Math 521 Classification: Cats and Dogs

Due on Tuesday, May 15, 2018

Kristin Holmbeck and Debbie Tonne

Contents

List of Figures

Theory

Introduction

The project we present in this report involves properly classifying two data sets successfully. In this context, the data sets are images of dogs and cats, but the same ideas and algorithms can be successfully applied to other data sets, such as sound waves. Since we are working with images, some preprocessing methods will be explored to add uniformity or variance to the data sets.

Preprocessing

Image *preprocessing* typically involves filtering or computing the Fourier transform of an image prior to analysis. To this end, we will discuss some basics, beginning with filtering.

Image filtering uses a *mask* matrix on subsets of an image to perform operations. One filter example is the averaging filter: Given an $m \times n$ mask size, the mask $m_a$will be

$$m_a = \frac{1}{mn} \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}$$

where the filtering operation involves an $m \times n$ neighborhood around each pixel of the original image matrix A.

Kernel Discriminant Analysis

Before detailing the kernel classification method, we will provide an intuitive example for explaining why one might want to use KDA over LDA. First, consider the toy problem of two concentric circles of data (Figure 1).

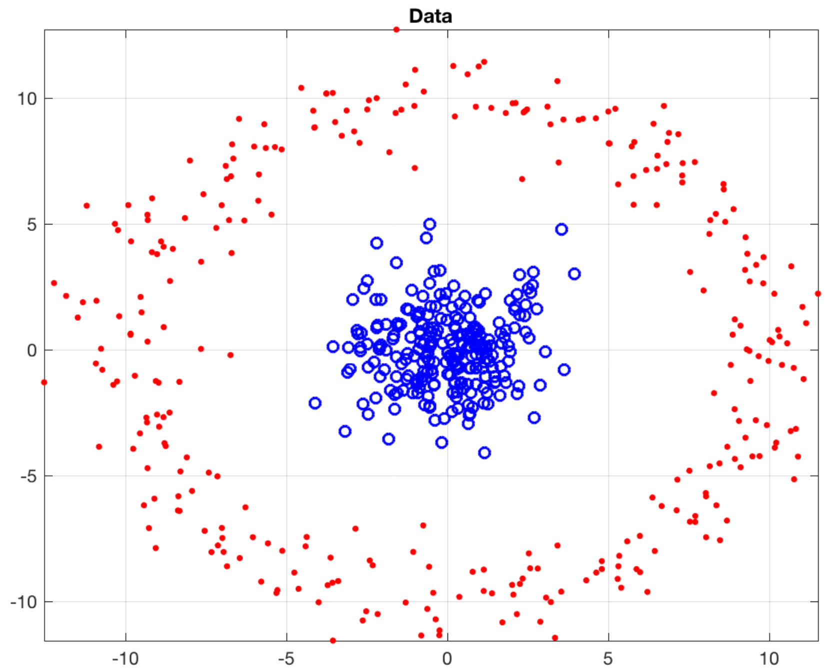

Figure 1: Concentric Data

There is no way to separate this data linearly, but the classes clearly have a defined separation. Kernel Discriminant Analysis (KDA) utilizes the ideas of LDA but with the data set X mapped onto a new *feature* space $\mathcal{F}$ where the data has a linear relationship in $\mathcal{F}$.

Following the LDA method, suppose our two-class data is given by X $X_1 \cup X_2$ where $X_1 = \{x_i\}_{i=1}^{l_1}$ and $X_2 = \{x_j\}_{j=1}^{l_2}$. Now, let Φ be a nonlinear mapping to some feature space $\mathcal{F}$, that is, we take a vector x from X and map it using $\Phi(x) \in \mathcal{F}$. From the LDA algorithm, we need to maximize

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

However, using the new space, we must maximize

$$J(w) = \frac{w^T S_B^\Phi w}{w^T S_w^\Phi w}$$

where

$$S_B^\Phi = (m_1^\Phi - m_2^\Phi)(m_1^\Phi - m_2^\Phi)^T \qquad \text{and}$$
$$S_W^\Phi = \sum_{i=1}^{2} \sum_{x \in X_i} (\Phi(x) - m_i^\Phi)(\Phi(x) - m_i^\Phi)^T$$

are the between-class and within-class scatter matrices, respectively, in the $\mathcal{F}$ space, and $m_i = \frac{1}{l_i} \sum_{j=1}^{l_i} \Phi(x_j^{(i)})$, the mean of the i^{th} class.

Kernel Trick

The kernel trick boils down to using a linear classifier to solve a non-linear problem.

A feature map is a map $\Phi : X \to \mathcal{F}$, where $\mathcal{F}$ is what we call the feature space. Every feature map defines a kernel

$$K(x, y) = \langle \Phi(x), \Phi(y) \rangle$$

where K is clearly symmetric and positive-definite. In the context of linear algebra, the kernel is the space equivalent to the null space. In the statistical context, the kernel is used as a measure of similarity. In particular, the kernel function k defines the distribution of similarities of points around a given point x, $k(x, y)$ denotes the similarity of point x with another given point y.

Explicitly computing the mappings of a function $\Phi(x)$ onto $\mathcal{F}$ can become intractable quick. To that end, we instead compute the inner products between the images (images in the linear algebra sense) of all pairs of data in the feature space.

For any $x, \hat{x}$ in X, some kernel functions $k(x, \hat{x})$ can be expressed as an inner product in another space V. In other words, $k : X \times X \to \mathbb{R}$ and $\Phi : X \to V$.

$$k(x, \hat{x}) = \langle \Phi(x), \Phi(\hat{x}) \rangle_V$$

KDA on concentric circle data

Let us revisit the concentric data problem from Figure 1, and compare the classification of LDA to KDA. For LDA, we obtain the failed classification (Figure 2) and the projection vector along with the data in Figure 3.

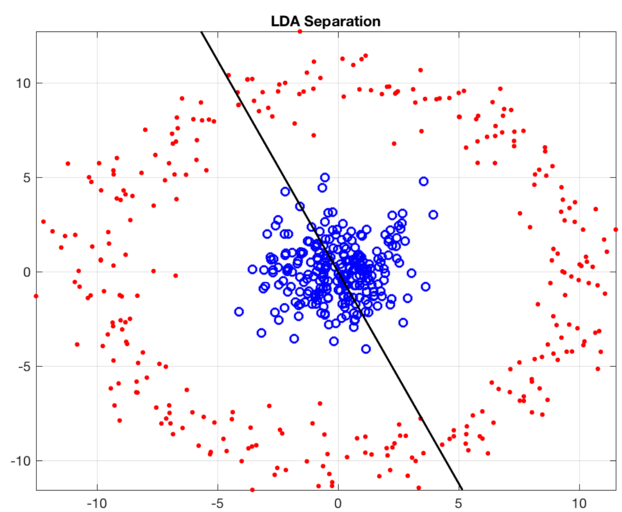

Figure 2: Figure 1 data with LDA projection vector

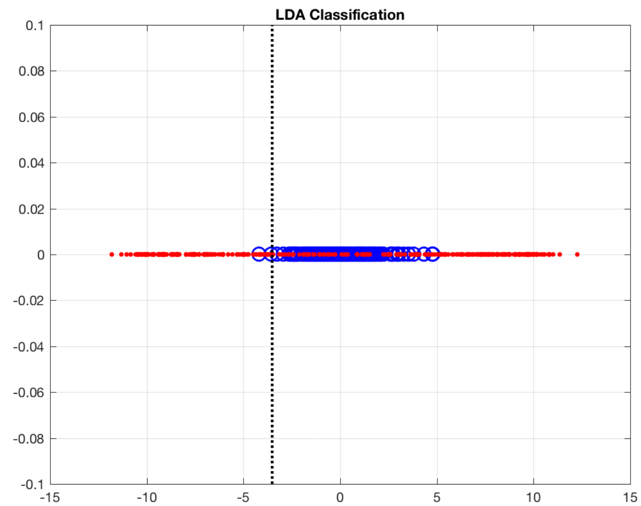

Figure 3: Figure 1 LDA classification

As described above, the kernel conversion is kind of tricky, and thus we cannot plot the equivalent projection vector. However, the successful classification is shown in Figure 4.

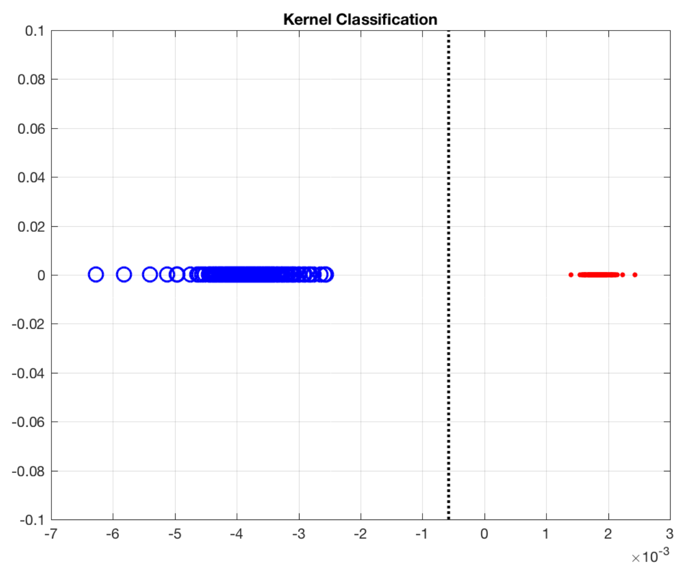

Figure 4: Figure 1 KDA classification

KDA on parabolic data

Another data set is shown in Figure 5. Again, this is easily separable, but it's clear that the separation is nonlinear. Again, for LDA, we obtain the failed classification (Figure 6) and the successful classification in Figure 8.

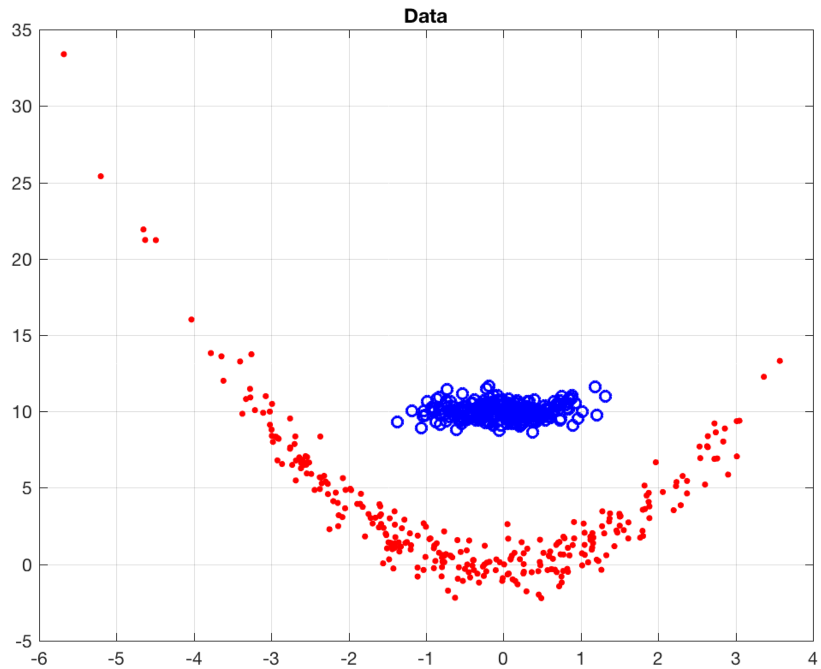

Figure 5: Parabolically-separable data

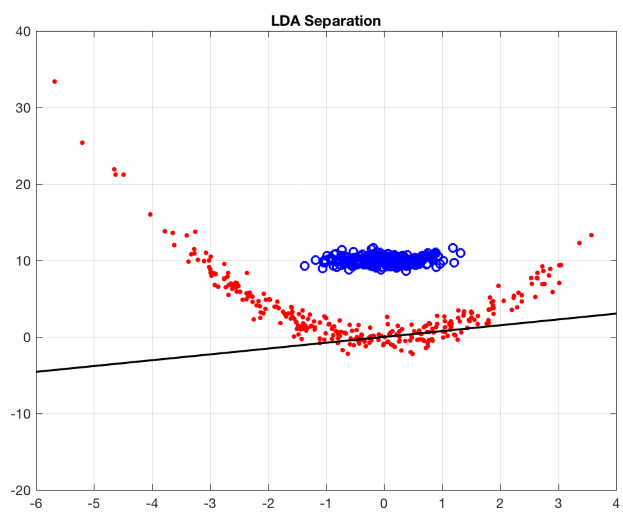

Figure 6: Figure 5 data with LDA projection vector

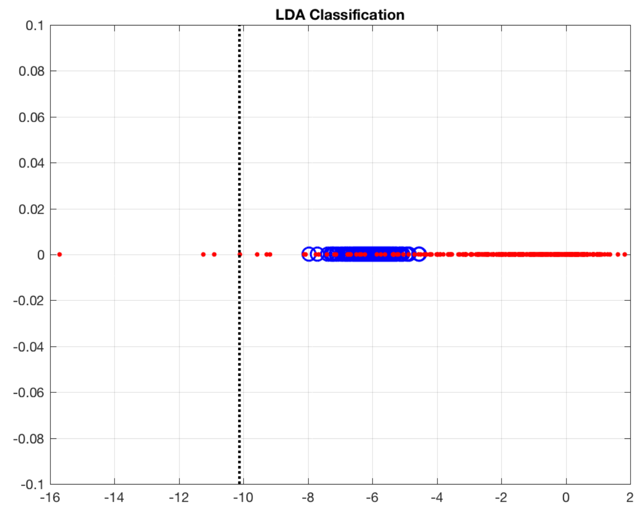

Figure 7: Figure 5 LDA classification

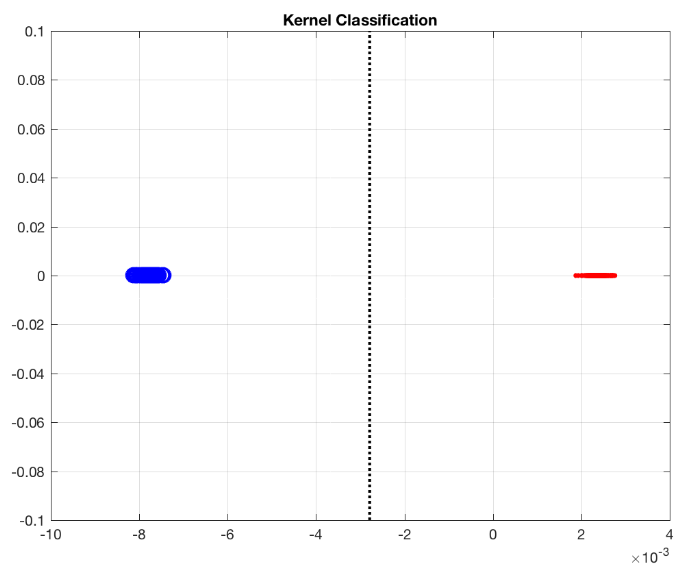

Figure 8: Figure 5 KDA classification

Classification

Singular Values

The Singular Value Decomposition (SVD) is an important first step in classification.

Results

Dogs and Cats

Classification Types

Code

References

[1] Chang, Jen-Mei. *Matrix Methods for Geometric Data Analysis and Recognition*. 2014.

[2] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Muller. Fisher discriminant analysis with kernels. In *Proc. IEEE Neural Networks for Signal Processing Workshop*, pages 4148. IEEE Computer Society Press, 1999.