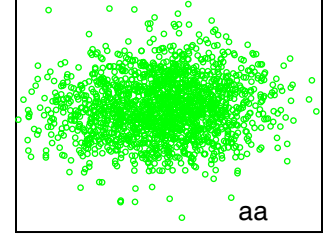
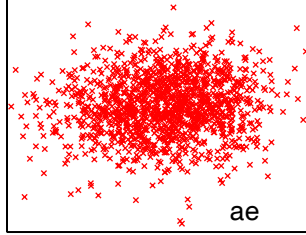
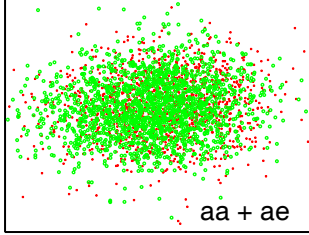


Fig. 1. Two dimensional embeddings of $N=3000$ utterances of the vowels ‘aa’ and ‘ae’ (as in the words HOT and HAT) extracted from the TIMIT corpus. Top row: results from LLE with $K=10$ nearest neighbors. Bottom row: results from PCA.

The total reconstruction error of the inputs is measured by the cost function:

$$\mathcal{E}(W) = \sum_i \left| \vec{X}_i - \sum_j W_{ij} \vec{X}_j \right|^2. \quad (1)$$

In this cost function, the weight W_{ij} stores the contribution of $\vec{X}_j$ to the linear reconstruction of $\vec{X}_i$. The cost function is minimized subject to two constraints: first, that each input is reconstructed only from its nearest neighbors, or $W_{ij}=0$ if $\vec{X}_j$ is not a neighbor of $\vec{X}_i$; second, that the reconstruction weights for each input sum to one, or $\sum_j W_{ij} = 1 \forall i$. The optimal weights for each input can be computed efficiently by solving a constrained least squares problem. These weights characterize the local geometric properties of the data set around each input—in particular, the distances and angles between each input and its neighbors.

The third and final step of the algorithm is to compute a low dimensional embedding that is characterized by the same reconstruction weights (and thus preserves the local geometric properties of the inputs). Specifically, we choose the low dimensional outputs $\vec{Y}_i$ to minimize the embedding cost function:

$$\Phi(Y) = \sum_i \left| \vec{Y}_i - \sum_j W_{ij} \vec{Y}_j \right|^2. \quad (2)$$

This cost function is minimized subject to two constraints that make the problem well-posed: first, that the outputs are centered on the origin, $\sum_i \vec{Y}_i = \vec{0}$; second, that the output covariance matrix equals the $d \times d$ identity matrix. The minimum of eq. (2) is obtained by computing the bottom $d+1$ eigenvectors of the $N \times N$ matrix $(I - W)^T(I - W)$. The bottom eigenvector (which represents a translational degree of freedom) is discarded, and the remaining d eigenvectors yield a solution that minimizes eq. (2) subject to the centering and orthogonality constraints.

The LLE algorithm has two free parameters: the number of neighbors K in the first step, and the target dimensionality d in the third step. The number of neighbors should always be greater than the target dimensionality. Moreover, if we represent the inputs by the vertices of an undirected graph with edges connecting

neighboring inputs, then LLE should only be applied to inputs that give rise to connected graphs. Naturally, the target dimensionality depends on the intended use of the embedding, with $d=2$ or $d=3$ typically chosen for exploratory analysis and visualization. For other tasks, however, estimating the underlying dimensionality of sampled manifolds is an important issue that LLE does not itself address. Various methods have been proposed [10, 11, 12] to estimate this dimensionality; these can be used in conjunction with the first step of LLE to select the target dimensionality, d .

3. EXPERIMENTAL RESULTS

We used LLE as a tool for exploratory analysis and visualization of audio signals. Data sets were generated from both speech and music. Some of our more interesting findings are reported below.

3.1. Speech

The multi-speaker TIMIT corpus [13] was used to perform several experiments on low dimensional embedding of speech data. The phonetic transcriptions of the corpus were used to locate segments of speech corresponding to particular phonemes. One frame was extracted from the middle 20-100 ms of each phonetic segment (downsampled to 8 kHz), with a frame length that depended on the phoneme identity—longer for vowels, and shorter for consonants. Frames were hamming windowed, and the log-power spectra (computed by FFTs) provided the high dimensional inputs for unsupervised learning by PCA and LLE.

Fig. 1 compares the two dimensional embeddings obtained by PCA and LLE for $N=3000$ utterances of the vowels ‘aa’ and ‘ae’ (as in the words HOT and HAT). The data set for this experiment contained 1342 ‘aa’ frames and 1658 ‘ae’ frames extracted from the middle 100 ms of each utterance. (Shorter utterances of duration less than 100 ms were discarded from the data collection.) The log-power spectra of these 100 ms segments gave rise to inputs with $D=401$ dimensions, and LLE was performed using $K=10$ nearest neighbors. The leftmost plots in Fig. 1 show both phonemes plotted together, while the adjacent plots show

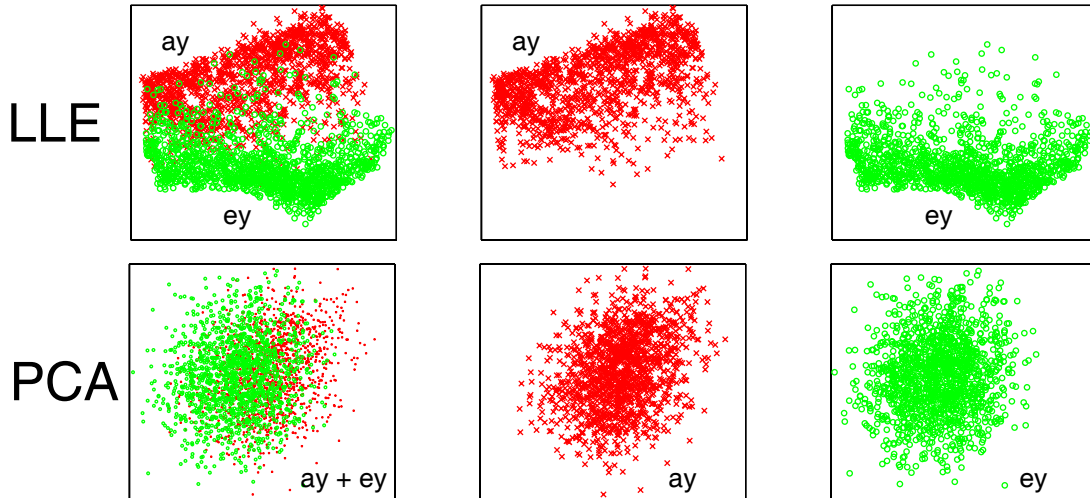


Fig. 2. Two dimensional embeddings of $N = 2731$ utterances of the vowels ‘ay’ and ‘ey’ (as in the words BITE and BAIT) extracted from the TIMIT corpus. Top row: results from LLE with $K = 10$ nearest neighbors. Bottom row: results from PCA.

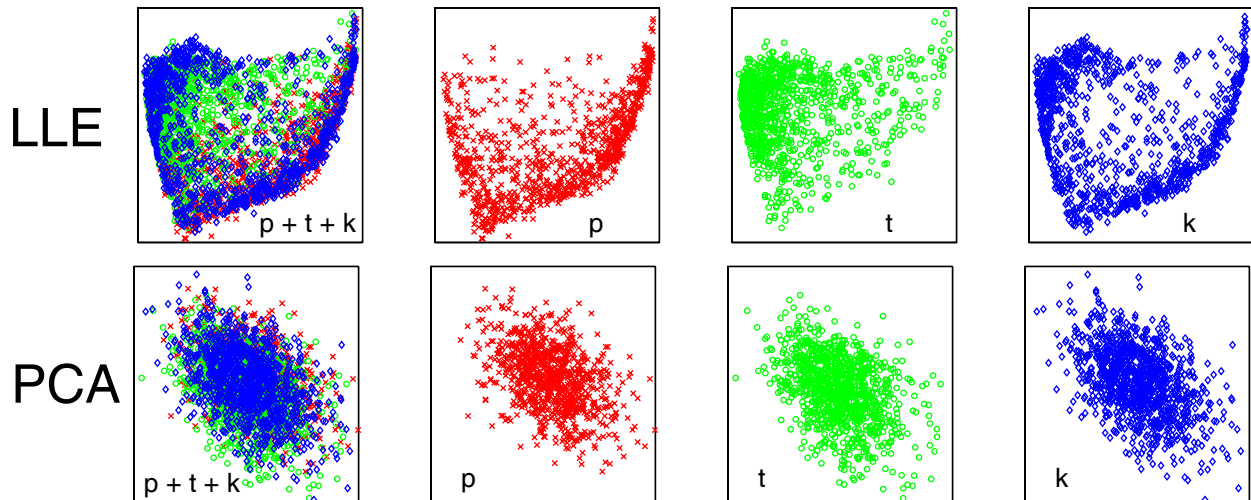


Fig. 3. Two dimensional embeddings of $N = 3000$ utterances of the plosives ‘p’, ‘t’, and ‘k’ extracted from the TIMIT corpus. Top row: results from LLE with $K = 10$ nearest neighbors. Bottom row: results from PCA.

each phoneme plotted independently (from the same embedding and on identical axes). Interestingly, the $d = 2$ embedding of LLE separates the vowels quite well, while the first two principal components from PCA fail to separate the vowels in any meaningful way.

Fig. 2 shows the results of a similar experiment for $N = 2371$ utterances of the vowels ‘ay’ and ‘ey’ (as in the words BITE and BAIT). The data set for this experiment contained 1366 ‘ay’ frames and 1364 ‘ey’ frames extracted from the middle 60 ms of each utterance. The log-power spectra of these 60 ms segments gave rise to inputs with $D = 241$ dimensions. While the first two linear principal components separate these vowels better than the previous experiment, LLE again provides a far cleaner separation.

Finally, Fig. 3 shows the results of a similar experiment for $N = 3000$ utterances of the plosives ‘p’, ‘t’, and ‘k’. The data

set for this experiment contained 937 ‘p’ frames, 1017 ‘t’ frames, and 1,046 ‘k’ frames extracted from the middle 40 ms of each utterance. The log-power spectra of these 40 ms segments gave rise to inputs with $D = 161$ dimensions. Though the plosives do not separate nearly as well as the vowels in Figs. 1 and 2, LLE again reveals considerably more structure than PCA. Moreover, the ‘p’ and ‘t’ frames do cluster in fairly distinct parts of the nonlinear embedding given by LLE.

3.2. Music

We experimented with LLE on a sample of monophonic music to see how it would represent acoustic variability arising from changes in pitch. A data set was generated from the opening 3.5 seconds of Bach’s Cello Suite No.1 Prelude, performed by Yo-Yo

Ma, sampled at 8 kHz. Log-power spectra were computed from 20 ms windows with a frame shift of 5 ms, generating $N = 664$ inputs in $D = 80$ dimensions. LLE was performed with $K = 8$ nearest neighbors.

Fig. 4 shows the aligned waveform, spectrogram, musical score, and results from LLE. All plots are on the same time scale, and the musical score has been adjusted to closely match the timing of the recording. The first and second coordinates of the low dimensional embedding are plotted separately as a function of the time. Interestingly, the second coordinate of the embedding ('lle2') is strongly correlated with the pitch of the note as given by the score, while the first coordinate ('lle1') appears to reflect a more general harmonic relationship.

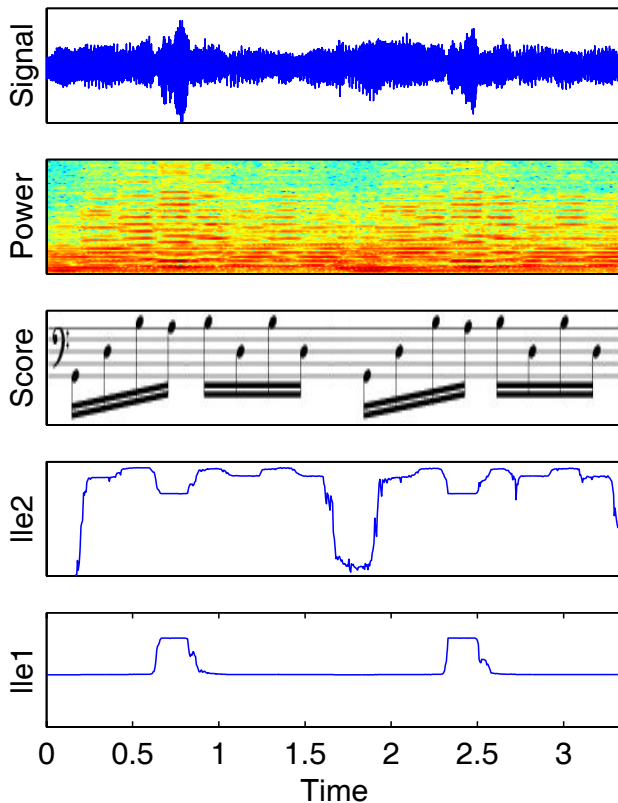


Fig. 4. Locally linear embedding (with $K = 8$ nearest neighbors) of the opening phrase of Bach's Cello Suite No. 1.

4. DISCUSSION

In this paper we have presented results from the exploratory analysis and visualization of speech and music using LLE. For speech, we observed that the nonlinear embeddings of LLE separated certain phonemes better than the linear projections of PCA. For monophonic music, LLE appeared to discover features that were strongly correlated with pitch.

Our results suggest several ways that LLE could be used for speech and audio processing. For example, the embeddings from LLE could be incorporated into the emission distributions of hidden Markov models—specifically, mixture distributions with

Gaussian factor analyzers [7, 14]. They could also be used for visualization and indexing of large multimedia data sets. These and other directions will be explored in future work.

5. ACKNOWLEDGEMENTS

We gratefully acknowledge K. Weinberger, A. Burgoyne, F. Sha, and S. Roweis for useful discussions. This work was supported by NSF award 0238323.

6. REFERENCES

- [1] D. Beymer and T. Poggio, "Image representation for visual learning," *Science*, vol. 272, pp. 1905, 1996.
- [2] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, 1996.
- [3] H. S. Seung and D. D. Lee, "The manifold ways of perception," *Science*, vol. 290, pp. 2268–2269, 2000.
- [4] I. T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, New York, 1986.
- [5] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.
- [6] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.
- [7] L. K. Saul and S. T. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *Journal of Machine Learning Research*, vol. 4, pp. 119–155, 2003.
- [8] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15(6), pp. 1373–1396, 2003.
- [9] T. Cox and M. Cox, *Multidimensional Scaling*, Chapman & Hall, London, 1994.
- [10] K. Pettis, T. Bailey, A. Jain, and R. Dubes, "An intrinsic dimensionality estimator from near-neighbor information," *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI*, vol. 1(1), pp. 25–37, 1979.
- [11] M. Brand, "Charting a manifold," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds., Cambridge, MA, 2003, MIT Press.
- [12] B. Kegl, "Intrinsic dimension estimation using packing numbers," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds., Cambridge, MA, 2003, MIT Press.
- [13] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "The DARPA TIMIT acoustic-phonetic continuous speech corpus," NTIS order number PB91-100354, 1992.
- [14] L. K. Saul and M. G. Rahim, "Maximum likelihood and minimum classification error factor analysis for automatic speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 8(2), pp. 115–125, 1999.