Lecture 24: Unsupervised Learning

Mark Hasegawa-Johnson

ECE 537, Fall 2022

Outline

What is Unsupervised Learning?

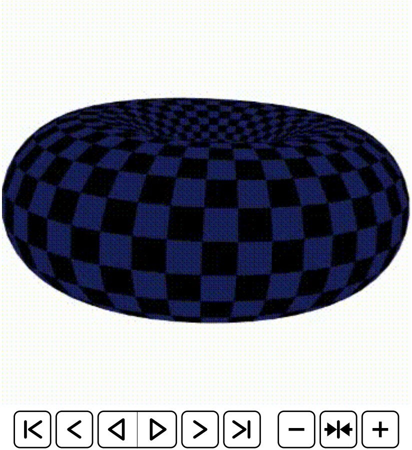
- **Supervised learning:** given pairs of data, (x_i, y_i), learn a mapping $f(x) \approx y$.
- **Unsupervised learning:** given unlabeled training examples, x_i, learn something about them.
  - Can x_i be decomposed into signal + noise?
  - Can we group the x's into "natural classes," i.e., groups of tokens that are similar to one another?
  - Can we design a classifier that puts x_i into its natural class?

Some types of unsupervised learning

- **Manifold learning:** decompose x_i into signal + noise
- **Clustering:** group the x's into natural classes
- **Self-supervised learning:** learn a classifier that puts x_i into its natural class

Outline

- Signals lie on a manifold if some perturbations are impossible ("perpendicular" to the manifold)
- If signals are on a manifold, then perturbations perpendicular to the manifold are always noise, and can be ignored.

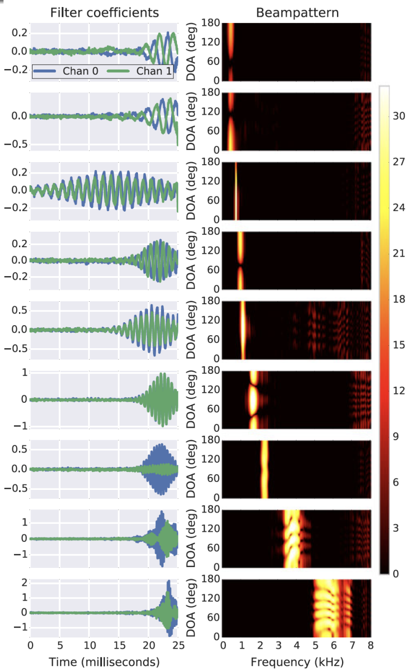


Speech Manifolds

- **The signal manifold:** Each sample is $s[n] = d[n] + \sum a_m s[n-m]$. The excitation, $d[n]$, is sparse: only about 10% of its samples should be nonzero.
- **The articulatory manifold:** The formant frequencies and bandwidths change slowly as a function of time, because they are shaped by positions of the tongue, jaw, and lips, and those things have mass.

The signal manifold:

- When CNNs are learned directly from the speech samples, the first-layer filters tend to look like a Fourier transform.
- Example at right: Figure 2, "Multichannel Signal Processing with Deep Neural Networks for Automatic Speech Recognition," Sainath et al., 2017, (c) IEEE



The articulatory manifold (Deng et al., Interspeech 2010)

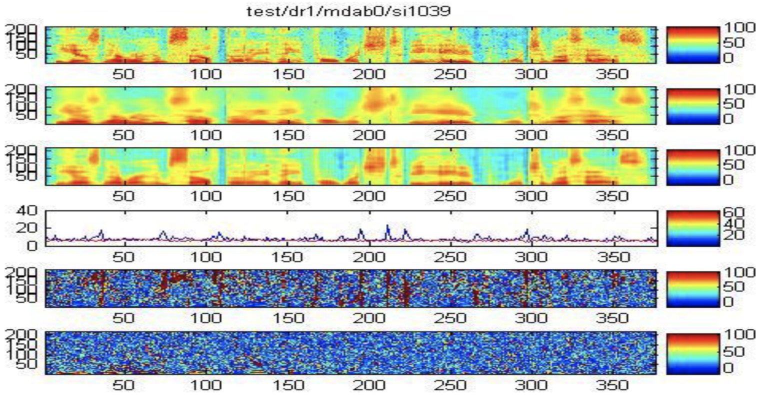
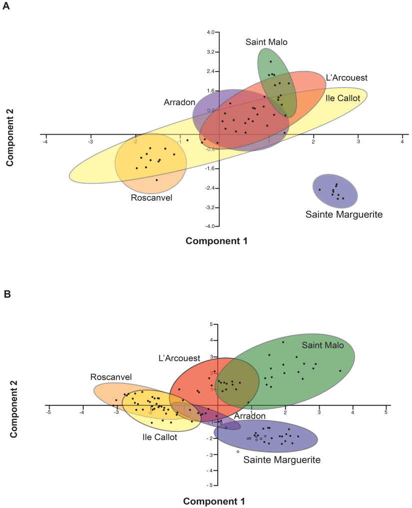


Fig. 4. Top to bottom: Original spectrogram from the test set; reconstruction from the 312-bit VQ coder; reconstruction from the 312-bit auto-encoder (2304-1000-312); coding errors as a function of time for the VQ coder (blue) and auto-encoder (red); spectrogram of the VQ coder residual; spectrogram of the auto-encoder residual.

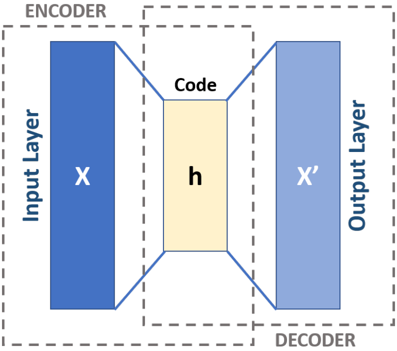
PCA: Manifold = Hyperplane

If the signal is constrained to lie on a hyperplane, then the hyperplane can be found using principal components analysis (PCA).



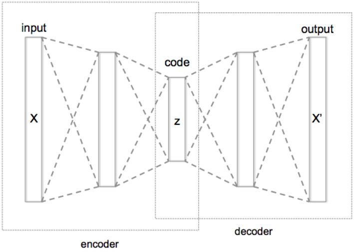
PCA is computed by a one-layer autoencoder

A one-layer autoencoder (one matrix multiply, then a hidden layer, then the inverse of the same matrix) computes the PCA of its input.



Two-layer autoencoder can compute a nonlinear manifold

- A two-layer autoencoder constrains the data, x, to lie on a **nonlinear** manifold of dimension $= \dim(z)$.
- The first layer nonlinearly transforms the input, then the second layer computes PCA of the result.

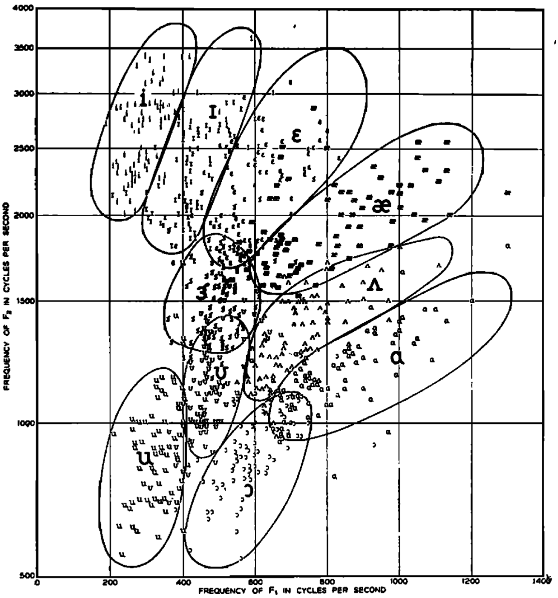


Summary: Manifolds

- Speech lies on manifolds of (at least) two timescales
  - **Signal manifold:** samples are predictable from previous samples
  - **Articulatory manifold:** formant frequencies and bandwidths are predictable from previous formants and bandwidths
- By learning to represent the manifolds, the early layers of an ASR learn to reject irrelevant variation (noise) and keep only relevant variation (signal)
- Autoencoders explicitly learn manifolds

Outline

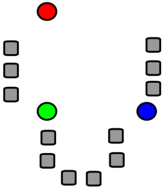
- The idea of **clustering** is to group the observed data into **natural classes** (things that sound similar).
- After grouping them into natural classes, we can then assign a label to each natural class.



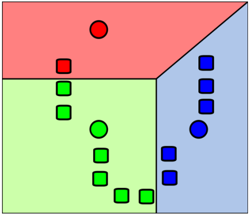
Peterson and Barney, 1952.

K-Means Clustering (https://en.wikipedia.org/wiki/K-means_clustering)

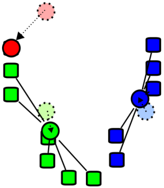
Step 0: Choose random initial "means"



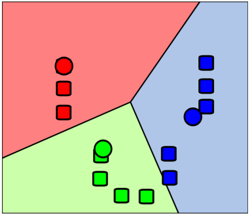
Step 1: Group each token with its closest mean



Step 2: Mean = average of its tokens



Step 3: Repeat step 1



Gaussian Mixture Modeling

Gaussian mixture modeling is like K-means, except that each cluster has a different covariance matrix. Result can be very similar to a natural vowel space.

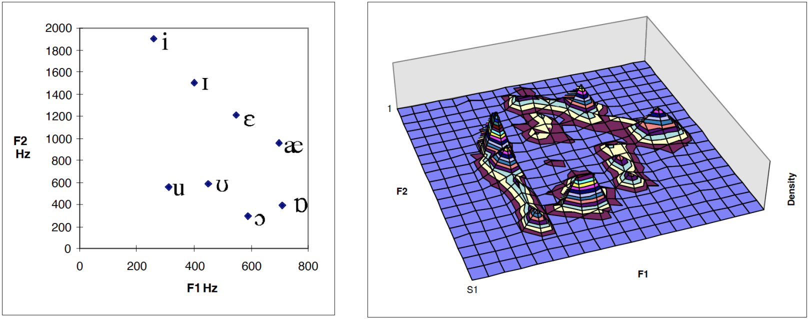


Fig. 1, "Building a Statistical Model of the Vowel Space for Phoneticians," (c) Matthew Aylett, 1998

Outline

Unsupervised Classifier Example

In 1965, Scudder ("Probability of Error of Some Adaptive Pattern-Recognition Machines") considered the following example:

- $Z_1, \ldots, Z_n, \ldots$ is a series of vectors.
- Each vector either contains signal + noise ($Z_n = X + N_n$), or just noise ($Z_n = N_n$).
- The signal, X, is the same every time it appears, but it is unknown.
- The noise, N_n, is zero-mean Gaussian noise with covariance matrix $K_N = \sigma^2 I$.

Unsupervised Classifier Example

Here are the questions Scudder asked:

1. Suppose a classifier was asked to determine whether or not the pattern is present. What it the optimum decision rule?
2. Suppose a classifier was trained **without any training labels**. Can it learn the optimum decision rule?

The Supervised Case

First, consider the supervised case. We don't know X, but we are given labels: $\theta_n = 1$ if $Z_n = X + N_n$, otherwise $\theta_n = 0$. The optimum decision rule turns out to be:

- Update the matched filter covariance estimate:

$$K_{n+1} = \left(K_n^{-1} + \theta_n K_N^{-1}\right)^{-1}$$

- Update the matched filter estimate:

$$H_{n+1} = H_n + K_N^{-1} K_{n+1} (Z_n - H_n)\theta_n$$

- Calculate the log likelihood ratio (LLR):

$$Q_n = Z_n K_n^{-1} Z_n - (Z_n - H_n)^T (K_N + K_n)^{-1} (Z_n - H_n)$$

- Threshold the LLR:

$$\hat{\theta}_n = \begin{cases} 1 & Q_n > \text{threshold} \\ 0 & Q_n \leq \text{threshold} \end{cases}$$

In the supervised case, as $n \to \infty$,

- The matched filter converges $H_n \to X$.
- The matched filter covariance disappears $K_n \to 0$
- The LLR converges to a linear function of Z_n:

$$Q_n \to 2H_n^T K_N^{-1} Z_n - \text{consta}$$

- The optimal classifier converges to a linear classifier.

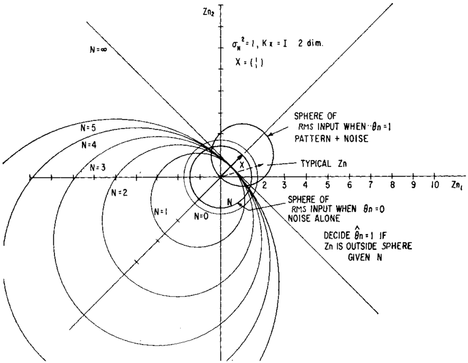


Fig. 2, "Probability of Error of Some Adaptive Pattern-Recognition Machines", (c) IEEE, 1965

The Self-supervised Case

Now, consider the self-supervised case. We don't know X, and we don't know θ_n. Instead, all we have is our own classifier output, $\hat{\theta}_n$, at each time step. Can we learn the optimum classifier?

- Calculate the log likelihood ratio (LLR):

$$Q_n = Z_n K_n^{-1} Z_n - (Z_n - H_n)^T (K_N + K_n)^{-1} (Z_n - H_n)$$

- Threshold the log likelihood ratio:

$$\hat{\theta}_n = \begin{cases} 1 & Q_n > \text{threshold} \\ 0 & Q_n \leq \text{threshold} \end{cases}$$

- Update the matched filter covariance estimate:

$$K_{n+1} = \left(K_n^{-1} + \hat{\theta}_n K_N^{-1} \right)^{-1}$$

- Update the matched filter estimate:

$$H_{n+1} = H_n + K_N^{-1} K_{n+1} (Z_n - H_n) \hat{\theta}_n$$

Figure at right shows the **supervised** case. Can we match it using a **self-supervised** learner?

- The $n = 0$ classifier is still a circle: any **small** vector is classified as $\hat{\theta}_n = 0$, any **large** vector is classified as $\hat{\theta}_n = 1$.
- This is a good thing! Some of the large vectors are, indeed, signals. But not all!

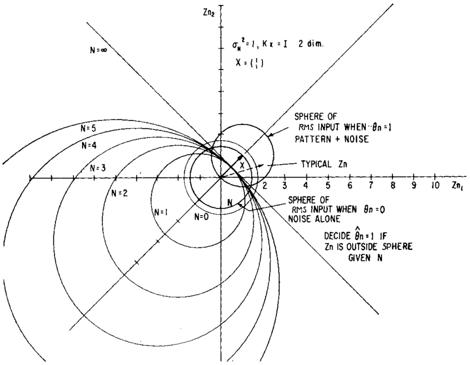


Fig. 2, "Probability of Error of Some Adaptive Pattern-Recognition Machines", (c) IEEE, 1965

The self-supervised learner learns a "matched filter," H_n, such that

- The hyperplane is $H_n^T Z_n = \text{threshold}$.
- H_n is the average of all of the Z vectors on the right side of the hyperplane.

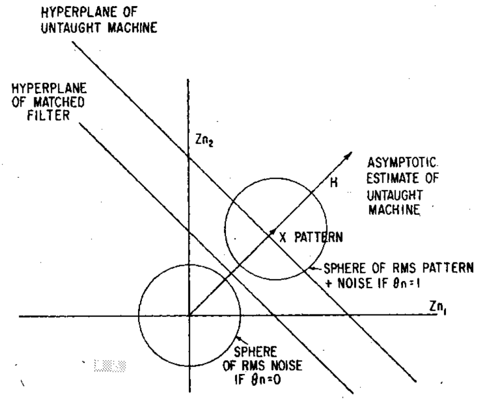


Fig. 4, "Probability of Error of Some Adaptive Pattern-Recognition Machines", (c) IEEE, 1965

Summary: Scudder's Theory of Self-Supervised Learning

- The classifier learns to call all big vectors "signal," and all small vectors "noise."
- It is biased: small signal vectors get misclassified as "noise."
- There is a threshold effect: if the noise covariance matrix, K_N, is too large, then the learner fails to converge.

Outline

Summary

- Manifold learning
  - Learn a low-dimensional representation that captures most of the signal variation
- Clustering
  - Classify each token to its nearest mean
  - Recompute each mean as the average of its tokens
- Self-supervised learning
  - The hyperplane is $H_n^T Z_n =$ threshold.
  - H_n is the average of all of the Z vectors on the right side of the hyperplane.