Fast Non-Linear Dimension Reduction

Nandakishore Kambhatla and Todd K. Leen
Department of Computer Science and Engineering
Oregon Graduate Institute of Science and Technology
19600 N.W. von Neumann Drive, Beaverton OR 97006-1999

***Abstract*—**

This paper presents a new algorithm for non-linear dimension reduction. The algorithm builds a *piece-wise linear* model of the data. This piece-wise linear model provides compression that is superior to the *globally linear* model produced by principal component analysis. On several examples the piece-wise linear model also provides compression that is superior to the *global non-linear* model constructed by a five-layer, autoassociative neural network. Furthermore, the new algorithm trains significantly faster than the autoassociative network.

I. Introduction

Pattern recognition systems typically involve high-dimensional feature vectors. However, classifiers that receive high-dimensional feature vectors can be slow to train, and may generalize poorly due to the large number of model parameters. These problems can be ameliorated by dimension-reduction techniques.

Dimension reduction techniques aim to map high-dimensional feature vectors onto lower-dimensional vectors, maintaining sufficient information to discriminate between object classes. This is achieved by identifying and reducing statistical redundancy in the original feature set.

Principal component analysis (PCA) is a classical statistical technique used for data analysis and dimension-reduction. In PCA dimension reduction, the original n-dimensional space is projected onto the m-dimensional *linear subspace* ($m < n$) spanned by the eigenvectors of the data's correlation or covariance matrix corresponding to the largest eigenvalues [1, 2, 3, 4]. One can choose the target dimension m by specifying the tolerable mean square error (MSE) in the reduced representation.

Since PCA is based on second moments it is unable to detect and eliminate higher-order redundancies in the data. In effect, the PCA projection models the original data as an m-dimensional Gaussian *signal* plus $n - m$ *noise* degrees of freedom. The components of the dimension-reduced vectors form global coordinates on the m-dimensional hyperplane spanned by the principal correlation eigenvectors. The original n-dimensional data vectors lie *near* this hyperplane in a least MSE sense.

This global, linear subspace model can fail to be optimal in several ways. As a simple example, suppose that the data lies on, or near, a *curved* p-dimensional submanifold of R^n. Projecting the data onto a p-dimensional *linear* subspace can result in a large error in the representation. To achieve a low error one must increase the target dimension to m, with $m > p$. Thus, one is forced to represent the data in a space of higher dimension than the number of intrinsic degrees of freedom (p in this example). Under such circumstances, *non-linear* dimension reduction can provide better compression than PCA.

This paper presents a new algorithm for non-linear dimension reduction. The basic idea is to construct a *locally linear* model of the input data. We demonstrate that our locally linear model can produce more accurate encodings than both PCA and the *global nonlinear* model produced by a five-layer autoassociative network. Furthermore, the locally-linear model can be significantly quicker to train than the autoassociative network.

II. Global Nonlinear Dimension Reduction

Several researchers [5, 6, 7] have used three-layer feedforward autoassociative networks to perform dimension reduction. These networks consist of an input layer that receives the raw data vectors $x \in R^n$, a hidden layer with $m < n$ nodes, and an output layer whose node activities we denote by $x' \in R^n$. The network weights and biases are trained by error backpropagation with the target outputs equal to the inputs $x' = x$. If the training is successful, the network approximates an identity mapping on the set of input vectors. Since the hidden layer has *fewer* nodes than the input and output layers, the network is forced to develop a compact (i.e. lower dimensional) representation of the data in the hidden layer.

For three-layer networks, auto-associative training is equivalent to a PCA projection. Baldi and Hornik [8] consider three-layer auto-associative networks with *linear* node activation. They show that the transformation performed by the trained network is an orthogonal projection onto the space spanned by the leading m eigenvectors of the input's covariance matrix. Bourlard [9] and Funahashi [10] extend this work, showing that even with *non-linear* nodes in the hidden layer a three-layer autoassociative network cannot achieve compression superior to PCA.

Recently, several researchers have realized that *five-layer* autoassociative networks *can* improve on PCA [11, 12, 13, 14]. These networks have three hidden layers. The first and third hidden layers of nodes have non-linear response, and are referred to here as the *mapping layers.* The $m < n$ nodes of the second hidden layer have linear response. The activities of these nodes form the compressed representation. We refer to this layer as the *representation layer.*

The activities of the nodes in the representation layer form coordinates on a, generally curved, submanifold of the input space. The second mapping layer maps these coordinate values back into R^n. The network must pick these coordinate mappings to cover the entire domain of the input data. We thus refer to five-layer autoassociative networks as a *global, nonlinear* compression technique.

III. Locally Linear Dimension Reduction

An alternative to laying down a single *global* coordinate system that covers a submanifold of the input space is to lay down *local* coordinate patches; with each patch responsible only for a small region of the input space. If the regions are small enough, then one can locally approximate the data manifold as a hyperplane. Within such regions, PCA is sufficient. This suggests a *locally linear* model for the data. [1]

A. VQPCA

In our implementation we use a vector quantizer (VQ) to define the regions for the local PCA. The disjoint regions defined by the VQ are called *Voronoi cells.* We use a standard competitive learning algorithm [17, and references therein] with Euclidean distortion measure, to train the VQ. The training algorithm places reference vectors (network weights) at the mean of the data points that fall in each Voronoi cell. We also implemented the partitioning using a hierarchical multi-stage vector quantizer [18]. The advantage of a multi-stage architecture is that we effectively obtain N Voronoi cells by training fewer than N weights. For example, using a two stage vector quantizer with N weights in each level, we effectively obtain N^2 Voronoi cells by training only $2N$ weights. We refer to the hybrid algorithm of clustering and local PCA as VQPCA.

To summarize, the model is constructed in two phases:

1. Train a VQ with N reference vectors, or weights, $(r_1, \ldots r_N)$.
2. Perform a *local* PCA within each Voronoi cell. For each cell V_i, the local covariance matrix $CM_i \equiv E_i[\,(x-r_i)(x-r_i)^T\,]$ is computed, where the expectation is over all training vectors that fall in V_i. Compute the eigenvectors $(e_1^i, \ldots, e_n^i)$ for each CM_i. [2]

Finally, a target dimension m is chosen (identical for all V_i). Each data point $x \in V_i$ is projected onto the leading m eigenvectors of CM_i to give local linear coordinates $z = (\, e_1^i \cdot x, \ \ldots, \ e_m^i \cdot x\,)$.

The compressed representation for each data point x consists of the index i of the Voronoi cell in which x falls, together with the m component vector z. The data is reconstructed from this representation according to

$$x' = r_i + \sum_{k=1}^{m} z_k \; e_k^i . \qquad (1)$$

We use the MSE to assess the accuracy of the compressed representation. To facilitate interpreting the error as a percentage of the signal strength, we report a normalized MSE defined as

$$\mathcal{E}_{norm} = \frac{E[\,||\,x-x'\,||^2\,]}{E[\,||x||^2\,]} \qquad (2)$$

where the expectation $E[\cdot]$ is over the points in the data set.

IV. Experimental results

In this section we compare the performance of VQPCA with five-layer networks (hereafter referred to as 5LNs),

[1] While this work was in progress, we became aware of the use of *local* PCA in several related contexts. Broomhead [15] suggests the use of local PCA, together with scaling arguments, to determine the dimension of signals arising from chaotic dynamical systems. In earlier work, Fukunaga [16] proposed an interactive algorithm that uses a local PCA to find the intrinsic dimension of data sets. The goal of our work is to provide dimension reduction of data using an algorithm that develops a locally linear data model.

[2] We computed the eigensystems using standard matrix techniques [19]. Alternatively one could use a neural network algorithm such as that given in [20].

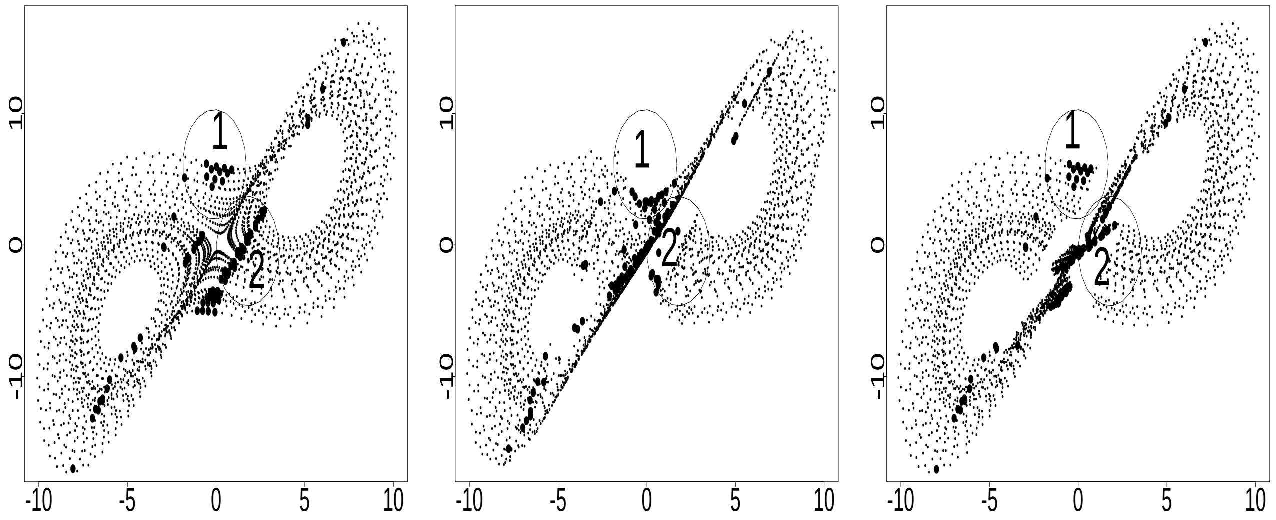

Figure 1: Lorenz attractor data : (a) Original data set, (b) Reconstructed data set from a 5LN-BFGS encoding and (c) Reconstructed data set from a VQPCA encoding. The darker points are those for which the 5LN had its squared error greater than 1.5.

and with PCA (computed from the data's covariance matrix) The first example is a low-dimensional problem that graphically contrasts compression using global coordinates with compression using local coordinates. The other examples deal with compression of speech signals.

We compare the algorithms in terms of the reconstruction errors and training times. All errors reported in this section are in terms of $\mathcal{E}_{norm}$ defined in (2). The 5LNs are trained using conjugate gradient descent (hereafter referred to as 5LN-CGD) or a quasi-Newton second order method (the BFGS algorithm [19]; hereafter referred to as 5LN-BFGS). We use the same number of nodes in both the mapping layers (first and third hidden layers) in order to limit the search in the space of network architectures. The optimal value of the number of nodes in the mapping layers, is estimated by varying it over a range of values, and choosing the value for which the test set error is lowest. Similarly, we estimate the optimal number of Voronoi cells for VQPCA by varying it over a range of values and then choosing the value for which the test set error is lowest. We implement the clustering for VQPCA using a flat VQ or a hierarchical multistage VQ.

A. A Geometrical Example

In this example, data from numerical integration of the Lorenz equations [21] (3-dimensional vectors) is compressed to a 2-dimensional representation. Figure 1(a) shows a two dimensional view of 2800 points on an orbit asymptotic to the attractor. Over much of the attractor, the data is well-located by two coordinates. However in

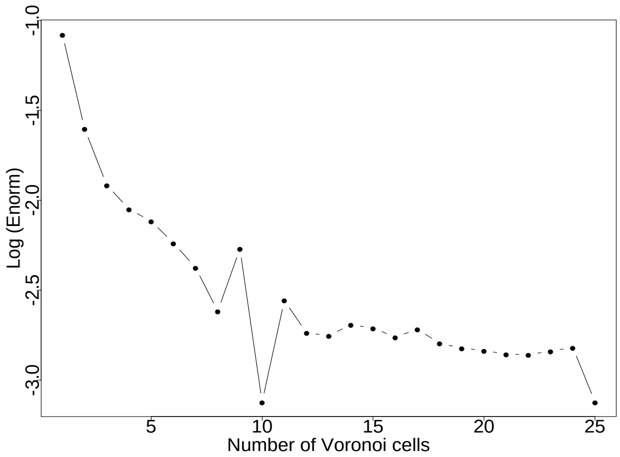

Figure 2: The variation of the logarithm of the reconstruction error $\mathcal{E}_{norm}$ (log base 10) with respect to the number of Voronoi cells for VQPCA for the Lorenz data set.

the region where the two spiral wings of attractor join, the data more nearly fills 3-space. In this transition region the orbits become inclined to the plane of each wing.

We ran experiments with 5LNs (both 5LN-CGD and 5LN-BFGS) configured as 3-mp-2-mp-3, with mp varying from 5 to 50 in increments of 5. We obtained the smallest reconstruction error $\mathcal{E}_{norm}$ with $mp = 15$ for a 5LN trained with the BFGS algorithm. For VQPCA, we varied the number of Voronoi cells from 1 to 25, and obtained the least error with 10 Voronoi cells. However the error with 10 Voronoi cells was anomalously *low* (see Figure 2), and we are therefore reporting the error for VQPCA with 22 cells.

Figure 1(b) shows the data reconstructed from a 2-dimensional representation generated by a 5LN (5LN-BFGS) with 15 nodes in the mapping layers. The 5LN fails to obtain an accurate encoding for the points lying near the transition regions between the two wings of the attractor. For example, it is clear from the figure that the 5LN fails badly in the regions marked 1 and 2. The 5LN attempts to cover the attractor with a single 2-dimensional coordinate system. In the transition regions, the data more nearly fills 3-space, so the encoding has a high error here. This geometric distortion is not limited to the transition regions, but spreads into the wings. This is clearly seen in Figure 1(b). Presumably this spreading occurs because the transformations constructed by each layer of the network are smooth.

Architecture	$\mathcal{E}_{norm}$	*Training Time (in seconds)*
PCA	0.08345	5
5LN	0.00266	8,330
VQPCA	0.00139	56

Table 1: Reconstruction errors and training times for a 2-D compression of the Lorenz data set by different architectures. The 5LN (5LN-BFGS) reported here had 40 nodes in the mapping layers, while the VQPCA reported here was with 22 Voronoi cells.

As seen in Figure 1(c), VQPCA does much better. This is because it is a purely local procedure, so the mismatch between the 2-dimensional model and the structure of the data in the transition regions does not affect the encoding in the adjacent regions. Hence for all the Voronoi cells except the ones which lie right in the transition regions, the compression is very accurate. This advantage is reflected in the errors $\mathcal{E}_{norm}$ reported in Table 1. The table also reports the time (in Sparc 2 cpu seconds), required to construct the encoding model. Note the tremendous time advantage of VQPCA, relative to the 5LNs.

B. Speech Compression

In these set of experiments, we compress speech signals using VQPCA, 5LNs and PCA. The data sets consist of the vowel portions extracted from the isolated utterances of letters and continuous speech. Each input vector consists of the lowest 32 discrete Fourier transform (DFT) coefficients (spanning the frequency range 0-4kHz), time-averaged over the central third of the sonorant (sounds produced solely by the vibration of the vocal chords).

It is widely recognized that sonorants are distinguished by the frequencies of the lowest *three* resonances of the vocal tract, called formants. A compressed encoding of two or three dimensions should be able to capture the formant frequencies. We therefore conducted experiments compressing the 32-dimensional input data down to 2 and 3 dimensions.

We varied the free parameters in the algorithms over a range of values, and estimated the optimal value by a value for which the test set error was lowest. For 5LNs trained with conjugate gradient descent (5LN-CGD), we varied the number of nodes in the mapping layers from 5 to 50 in increments of 5. However for 5LNs trained with the second order method (5LN-BFGS), the increased amount of storage space required makes it non feasible for larger networks. Hence, for 5LN-BFGS, we varied the number of nodes in the mapping layers from 5 to 25 in increments of 5. All errors reported for 5LNs are the best of three runs with different random initializations for the weights.

Similarly, we varied the number of Voronoi cells for VQPCA from 1 to 50. For the multistage VQPCA (VQPCA-MS), we used a configuration of N weights in the first level and N weights in the second level (denoted by NxN). This effectively results in a partitioning of the input space into N^2 Voronoi cells. In the experiments described below, we varied N from 2 to 10.

B.1. The ISOLET database

The first data set that we used consists of isolated utterances of the letters A,E,F,O and R, spoken by both males and females, from the ISOLET database [22]. The training set contained 225 utterances. The test set contained 75 utterances from speakers not represented in the training set.

Tables 2 and 3 summarize the relative performance of PCA, 5LN (CGD and BFGS) and VQPCA with flat and multi-stage clustering in terms of compression accuracy, and training times. We note that using VQPCA-MS, we obtained an error .6-.8 times the error obtained by using the 5LNs (either 5LN-CGD or 5LN-BFGS). Moreover, training the VQPCA-MS was 16-48 times faster than training the 5LNs (5LN-CGD or 5LN-BFGS). In separate

Architecture	$\mathcal{E}_{norm}$	*Training Time (in seconds)*
PCA	0.0623	5
5LN-CGD	0.0566	2,609
5LN-BFGS	0.0523	3,845
VQPCA	0.0352	866
VQPCA-MS	0.0349	80

Table 2: Reconstruction errors (for the **test** set) and training times for 2-D compression of the ISOLET data. The 5LN-CGD had 40 nodes in the mapping layers while the 5LN-BFGS had 10 nodes in the mapping layers. The VQPCA was with 46 Voronoi cells. The VQPCA-MS had a two-level 8x8 configuration.

Architecture	$\mathcal{E}_{norm}$	*Training Time (in seconds)*
PCA	0.04502	5
5LN (CGD)	0.04089	1,287
5LN (BFGS)	0.03795	2,910
VQPCA	0.03008	850
VQPCA-MS	0.03013	80

Table 3: Reconstruction errors (for the **test** set) and training times for 3-D compression of the ISOLET data. Both the 5LN-CGD and the 5LN-BFGS had 10 nodes in the mapping layers. The VQPCA was with 45 Voronoi cells. The VQPCA-MS had a two level 8x8 configuration.

Architecture	$\mathcal{E}_{norm}$	*Training Time (in seconds)*
PCA	0.00473	11
5LN (CGD)	0.00480	2,861
5LN (BFGS)	0.00440	22,879
VQPCA	0.00325	2,086
VQPCA-MS	0.00311	168

Table 5: Reconstruction errors (for the **test** set) and training times for 3-D compression of the TIMIT data. The 5LN-CGD had 15 nodes in the mapping layers while the 5LN-BFGS had 20 nodes in the mapping layers. The VQPCA was with 43 Voronoi cells. The VQPCA-MS had a two level 10x10 configuration.

experiments, we found that using the encodings generated by VQPCA with flat clustering, we were able to achieve a higher classification accuracy than with the encodings generated by 5LNs or PCA.

B.2. The TIMIT database

The second data set we used was from the TIMIT [23] database. The data set consists of the 12 monothongal vowels (*/iy/, /ih/, /eh/, /ae/, /ix/, /ax/, /ah/, /uw/, /uh/, /ao/, /aa/,* and */er/*) extracted from continuous speech from both males and females. The diphthongs were excluded as they exhibit spectral change which makes them inappropriate to use in experiments using time-averaged spectral coefficients. The training set contained 1200 vectors. The test set contained 816 vectors taken from utterances spoken by speakers not represented in the training set.

Tables 4 and 5 summarize the relative performance of PCA, 5LN (CGD and BFGS) and VQPCA (with various clustering schemes) in terms of compression accuracy, and training times. We note that training the VQPCA-MS was 12-136 times faster than training the 5LNs (5LN-CGD or 5LN-BFGS). Moreover, using VQPCA-MS, we obtained an error of .6-.7 times the error obtained by using 5LN-CGD or 5LN-BFGS. In separate experiments, we again found that using the encodings generated by VQPCA with a flat clustering, we were able to achieve a higher classification accuracy than with the encodings generated by 5LNs or PCA.

Architecture	$\mathcal{E}_{norm}$	*Training Time (in seconds)*
PCA	0.00606	11
5LN (CGD)	0.00619	2,107
5LN (BFGS)	0.00561	12,663
VQPCA	0.00362	1,454
VQPCA-MS	0.00350	168

Table 4: Reconstruction errors (for the **test** set) and training times for 2-D compression of the TIMIT data. The 5LN-CGD had 20 nodes in the mapping layers while the 5LN-BFGS had 10 nodes in the mapping layers. The VQPCA was with 50 Voronoi cells. The VQPCA-MS had a two level 9x9 configuration.

V. Discussion

We have presented a new hybrid algorithm (VQPCA) for non-linear compression. VQPCA approximates the data distribution with a set of local hyperplanes. The location and the distribution of this set captures the large scale, non-linear structure of the data, while coordinates on the hyperplanes capture the local variations.

We have compared VQPCA with five-layer networks, and with PCA. For the examples that we presented, both of the non-linear techniques (VQPCA and five-layer networks) have a definite advantage over PCA for accurate compression. In general, the advantage offered by a *non-linear* technique will depend on the structure of the data. Our results indicate that VQPCA can generate more accurate encodings, and can take significantly less time to train, than five-layer networks.

Issues that remain to be explored include establishing criteria for selecting the number of Voronoi cells and the target dimension. Information theoretic criteria can presumably be used to select these parameters. Allowing compression to different dimensions in different Voronoi cells should further enhance the efficiency of compression for storage/transmission purposes.

Acknowledgements

We would like to thank Professors Ronald Cole and Mark Fanty for continued interest in this work, and for access to speech data and tools.

References

[1] Satosi Watanabe. Karhunen-Loeve expansion and factor analysis, theoretical remarks and applications. In *Transactions of the 4th Prague Conference on Information Theory, Statistical Decision Functions and Random Processes*, pages 645–660, 1965.

[2] S Watanabe. Feature compression. In J. T. Tou, editor, *Advances in information Systems Science, vol. 3*, pages 63–111. Plenum, 1970.

[3] P.A. Devijner and J. Kittler. *Pattern Recognition, a Statistical Approach*. Prentice / Hall, Englewood Cliffs, New Jersey, 1982.

[4] E. Oja. *Subspace Methods of Pattern Recognition*. John Wiley & Sons Inc., New York, 1983.

[5] Garrison W. Cottrell and Janet Metcalfe. EMPATH: Face, emotion, and gender recognition using holons. In R. Lippmann, John Moody, and D. Touretzky, editors, *Advances in Neural Information Processing Systems 3*, pages 564–571. Morgan Kauffmann, 1991.

[6] Garrison W. Cottrell, Paul Munro, and David Zipser. Learning internal representations from gray-scale images: an example of extensional programming. In *Proceedings of the Ninth Annual Cognitive Science Society Conference, Seattle, Wa*, pages 461–473, 1987.

[7] B.A. Golomb, D.T. Lawrence, and T.J. Sejnowski. Sexnet: A neural network identifies sex from human faces. In R. Lippmann, John Moody, and D. Touretzky, editors, *Advances in Neural Information Processing Systems 3*, pages 572–577. Morgan Kauffmann, 1991.

[8] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: learning from examples without local minima. *Neural Networks*, 2:53–58, 1989.

[9] H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biol. Cyb.*, 59:291–294, 1988.

[10] Ken-ichi Funahashi. On the approximate realization of identity mappings by three-layer neural networks. Technical report, Toyohashi University of Technology, Department of Information and Computer Sciences, 1990. Translation of Japanese paper in Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J73-A, No. 1, 1990, pp 139-145.

[11] Shiro Usui, Shigeki Nakauchi, and Masae Nakano. Internal color representation acquired by a five-layer neural network. In O. Simula T. Kohonen, K. Makisara and J. Kangas, editors, *Artificial Neural Networks*. Elsevier Science Publishers, North-Holland, 1991.

[12] E. Oja. Data compression, feature extraction, and autoassociation in feedforward neural networks. In *Artificial Neural Networks*, pages 737–745. Elsevier Science Publishers B.V. (North-Holland), 1991.

[13] Mark A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37:233–243, 1991.

[14] Aran Namphol, Mohammed Arozullah, and Steven Chin. Higher order data compression with neural networks. In *Proceedings of the IJCNN*, pages I 55–I 59, June 1991.

[15] D. S. Broomhead. Signalprocessing for nonlinear systems. In Simon Haykin, editor, *Adaptive Signal Processing, SPIE Proceedings Vol. 1565*, pages 228–243. SPIE, July 1991.

[16] Keinosuke Fukunaga and David R. Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers*, C-20(2):176–183, 1971.

[17] Stanley C. Ahalt, Ashok Krishnamurthy, Prakoon Cheen, and Douglas Melton. Competitive learning algorthms for vector quantization. *Neural Networks*, 3:277–290, 1990.

[18] Biing-Hwang Juang and A.H. Gray Jr. Multiple stage vector quantization for speech coding. In *Proceeding of the IEEE International Conference on Acoustics and Signal Processing*, pages 597–600, 1982.

[19] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes – the Art of Scientific Computing*. Cambridge University Press, Cambridge / New York, 1987.

[20] T. Sanger. An optimality principle for unsupervised learning. In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems 1*. Morgan Kauffmann, 1989.

[21] John Guckenheimer and Philip Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, volume 42 of *Applied Mathematical Sciences*. Springer-Verlag, 1983.

[22] Ron Cole, Yeshwant Muthusamy, and Mark Fanty. The ISOLET spoken letter database. Technical Report CSE 90-004, Oregon Graduate Institute of Science & Technology, March 1990.

[23] W.M. Fisher and G.R. Doddington. The darpa speech recognition research database : specification and status. In *Proceedings of the DARPA Speech Recognition Workshop*, pages 93–99, Palo Alto, CA, 1986.