Linear Data Analysis
Nonlinear Separation - Embedding and Gram Matrix

Cain Susko

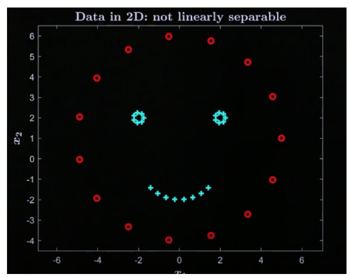
Queen's University
School of Computing

March 29, 2022

a Some Data are Nonlinearly Separable

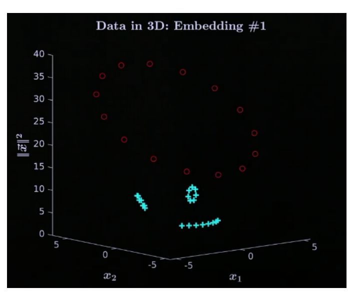
This lecture covers data that is not linearly separable but visually, looks like to can be separated. This can be done using Embedding, which is the opposite of projection.

Example Given the data:

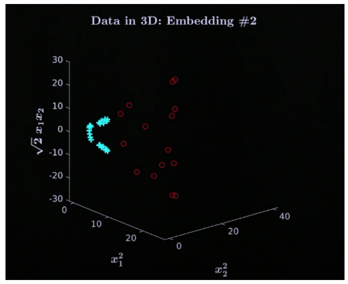


While this data is not linearly separable, it does seem like it can be separated somehow.

Augment To address this we will take the 2D data and project it into 3 dimensions. We could do this by **appending the squared norm of each operation to it's self**, this would result in the following plot:



Map We can also use whats known as a quadratic embedding. Given the same data it would produce:



Both of these techniques allow for the creation of a hyperplane that can then be projected back down to 2 dimensions.

Observation Notation

For $a \in \mathbb{R}$, we write the i^{th} row as

$$\underline{a}_i$$

Which is an unusual notation but will be very useful for us. Additionally, we will allow ourselves a definition for the dot-product; which is a useful fiction, but not mathematically rigorous. It is defined as:

$$\underline{a}_i \cdot \underline{a}_j = \underline{a}_i \underline{a}_j^\top$$

Note: in this and subsequnet notes, we will treat a row as if it is a vector: $\underline{a}_i \in \mathbb{R}^n$

b Embedding a Vector Space

This section will cover what is meant by embedding a vector. embedding is done using a mapping:

$$\phi : \mathbb{R} \hookrightarrow \mathbb{R}^P$$

This will be written as:

$$\underline{\phi}(\underline{u})$$

Note: the hook arrow denotes a embedding.

Example: Augmentation Given the embedding statement:

$$\underline{u} \hookrightarrow \hat{u}$$

We expand the equation to be:

$$[u_1 \;\; u_2] \hookleftarrow [u_1 \;\; u_2 \;\; ||\underline{u}||^2]$$

Example: Polynomial Given the same statement, a polynomial (quadratic) embedding would look like:

$$[u_1 \;\; u_2] \hookrightarrow [u_1^2 \;\; u_2^2 \;\; u_1 u_2 \sqrt{2}]$$

Uses

What are these techniques for embedding used for? Given the data $A \in \mathbb{R}^{m \times n}$, we will embed A such that $A \hookrightarrow \hat{A}$ where:

$$\hat{A} \in \mathbb{R}^{m \times P}$$

If we then did an ordinary PCA the scatter matrix would be:

$$\hat{S} \in \mathbb{R}^{P \times P}$$

And the eigenvectors would be such that:

$$\hat{v} \in \mathbb{R}^{P}$$

Recall With polynomials, the number of coefficients grows and the term is combinatorially in n (as we are raising the vectors of $\underline{u}$ by a power). This is not very computationally efficient and takes up a lot of memory.

c Gram Matrix for an Embedding

Recall the right transpose of a matrix:

$$AA^{\top}$$

It is such that entry (i, j) equals:

$$\underline{a}_i \underline{a}_j^{\top} = \underline{a}_i \cdot \underline{a}_j$$

Consider If we take the right transpose of embedded data we then get:

$$\hat{A}\hat{A}^\top$$

Such that each entry (i, j) is:

$$\underline{\phi}(\underline{a}_i)[\underline{\phi}(\underline{a}_j)]^\top = \underline{\phi}(\underline{a}_i) \cdot \underline{\phi}(\underline{a}_j)$$

Thus, it is clear to see that in either case what we end up doing is either transpositions or dot products.

Kernel

A kernel function is:

$$k(\underline{u}, \underline{v}) = k(\underline{v}, \underline{u})$$

If we are given the finite data $\underline{u}_i \in \mathbb{R}^n$, then the matrix K would be:

$$K_{ij} = k(\underline{u}_i, \underline{u}_j)$$

Where K is positive semi-definite (k is not a kernel function otherwise) and K is known as a **Gram Matrix**.

Example: Augmented Given the embedding:

$$[u_1 \;\; u_2] \hookrightarrow [u_1 \;\; u_2 \;\; (u_1^2 + u_2^2)]$$

The kernel function is equal to:

$$k(\underline{u}, \underline{v}) = \underline{\phi}(\underline{u}) \cdot \underline{\phi}(\underline{v})$$

Which then equals:

$$= \underline{u} \cdot \underline{v} + (\underline{u} + \underline{v})^2$$

Thus, this is the kernel function for an Augmented Embedding

Example: Polynomial Given:

$$[u_1 \ \ u_2] \hookrightarrow [u_1 \ \ u_2 \ \ \sqrt{2}u_1u_2]$$

The kernel function would be:

$$k(\underline{u},\underline{v}) = u_1^2v_1^2 + u_2^2v_2^2 + 2u_1v_1u_2v_2$$

Which simplifies to:

$$= (\underline{u} \cdot \underline{v})^2$$

Note: This technique for embedding is called polynomial because it produces a polynomial kernel.

Usage Both of these techniques produce a kernel which allows us to do embedding computations much faster. Thus, we can compute entry (i, j) of $\hat{A}\hat{A}^\top$ as:

$$k(\underline{a}_i, \underline{a}_j)$$

We then gather the outputs of the kernel function into the augmented weight vector $\hat{W}$ which is symmetric and positive semi-definite. With this new matrix we may want to perform PCA, which *is* possible. doing a PCA on a $\hat{W}$ is known as the kernel trick for PCA.

Kernel Types

There are a few kinds of general kernels:

Linear	$k(\underline{u},\underline{v}) = \underline{uv}^\top$
Quadratic	$k(\underline{u},\underline{v}) = (\underline{uv} + 1)^2$
Polynomial, order l	$k(\underline{u},\underline{v}) = (\underline{u}undv^\top + c)^l$
Guassian	$k(\underline{u},\underline{v}) = \exp(\frac{-\|\|\underline{u} - \underline{v}\|\|^2}{2\sigma^2})$

In MatLab, you can use a gaussian kernel on **standardized** data by doig:

$$k(\underline{u},\underline{v}) = \exp(-||\underline{u} - \underline{v}||^2)$$

Gram Matrix and PCA

We know that The PCA of a matrix is derived from it's scatter matrix of variables. In order to use the kernel trick with PCA, we must first make a scatter matrix of *Observations*. This is done by replacing $AA^\top$ with $\hat{W}$ as the scatter matrix.

Learning Outcomes

Students should now be able to:

- relate an embedding to a kernel function
- select kernel function from a library of functions
- compute a Gram matrix from given data