Novelty Detection

Cate Welch

May 14, 2015

Contents

1 Introduction **2**
- 1.1 The Four Fundamental Subspaces . . . 2
- 1.2 The Spectral Theorem . . . 4
- 1.3 The Singular Value Decomposition . . . 5

2 The Principal Components Analysis (PCA) **8**
- 2.1 The Best Basis . . . 9
- 2.2 Connections to the SVD . . . 12
- 2.3 Computing the Proper Rank . . . 13
- 2.4 A Second Argument That This is the Best Basis . . . 14
- 2.5 Application: Linear Classifier and PCA . . . 15

3 The Kernel PCA **18**
- 3.1 Important Definitions . . . 18
- 3.2 Formulating the Standard PCA With Dot Products . . . 20
- 3.3 The Kernel PCA Algorithm . . . 22
- 3.4 Novelty Detection Using the Kernel PCA . . . 22

4 Application: Spiral Results and the Kernel PCA **26**
- 4.1 Spiral Results . . . 26
- 4.2 The Kernel PCA . . . 27

5 Appendix **29**
- 5.1 Mean Subtracting in High-Dimensional Space . . . 29
- 5.2 The Classifier in MATLAB . . . 29

Abstract

Given a set of data, if the data is linear, the Principal Components Analysis (PCA) can find the best basis for the data by reducing the dimensionality. The norm of the reconstruction error determines what is normal and what is novel for the PCA. When the data is non-linear, the Kernel Principal Components Analysis, an extension of the PCA, projects the data into a higher dimensional the feature space, so that it is linearized and the PCA can be performed. For the kernel PCA, the maximum value of a normal data point separates what is novel and what is normal. The accuracy of novelty detection using breast cancer cytology was compared using the PCA and Kernel PCA methods.

1 Introduction

Novelty detection determines, from a set of data, which points are considered normal and which ones are novel. Given a set of data, we train a set of normal observations and determine a distance from this normal region that would classify a point as novel. That is, if an observation has a value that is larger than the decided normal value, it is considered novel.

There are many applications of novelty detection such as pattern recognition and disease detection. In our case, we used novelty detection to detect whether a given breast cancer tumor is malignant or benign.

In order to perform novelty detection, we change the basis, by reducing its size, so that it is easier to work with. For example, if the data is linear, the Principal Component Analysis (PCA) is used to find the best basis for the data. We can often find a lower dimensional subspace to work in that encapsulates most of the data so that the data is more manageable. However, many times the data is not linear and we cannot use the PCA directly. The kernel PCA is used for nonlinear data, and instead of reducing the dimension of the subspace, the kernel PCA increases the dimension of the subspace (called the feature space) where the data behaves linearly. In this higher dimension, since the data now acts linearly, the PCA can be used. Once in the appropriate subspace and we have found the best basis for our data, we can define the threshold of what is normal and what is novel within our data. Before going into the PCA and kernel PCA, there are some important concepts and theorems that are explained below.

1.1 The Four Fundamental Subspaces

Given any $m \times n$ matrix A, consider the mapping $A : \mathbb{R}^n \to \mathbb{R}^m$,

$$\mathbf{x} \to A\mathbf{x} = \mathbf{y},$$

where $\mathbf{x}$ and $\mathbf{y}$ are column vectors.

1. The **row space** of A is a subspace of $\mathbb{R}^n$ formed by taking all possible linear combinations of the rows of A. That is,

$$\text{Row}(A) = \{\mathbf{x} \in \mathbb{R}^n | x = A^T\mathbf{y},\ y \in \mathbb{R}^m\}.$$

2. The **null space** of A is a subspace of $\mathbb{R}^n$ formed by

$$\text{Null}(A) = \{\mathbf{x} \in \mathbb{R}^n | A\mathbf{x} = \mathbf{0}\}.$$

3. The **column space** of A is a subspace of $\mathbb{R}^m$ formed by taking all possible linear combinations of the columns of A. That is,

$$\text{Col}(A) = \{\mathbf{y} \in \mathbb{R}^m | y = A\mathbf{x} \in \mathbb{R}^n\}.$$

   The column space is also the image of the mapping. For all $\mathbf{x}$, $A\mathbf{x}$ is a linear combination of the columns of A:

$$A\mathbf{x} = x_1\mathbf{a_1} + x_2\mathbf{a_2} + ... + x_n\mathbf{a_n}.$$

4. The **null space** of A^T is

$$\text{Null}(A^T) = \{\mathbf{y} \in \mathbb{R}^m | A^T\mathbf{y} = \mathbf{0}\}.$$

Theorem 1 *Let A be an $m \times n$ matrix. Then the null space of A is orthogonal to the row space of A and the null space of A^T is orthogonal to the column space of A.*

Proof of the first statement:
Let A be an $m \times n$ matrix. Null(A) is defined as the set of all $\mathbf{x}$ for which

$$A\mathbf{x} = \mathbf{0}.$$

Organizing this in terms of the rows of A, we have

$$\sum_{k=1}^{n} \mathbf{a}_{ik} \cdot \mathbf{x}_k = \begin{pmatrix}\mathbf{a}_{11} & \mathbf{a}_{12} \dots \mathbf{a}_{1n}\end{pmatrix} \cdot \begin{pmatrix}\mathbf{v}_1 & \mathbf{v}_2 \dots \mathbf{v}_n\end{pmatrix}.$$

Therefore, every row of A is perpendicular (orthogonal) to every vector in the null space of A. Since the rows of A span the row space, Null(A) must be the orthogonal complement of Row(A). In other words, the row space of a matrix is orthogonal to the null space because $A\mathbf{x} = \mathbf{0}$ means the dot product of $\mathbf{x}$ with each row of A is 0. But then the product of $\mathbf{x}$ with any combination of rows of A must be 0. The second statement follows a similar proof.

Dimensions of the Subspaces
Given a matrix A that is $m \times n$ with rank k, the dimensions of the four subspaces are:

- $\dim(\text{Row}(A)) = k$
- $\dim(\text{Null}(A)) = n - k$
- $\dim(\text{Col}(A)) = k$
- $\dim(\text{Null}(A^T)) = m - k$

1.2 The Spectral Theorem

A **symmetric matrix** is one in which $A = A^T$. The Spectral Theorem outlines some of the properties of symmetric matrices.

Theorem 2 ***The Spectral Theorem:*** *If A is an $n \times n$ symmetric matrix, then A is orthogonally diagonalizable, with $D = diag(\lambda_1, \lambda_2, ..., \lambda_n)$. That is, if V is the matrix whose columns are orthonormal eigenvectors of A, then*

$$A = VDV^T.$$

Corollary 1 *A has n real eigenvalues (counting multiplicity).*

Corollary 2 *For each distinct λ, the algebraic and geometric multiplicities are the same. Where the algebraic multiplicity is the number of times λ is a repeated root of the characteristic polynomial and the geometric multiplicity is the dimension of E_λ. That is, the number of corresponding eigenvectors.*

Corollary 3 *The eigenspaces are mutually orthogonal- both for distinct eigenvalues, and each E_λ has an orthonormal basis. That is, a subset $v_1, v_2, ..., v_n$ of a vector space V, with the inner product $\langle\ ,\ \rangle$, is orthonormal if $\langle v_i, v_j \rangle = 0$ when $i \neq j$, which means that all vectors are mutually perpendicular. Also, $\langle v_i, v_i \rangle = 1$.*

In summary, the Spectral Theorem says that if a matrix is real and symmetric, then its eigenvectors form an orthonormal basis for $\mathbb{R}^n$.

The Spectral Decomposition: Since A is orthogonally diagonalizable, then by the Spectral Theorem, $A = VDV^T$:

$$A = (\mathbf{q_1}\ \mathbf{q_2}\ \dots\ \mathbf{q_n}) \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{pmatrix} \begin{pmatrix} \mathbf{q}_1^T \\ \mathbf{q}_2^T \\ \vdots \\ \mathbf{q}_n^T \end{pmatrix}$$

so that:

$$A = (\lambda_1\mathbf{q}_1\ \lambda_2\mathbf{q}_2\ \dots\ \lambda_n\mathbf{q}_n) \begin{pmatrix} \mathbf{q}_1^T \\ \mathbf{q}_2^T \\ \vdots \\ \mathbf{q}_n^T \end{pmatrix}$$

so finally:

$$A = \lambda_1\mathbf{q}_1\mathbf{q}_1^T + \lambda_2\mathbf{q}_2\mathbf{q}_2^T + \cdots + \lambda_n\mathbf{q}_n\mathbf{q}_n,$$

where the $\mathbf{q}_i$ are the eigenvectors and the λ_i are the eigenvalues. That is, A is a sum of n rank one matrices, each of which, by definition, is a projection matrix.

1.3 The Singular Value Decomposition

The Singular Value Decomposition is useful because it can be used on any matrix, not just a symmetric matrix like the Spectral Theorem. It finds the rank of the matrix and gives orthonormal bases for all four matrix subspaces. For the remainder of this paper, SVD will refer to the Singular Value Decomposition.

Assume A is an $m \times n$ matrix. That is, multiplication by A maps $\mathbb{R}^n$ to $\mathbb{R}^m$ by $\mathbf{x} \to A\mathbf{x}$. Although A is not symmetric, A^TA is an $n \times n$ matrix and symmetric since $(A^TA)^T = A^TA^{TT} = A^TA$, so the Spectral Theorem can be applied. From the Spectral Theorem, we know that A^TA is orthogonally diagonalizable. Thus, for eigenvalues $\{\lambda_i\}_{i=1}^n$ and orthonormal eigenvectors $\mathbf{v}_i$, $V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$:

$$A^TA = VDV^T$$

where D is the diagonal matrix with λ_i along the diagonal.

Before stating the SVD, it is important to define the singular values of A. The singular values are $\sigma_i = \sqrt{\lambda_i}$ where λ_i is an eigenvalue of A^TA. These are always real because, by the Spectral Theorem, the eigenvalues of a symmetric matrix are real. Also, the matrix A^TA is positive semi-definite (a symmetric $n \times n$ matrix) which is a matrix such that its eigenvalues are all non-negative.

It is also important to note that any diagonal matrix of eigenvalues or singular values of A^TA will be ordered from high to low. That is, assume $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ and therefore, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$.

The Singular Value Decomposition: Let A be any $m \times n$ matrix of rank r. Then A can be factored as:

$$A = U\Sigma V^T$$

where:

- The columns of U are constructed by the eigenvectors, $\mathbf{u}_i$, of AA^T. It is an orthogonal $m \times m$ matrix.
- $\Sigma = diag(\sigma_1, \ldots, \sigma_n)$ where σ_i is the i^{th} singular value of A. It is a diagonal $m \times n$ matrix.
- The columns of V are constructed by the eigenvectors, $\mathbf{v}_i$, of A^TA. It is an orthogonal $n \times n$ matrix.

The $\mathbf{u}_i$'s are called the **left singular vectors** and the $\mathbf{v}_i$'s are the **right singular vectors**.

Simple example of the SVD

Problem: Construct a singular value decomposition of $A = \begin{bmatrix} 4 & 11 & 14 \\ 8 & 7 & -2 \end{bmatrix}$.

By direct computation from the characteristic equation, the eigenvalues of A^TA are $\lambda_1 = 360$, $\lambda_2 = 90$, and $\lambda_3 = 0$.

Step 1: Find V. Recall that V is constructed by the eigenvectors of A^TA.

The corresponding eigenvectors are $\mathbf{v}_1 = \begin{bmatrix} \frac{1}{3} \\ \frac{2}{3} \\ \frac{2}{3} \end{bmatrix}$, $\mathbf{v}_2 = \begin{bmatrix} \frac{-2}{3} \\ \frac{-1}{3} \\ \frac{2}{3} \end{bmatrix}$, $\mathbf{v}_3 = \begin{bmatrix} \frac{2}{3} \\ \frac{-2}{3} \\ \frac{1}{3} \end{bmatrix}$.

So $V^T = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} & \frac{2}{3} \\ \frac{-2}{3} & \frac{-1}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{-2}{3} & \frac{1}{3} \end{bmatrix}$.

Since the eigenvalues are 360, 90, and 0, the corresponding singular values are

$$\sigma_1 = \sqrt{360} = 6\sqrt{10}, \quad \sigma_2 = \sqrt{90} = 3\sqrt{10}, \quad \text{and} \quad \sigma_3 = 0.$$

So $\Sigma = \begin{bmatrix} 6\sqrt{10} & 0 & 0 \\ 0 & 3\sqrt{10} & 0 \end{bmatrix}$.

When A has rank r, the first r columns of U are the normalized vectors obtained from $A\mathbf{v}_1, \ldots, A\mathbf{v}_r$.

We know that $||A\mathbf{v}_1|| = \sigma_1$ and $||A\mathbf{v}_2|| = \sigma_2$. Thus,

$$\mathbf{u}_1 = \frac{1}{\sigma_1} A\mathbf{v}_1 = \frac{1}{6\sqrt{10}} \begin{bmatrix} 18 \\ 6 \end{bmatrix} = \begin{bmatrix} \frac{3}{\sqrt{10}} \\ \frac{1}{\sqrt{10}} \end{bmatrix},$$

$$\mathbf{u}_2 = \frac{1}{\sigma_2} A\mathbf{v}_2 = \frac{1}{3\sqrt{10}} \begin{bmatrix} 3 \\ -9 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{10}} \\ \frac{-3}{\sqrt{10}} \end{bmatrix}.$$

Thus, the SVD of A is:

$$A = \begin{bmatrix} \frac{3}{\sqrt{10}} & \frac{1}{\sqrt{10}} \\ \frac{1}{\sqrt{10}} & \frac{-3}{\sqrt{10}} \end{bmatrix} \begin{bmatrix} 6\sqrt{10} & 0 & 0 \\ 0 & 3\sqrt{10} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{3} & \frac{2}{3} & \frac{2}{3} \\ \frac{-2}{3} & \frac{-1}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{-2}{3} & \frac{1}{3} \end{bmatrix}.$$

Another way to express the SVD is: Let $A = U\Sigma V^T$ be the SVD of A with rank r. Then:

$$A = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T.$$

In this form, we can see that A can be approximated by the sum of rank one matrices.

When m or n is very large, we might not want to use all of U and V. In this case, we would use the **reduced SVD**:

$$A = \tilde{U}\tilde{\Sigma}\tilde{V}^T$$

where A is an $m \times n$ matrix with rank r, $\tilde{U}$ is $m \times r$ with orthogonal columns, $\tilde{\Sigma}$ is an $r \times r$ square matrix, and $\tilde{V}$ is $n \times r$.

There a several different ways to compute the proper rank r for the reduced SVD, listed in 2.3.

Theorem 3 *The non-zero eigenvalues of A^TA and AA^T are the same for an $m \times n$ matrix A.*

Proof:
For any eigenvalue λ of AA^T, we have $AA^T\mathbf{x} = \lambda\mathbf{x}$ with $\mathbf{x} \neq \mathbf{0}$. That is, multiplying through by A^T, we have where we have set $\mathbf{y} = A^T\mathbf{x}$. Thus, provided that $\mathbf{y} \neq \mathbf{0}$, any eigenvalue of AA^T is also an eigenvalue of A^TA. However, if $\mathbf{y} = \mathbf{0}$, then $\lambda\mathbf{x} = AA^T\mathbf{x} = A\mathbf{y} = A\mathbf{0} = \mathbf{0}$, and since $\mathbf{x} \neq \mathbf{0}$, we have $\lambda = 0$. Consequently, this analysis only holds if we are considering the non-zero eigenvalues of AA^T.

For any eigenvalue λ of A^TA, we have $A^TA\mathbf{x} = \lambda\mathbf{x}$ with $\mathbf{x} \neq \mathbf{0}$. That is, multiplying through by A, we have

$$A(A^TA)\mathbf{x} = \lambda A\mathbf{x} \Rightarrow (AA^T)(A\mathbf{x}) = \lambda(A\mathbf{x}) \Rightarrow AA^T\mathbf{y} = \lambda\mathbf{y},$$

where we have set $\mathbf{y} = A\mathbf{x}$. Thus, provided that $\mathbf{y} \neq \mathbf{0}$, any eigenvalue of A^TA is also an eigenvalue of AA^T. However, if $\mathbf{y} = \mathbf{0}$, then $\lambda\mathbf{x} = A^TA\mathbf{x} = A^T\mathbf{y} = A^T\mathbf{0} = \mathbf{0}$, and since $\mathbf{x} \neq \mathbf{0}$, we have $\lambda = 0$. Consequently, this analysis only holds if we are considering the non-zero eigenvalues of A^TA.

Thus, the matrices AA^T and A^TA have the same positive eigenvalues, as required.

To compute the four subspaces of A: Let $A = U\Sigma V^T$ be the SVD of A with rank r and the singular values are ordered highest to lowest:

a) A basis for the columnspace of A, Col(A) is $\{\mathbf{u}_i\}_{i=1}^{r}$.

b) A basis for the nullspace of A, Null(A) is $\{\mathbf{v}_i\}_{i=r+1}^{n}$.

c) A basis for the rowspace of A, Row(A) is $\{\mathbf{v}_i\}_{i=1}^{r}$.

d) A basis for the nullspace of A^T, Null(A^T) is $\{\mathbf{u}_i\}_{i=r+1}^{m}$.

2 The Principal Components Analysis (PCA)

The PCA finds the best basis for the data by finding a lower dimensional subspace that encapsulates most of the data. The principal components of the data end up being the eigenvectors of the covariance matrix, C, and then these vectors are used as the basis vectors for the data.

Getting Started

Assume we have a set of data $\left\{\mathbf{x^{(1)}}, \ldots, \mathbf{x^{(p)}}\right\}$ where each $\mathbf{x^{(i)}} \in \mathbb{R}^n$. Also assume that the columns of Φ are orthonormal. Let X be the $p \times n$ matrix formed from $X = \left[\mathbf{x^{(1)}} \ldots \mathbf{x^{(p)}}\right]^T$.

Given the basis $\Phi = \left[\boldsymbol{\phi}_1 \boldsymbol{\phi}_2 \ldots \boldsymbol{\phi}_n\right]$ in $\mathbb{R}^n$, we can expand any $\mathbf{x} \in \mathbb{R}^n$ in terms of

$$\mathbf{x} = \sum_{k=1}^{n} \alpha_k \boldsymbol{\phi}_k = \Phi\alpha,$$

where $\alpha_k = \boldsymbol{\phi}_k^T \mathbf{x}$. Therefore, $\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\phi}_1^T \\ \boldsymbol{\phi}_2^T \\ \vdots \\ \boldsymbol{\phi}_n^T \end{bmatrix} \mathbf{x} = \Phi^T \mathbf{x}$.

Substituting this for $\boldsymbol{\alpha}$, we get

$$\mathbf{x} = \Phi\Phi^T \mathbf{x}.$$

When Φ is a square matrix, $\Phi\Phi^T$ is the identity. Otherwise,

$$\begin{aligned} \mathbf{x} &= \Phi\Phi^T \mathbf{x} \\ \mathbf{x} &= \boldsymbol{\phi}_1\boldsymbol{\phi}_1^T\mathbf{x} + \boldsymbol{\phi}_2\boldsymbol{\phi}_2^T\mathbf{x} + \cdots + \boldsymbol{\phi}_n\boldsymbol{\phi}_n^T\mathbf{x}, \end{aligned}$$

which represents a projection of $\mathbf{x}$ into the subspace spanned by the columns of Φ.

If we are interested in the subspace spanned by the first k columns of Φ, where $k < n$, we can write

$$\mathbf{x} = \sum_{j=1}^{k} \alpha_j \boldsymbol{\phi}_j + \sum_{j=k+1}^{n} \alpha_j \boldsymbol{\phi}_j.$$

The second term is the error in using k basis vectors to express $\mathbf{x}$:

$$\mathbf{x}_{\text{err}} = \sum_{j=k+1}^{n} \alpha_j \boldsymbol{\phi}_j,$$

therefore,

$$||\mathbf{x}_{\text{err}}||^2 = \alpha_{k+1}^2 + \cdots + \alpha_n^2,$$

by the Pythagorean Theorem on vectors.

This second term, $||\mathbf{x}_{\text{err}}||^2 = \alpha_{k+1}^2 + \cdots + \alpha_n^2$, is known as the **reconstruction error**. Finding the best basis requires us to minimize this reconstruction error, as shown in the following section.

2.1 The Best Basis

To find the best basis, assume:

1. The data has been mean subtracted.
2. The best basis should be orthonormal.
3. The data does not take up all of $\mathbb{R}^n$. It lies in some linear subspace.
4. If we have p data points, the error using k columns will be defined by the mean square error of
$$\text{Error} = \frac{1}{p}\sum_{j=1}^{p} ||\mathbf{x}_{\text{err}}^{(j)}||^2,$$
where $||\mathbf{x}_{\text{err}}^{(j)}||^2$ is defined above.

Finding the Best Basis

Before finding the best basis, we define the covariance matrix, C, of X. The covariance between coordinate i and coordinate j is

$$C_{ij} = \frac{1}{p}\sum_{n=1}^{p} x_i^{(n)} x_j^{(n)}.$$

Since $x_i^{(n)}$ is the i^{th} "row" of $\mathbf{x}^{(n)}$ and $x_j^{(n)}$ is the i^{th} "column" of $\mathbf{x}^{(n)}$, the covariance matrix can be expressed as:

$$C = \frac{1}{p}\sum_{p=1}^{N} \mathbf{x}^{(n)}\left(\mathbf{x}^{(n)}\right)^T.$$

Thus, if the matrix X stores the data in columns so that X is $n \times p$,

$$C = \frac{1}{p}X^T X.$$

Now to find the best basis, we expand the mean square error term from the fourth assumption above,

$$\begin{aligned}
\frac{1}{p}\sum_{i=1}^{p}||\mathbf{x}_{\text{err}}^{(i)}||^2 &= \frac{1}{p}\sum_{i=1}^{p}\left(\sum_{j=k+1}^{n}\left(\alpha_j^{(i)}\right)^2\right), \quad \text{using } \alpha_k = \boldsymbol{\phi}_k^T\mathbf{x} \text{ as above,} \\
&= \frac{1}{p}\sum_{i=1}^{p}\left(\sum_{j=k+1}^{n}\boldsymbol{\phi}_j^T\mathbf{x}^{(i)}\left(\mathbf{x}^{(i)}\right)^T\boldsymbol{\phi}_j\right) \\
&= \sum_{j=k+1}^{n}\left(\frac{1}{p}\sum_{i=1}^{p}\boldsymbol{\phi}_j^T\mathbf{x}^{(i)}\left(\mathbf{x}^{(i)}\right)^T\boldsymbol{\phi}_j\right) \\
&= \sum_{j=k+1}^{n}\boldsymbol{\phi}_j^T\left[\frac{1}{p}\sum_{i=1}^{p}\mathbf{x}^{(i)}\left(\mathbf{x}^{(i)}\right)^T\right]\boldsymbol{\phi}_j, \quad \text{using the definition of } C, \\
&= \sum_{j=k+1}^{n}\boldsymbol{\phi}_j^T C\boldsymbol{\phi}_j.
\end{aligned}$$

Thus, to minimize the mean squared error on the k-term expansion $\left\{\boldsymbol{\phi}_i\right\}_{i=1}^{k}$, we need to minimize

$$\sum_{j=k+1}^{n}\boldsymbol{\phi}_j^T C\boldsymbol{\phi}_j,$$

where the $\boldsymbol{\phi}_i$ form an orthonormal set.

One way to minimize the mean squared error on $\left\{\boldsymbol{\phi_i}\right\}_{i=1}^{k}$ is to find the best 1-dimensional subspace and then to reduce the dimensionality of the data by 1, then find the best 1-dimensional subspace for the remaining data, and so on. To do this, we find:

$$\min_{\boldsymbol{\phi}_2,\ldots,\boldsymbol{\phi}_n}\sum_{j=2}^{n}\boldsymbol{\phi}_j^T C\boldsymbol{\phi}_j$$

which is equivalent to

$$\max_{\boldsymbol{\phi}_1\neq 0}\frac{\boldsymbol{\phi}_1^T C\boldsymbol{\phi}_1}{\boldsymbol{\phi}_1^T\boldsymbol{\phi}_1}$$

since the basis is orthonormal.

Theorem 4 *The maximum (over non-zero $\boldsymbol{\phi}$) of*

$$\frac{\boldsymbol{\phi}^T C\boldsymbol{\phi}}{\boldsymbol{\phi}^T\boldsymbol{\phi}}$$

where C is the covariance matrix of X occurs at $\boldsymbol{\phi} = \mathbf{v}_1$, the first eigenvector of C, where λ_1 is the largest eigenvalue of C.

This shows how to find the best one dimensional basis vector, but then we need to find the next one. The next basis vector satisfies:

$$\max_{\phi \perp \mathbf{v}_1} \frac{\boldsymbol{\phi}^T C \boldsymbol{\phi}}{\boldsymbol{\phi}^T \boldsymbol{\phi}}$$

which will be the second eigenvector of C. This is a result of the Spectral Theorem that said that the eigenvectors are orthogonal.

Proof:
Let $\{\lambda_i\}_{i=1}^n$ and $\{\mathbf{v}_i\}_{i=1}^n$ be the eigenvalues and eigenvectors of the covariance matrix, C. Also, let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. Since C is symmetric, by the Spectral Theorem, $C = VDV^T$, so C can be written as

$$C = \lambda \mathbf{v}_1 \mathbf{v}_1^T + \cdots + \lambda_n \mathbf{v}_n \mathbf{v}_n^T = VDV^T,$$

where $VV^T = V^T V = I$ because V and V^T are square, orthogonal matrices.

We can write any vector $\boldsymbol{\phi}$ as:

$$\boldsymbol{\phi} = a_1 \mathbf{v}_1 + \cdots + a_n \mathbf{v}_n = V\mathbf{a}.$$

We also know that

$$\boldsymbol{\phi}^T \boldsymbol{\phi} = a_1^2 + a_2^2 + \cdots + a_n^2$$

because

$$(V\mathbf{a})^T V\mathbf{a} = \mathbf{a}^T V^T V\mathbf{a}.$$

We know $V^T V = I$, so

$$\boldsymbol{\phi}^T \boldsymbol{\phi} = \mathbf{a}^T \mathbf{a} = a_1^2 + a_2^2 + \cdots + a_n^2.$$

Also,

$$\boldsymbol{\phi}^T C \boldsymbol{\phi} = \lambda_1 a_1^2 + \lambda_2 a_2^2 + \cdots + \lambda_n a_n^2.$$

This is because, using $C = VDV^T$ and $\boldsymbol{\phi} = V\mathbf{a}$,

$$\begin{aligned}(V\mathbf{a})^T C (V\mathbf{a}) =& \mathbf{a}^T V^T V D V^T (V\mathbf{a}) = \mathbf{a}^T (V^T V) D (V^T V)\mathbf{a} && (1)\\ =& \mathbf{a}^T D\mathbf{a} = \lambda_1 a_1^2 + \lambda_2 a_2^2 + \cdots + \lambda_n a_n^2. && (2)\end{aligned}$$

Subbing these into

$$\frac{\boldsymbol{\phi}^T C \boldsymbol{\phi}}{\boldsymbol{\phi}^T \boldsymbol{\phi}},$$

we get

$$\frac{\boldsymbol{\phi}^T C \boldsymbol{\phi}}{\boldsymbol{\phi}^T \boldsymbol{\phi}} = \frac{\lambda_1 a_1^2 + \lambda_2 a_2^2 + \cdots + \lambda_n a_n^2}{a_1^2 + a_2^2 + \cdots + a_n^2}.$$

Since we are maximizing $\dfrac{\boldsymbol{\phi}^T C \boldsymbol{\phi}}{\boldsymbol{\phi}^T \boldsymbol{\phi}}$ and λ_1 is the largest eigenvalue, we get

$$\frac{\boldsymbol{\phi}^T C \boldsymbol{\phi}}{\boldsymbol{\phi}^T \boldsymbol{\phi}} = \frac{\lambda_1 a_1^2 + \lambda_2 a_2^2 + \cdots + \lambda_n a_n^2}{a_1^2 + a_2^2 + \cdots + a_n^2} \leq \lambda_1.$$

To have this quantity equal to λ_1, then, since $\mathbf{v}_1$ is the eigenvector of C corresponding to λ_1,

$$\frac{\boldsymbol{\phi}^T C \boldsymbol{\phi}}{\boldsymbol{\phi}^T \boldsymbol{\phi}} = \frac{\lambda_1 a_1^2 + \lambda_2 a_2^2 + \cdots + \lambda_n a_n^2}{a_1^2 + a_2^2 + \cdots + a_n^2} = \lambda_1$$

if and only if $\boldsymbol{\phi} = \mathbf{v}_1$.

This shows how to find the best one-dimensional basis vector, so the next basis vector satisfies:

$$\max_{\boldsymbol{\phi} \perp \mathbf{v}_1} \frac{\boldsymbol{\phi}^T C \boldsymbol{\phi}}{\boldsymbol{\phi}^T \boldsymbol{\phi}},$$

which by the same argument as above will be the second eigenvector of C. Now we conclude with the Best Basis Theorem:

Theorem 5 ***The Best Basis Theorem:*** *Suppose that:*

- *X is a $p \times n$ matrix of p points in $\mathbb{R}^n$.*
- *X_m is the $p \times n$ matrix of mean subtracted data.*
- *C is the covariance matrix of X, $C = \frac{1}{p} X_m^T X_m$.*

Then the best (in terms of the mean squared error) orthonormal k-dimensional basis is given by the leading k eigenvectors of C, for any k.

In conclusion, the Principal Components of the data are the eigenvectors of the covariance, which are used as the basis vectors. Because we have found the optimal basis for our data, we can then perform Novelty Detection.

2.2 Connections to the SVD

Recall that the (reduced) SVD of the matrix X is $X = U\Sigma V^T$, so that Σ is $k \times k$ and k is the rank of X. Using this in the formula for the covariance matrix, C, and $U^T U = I$,

$$C = \frac{1}{p} X^T X = \frac{1}{p} V \Sigma U^T U \Sigma V^T = V \left(\frac{1}{p} \Sigma^2 \right) V^T.$$

Comparing this to the Best Basis Theorem, the best basis vectors for the rowspace of X are the right singular vectors for X. Similarly, if we formulate the other covariance matrix (thinking of X, which is $p \times n$, as n points in $\mathbb{R}^p$):

$$C_2 = \frac{1}{n} X X^T = \frac{1}{n} U \Sigma V^T V \Sigma U^T = U \left(\frac{1}{n} \Sigma^2 \right) U^T.$$

Comparing this to the Best Basis Theorem, the best basis vectors for the columnspace of X are the left singular vectors for X. Also, using the SVD, knowledge of the best basis

vectors of the rowspace gives the best basis vectors of the columnspace (and vice-versa) by the relationships:

$$X\mathbf{v}_i = \sigma_i \mathbf{u}_i \quad \text{and} \quad X^T\mathbf{u}_i = \sigma_i \mathbf{v}_i$$

where σ_i can be found from the eigenvalues of the covariance matrix:

$$\lambda_i = \frac{\sigma_i^2}{p} \qquad \text{or} \qquad \lambda_i = \frac{\sigma_i^2}{n}.$$

2.3 Computing the Proper Rank

When we do not have any $\lambda_i = 0$ to drop for the rank, these are some ways to determine the proper rank:

- If there is a large gap in the graph of the eigenvalues (as in a large jump in the value of the eigenvalues), we will use that as our rank (the index of the eigenvalue just prior to the gap). That is, say $\lambda_5 = 50$, $\lambda_6 = 0.1$, and $\lambda_7 = 0.01$. There is a large gap between the 5^{th} and 6^{th} eigenvalues, so we can drop the eigenvalues after λ_5 and use the first five.

- It is better to use the scaled eigenvalues as we are interested in looking at their relative sizes. Denote $\tilde{\lambda}_i$ as the unscaled eigenvalues of C. Then:

  $$\lambda_i = \frac{\tilde{\lambda}_i}{\sum_{j=1}^{n} \tilde{\lambda}_j}$$

  are the normalized eigenvalues of C. It is also important to note that λ_i are nonzero and sum to one.

- The normalized eigenvalue λ_i can be interpreted as the percentage of the total variance captured by the i^{th} eigenspace (the span of the i^{th} eigenvector). This is also referred to as the percentage of the total energy captured by the i^{th} eigenspace. We can think of the dimension as a function of the total energy (or total variance) that we want to encapsulate in a k-dimensional subspace.

  The principal component dimension (also known as the Karhunen-Loeve, KL) is the number of eigenvectors required to explain a given percentage of the energy:

  $$\text{KLD}_d = k$$

  where k is the smallest integer so that:

  $$\lambda_1 + \cdots + \lambda_k \geq d$$

  and λ_i are the normalized, ordered eigenvalues.

  The value of d depends on the problem. If there is a lot of noise in the data, we may choose $d = 0.6$ whereas if the graph of the data is smooth with minimal noise, we may choose $d = 0.99$.

2.4 A Second Argument That This is the Best Basis

Below is another way to show that the principal component basis for the k-term expansion forms the best k-dimensional subspace for the data (in terms of minimizing the mean square error). Note that this isn't the only basis for which this is true, as any orthonormal basis will produce the same mean square error.

For any orthonormal basis $\{\boldsymbol{\psi}_i\}_{i=1}^n$,

$$\frac{1}{p}\sum_{i=1}^{p}||\mathbf{x}^{(i)}|| = \sum_{j=1}^{k}\boldsymbol{\psi}_j^T C\boldsymbol{\psi}_j + \sum_{j=k+1}^{n}\boldsymbol{\psi}_j^T C\boldsymbol{\psi}_j$$

where this sum is a constant. Thus, minimizing the second term of the sum is the same as maximizing the first term of the sum. Using the eigenvector basis,

$$\sum_{j=1}^{k}\boldsymbol{\phi}_j^T C\boldsymbol{\phi}_j = \lambda_1 + \lambda_2 + \cdots + \lambda_k.$$

Evaluating the second term,

$$\frac{1}{p}\sum_{i=1}^{p}||\mathbf{x}_{err}^{(i)}||^2 = \sum_{j=k+1}^{n}\boldsymbol{\phi}_j^T C\boldsymbol{\phi}_j = \lambda_{k+1} + \cdots + \lambda_n.$$

Remember that the first k eigenvectors of C form the best basis over all k-term expansions.

To prove this, let $\{\boldsymbol{\psi}_i\}_{i=1}^k$ be any other k-dimensional basis. Each $\boldsymbol{\psi}_i$ can be written in terms of its coordinates with respect to the eigenvectors of C,

$$\boldsymbol{\psi}_i = \sum_{l=1}^{n}\left(\boldsymbol{\psi}_i^T\boldsymbol{\phi}_l\right)\boldsymbol{\phi}_l = \Phi\alpha_i$$

so that $\alpha_{il} = \boldsymbol{\psi}_i^T\boldsymbol{\phi}_l$. Using this,

$$\boldsymbol{\psi}_i^T C\boldsymbol{\psi}_i = \alpha_i^T D\alpha_i,$$

where D is the diagonal matrix of the eigenvalues of C.

Next, consider the mean squared projection of the data onto this k-dimensional basis:

$$\sum_{j=1}^{k}\boldsymbol{\psi}_j^T C\boldsymbol{\psi}_j = \sum_{j=1}^{k}\alpha_j^T D\alpha_j = \sum_{j=1}^{k}\lambda_1\alpha_{j1}^2 + \cdots + \lambda_n\alpha_{jn}^2,$$

so that:

$$\sum_{j=1}^{k}\boldsymbol{\psi}_j^T C\boldsymbol{\psi}_j = \lambda_1\sum_{j=1}^{k}\alpha_{j1}^2 + \lambda_2\sum_{j=1}^{k}\alpha_{j2}^2 + \cdots + \lambda_n\sum_{j=1}^{k}\alpha_{jn}^2.$$

Now we want to show that each of these coefficients is less than one. Consider the coefficients

$$\{\alpha_{1i},\ldots,\alpha_{ki}\} = \{\boldsymbol{\psi}_1^T\boldsymbol{\phi}_i,\ldots,\boldsymbol{\psi}_k^T\boldsymbol{\phi}_i\}.$$

Because the coefficients from the projection of $\boldsymbol{\phi}_i$ onto the subspace spanned by the $\boldsymbol{\psi}$'s,

$$\text{Proj}_\Psi(\boldsymbol{\phi}_i) = \alpha_{1i}\boldsymbol{\psi}_1 + \cdots + \alpha_{ki}\boldsymbol{\psi}_k$$

where we have what we need:

$$\sum_{j=1}^{k} \alpha_{ji}^2 = ||\text{Proj}_\Psi(\boldsymbol{\phi}_i)||^2 \leq 1$$

with equality if and only if $\boldsymbol{\phi}_i$ is in the span of the columns of Ψ. Now, we can say that the maximum of

$$\sum_{j=1}^{k} \boldsymbol{\psi}_j^T C \boldsymbol{\psi}_j = \lambda_1 \sum_{j=1}^{k} \alpha_{ji}^2 + \lambda_2 \sum_{j=1}^{k} \alpha_{j2}^2 + \cdots + \lambda_n \sum_{j=1}^{k} \alpha_{jn}^2$$

is found by replacing the coefficients of the first k terms to 1, and the remaining coefficients to zero. This corresponds to the value of the error found by using the eigenvector basis.

Thus, we have shown that the first k eigenvectors of the covariance matrix forms the best k dimensional subspace for the data. But, as stated above, any basis for that k dimensional subspace would work.

2.5 Application: Linear Classifier and PCA

The Data

The data we used was breast cancer data and we set out to classify whether a tumor is benign or malignant. There were 699 original observations, each with 9 integer-valued measurements. These integer valued measurements are graded with a value from 1 to 10, with 1 being typically benign. The 9 integer-valued measurements are cell shape, uniformity of cell size, clump thickness, bare nuclei, cell size, normal nucleoli, clump cohesiveness, nuclear chromatin, and mitosis. There were 16 observations that had missing numbers for "Bare Nuclei," so we removed these observations, leaving us with a total of 683 observations. A return of a classification of 1 is malignant while a return of 0 is benign. We have 239 malignant observations and 444 benign observations. We used $\frac{2}{3}$ of the data for building the model and the remaining $\frac{1}{3}$ for testing. We used both a linear classifier method and also the PCA, both results are shown below.

Linear Classifier

For the linear classifier, start with this equation below where the x_i is the integer-valued measurement and the a's are unknown constants that we aim to find. We will use both benign and malignant observations to "train" the data and to find the a values and then can use the untrained data to be classified as either benign or malignant. The X is the matrix that contains the observations.

$$a_1x_1^{(i)} + a_2x_2^{(i)} + \cdots + a_9x_9^{(i)} + b = y^{(i)} \begin{cases} = 0 & \text{if benign} \\ = 1 & \text{if malignant} \end{cases}$$

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_9^{(1)} & 1 \\ x_1^{(2)} & x_2^{(2)} & \dots & x_9^{(2)} & 1 \\ \vdots & \vdots & & & \vdots \\ x_1^{(683)} & x_2^{(683)} & \dots & x_9^{(683)} & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_9 \\ b \end{bmatrix} = \mathbf{y}$$

$$X\mathbf{a} = \mathbf{y}$$

But X is an overdetermined matrix, meaning there are more equations than there are unknowns, so it not invertible. We instead break X down using the reduced SVD, explained in 1.3, (so that Σ is an invertible matrix), so $X = U\Sigma V^T$.

$$\begin{aligned} U\Sigma V^T\mathbf{a} &= \mathbf{y}, \quad \text{then, multiplying by } U^T \\ U^TU\Sigma V^T\mathbf{a} &= \mathbf{y}, \quad \text{where } U^TU = I \text{ since } U \text{ is orthogonal, so} \\ \Sigma V^T\mathbf{a} &= U^T\mathbf{y}, \quad \text{taking the inverse of } \Sigma, \\ V^T\mathbf{a} &= \Sigma^{-1}U^T\mathbf{y} \\ VV^T\mathbf{a} &= V\Sigma^{-1}U^T\mathbf{y} \end{aligned}$$

where Σ^{-1} has entries $\dfrac{1}{\sigma_i}$ along the diagonal. Since V is not orthogonal, $VV^T \neq I$, we have found a projection of $\mathbf{a}$ in the columnspace of of X.

Using the breast cancer data, we get $\mathbf{a} = \begin{bmatrix} 0.3043 \\ 0.1128 \\ 0.1071 \\ 0.1529 \\ 0.0405 \\ 0.5683 \\ 0.1442 \\ 0.2214 \\ 0.0103 \\ 0.4798 \end{bmatrix}$.

To test our test data, we put our test observations into the matrix X and, using the $\mathbf{a}$ vector above, look at $\mathbf{y}$ for the classification 1 or 0 to determine if a tumor is malignant or

		Prediction	
		0	1
Actual	0	TN	FP
	1	FN	TP

Figure 1: Confusion Matrix

benign. The classifier now looks like:

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_9^{(1)} & 1 \\ x_1^{(2)} & x_2^{(2)} & \dots & x_9^{(2)} & 1 \\ \vdots & \vdots & & & \vdots \\ x_1^{(683)} & x_2^{(683)} & \dots & x_9^{(683)} & 1 \end{bmatrix} \begin{bmatrix} 0.3043 \\ 0.1128 \\ 0.1071 \\ 0.1529 \\ 0.0405 \\ 0.5683 \\ 0.1442 \\ 0.2214 \\ 0.0103 \\ 0.4798 \end{bmatrix} = \mathbf{y}.$$

PCA

To measure novelty for the PCA, we look at the norm of the reconstruction error of the projection. To do this, we project the data onto the subspace formed by the k eigenvectors (which is the subspace formed by the best basis) and we then compute each observation's reconstruction error, defined in Section 2 by projecting it back into the original subspace. A higher reconstruction error value is what is considered novel while a smaller value is normal. For the PCA, we performed the SVD on the "benign" data and after reducing the dimensionality, we used 7 principal components, or basis vectors out of the original 9. However, the results using this method were poor, so we tried something else and instead looked at the nullspace of the benign data (the last two columns in the SVD) for the basis and our performance using this method was much better, as shown below.

How to Measure Performance

To check the performance of our models, we use a confusion matrix (Figure 1). The confusion matrix shows for each pair of classes, how many benign tumors were incorrectly classified as malignant (false positive, FP), how many malignant tumors were incorrectly classified as benign (false negative, FN), and how many benign and malignant tumors were correctly classified (true negative, TN, and true positive, TP, respectively). In our case, a negative refers to the tumor being classified as benign.

Clearly, we want the TN and TP cells to be as close to 100% as possible.

The confusion matrix for our the linear classifier is

$$C = \begin{bmatrix} 0.9745 & 0.0255 \\ 0.0651 & 0.9349 \end{bmatrix}.$$

The confusion matrix for the PCA method using 7 principal components is

$$C = \begin{bmatrix} 0.890 & 0.110 \\ 0.140 & 0.760 \end{bmatrix},$$

while the confusion matrix using the 2 components from the nullspace is

$$C = \begin{bmatrix} 0.912 & 0.088 \\ 0.004 & 0.996 \end{bmatrix}.$$

Thus, our linear classifier performed better than the PCA method. We have that 97.5% of benign tumors were correctly classified as benign and 93.5% of malignant tumors were classified as malignant for the linear classifier compared to 91.2% correct benign classification and 99.6% correct malignant classification for the more successful PCA model. If we were to try to improve this classifier, we would want to try to increase the number of true positives more so than true negatives because it is better to falsely classify a benign tumor as malignant, as the tumor is still harmless, than to classify a malignant tumor as benign.

3 The Kernel PCA

While the PCA deals with linear data, the Kernel PCA is interested in principal components, or features, that are nonlinearly related to the input variables. To do this, we compute the dot products in the feature space by means of a kernel function in the input space. Given any algorithm that can be expressed by dot products, that is, without explicit use of the variables, the kernel method allows us to construct a nonlinear versions of it.

The PCA tries to find a low-dimensional linear subspace that the data are confined to. However, sometimes the data are confined to a low-dimensional nonlinear subspace, which is where the Kernel PCA comes in. Looking at Figure 2, the data are mainly located along (or at least close to) a curve in 2-D but the PCA cannot reduce the dimensionality from two to one because the points aren't located along a straight line. The Kernel PCA can recognize that this data is one-dimensional but in a higher dimensional space, called the feature space. The Kernel PCA does not explicitly compute the higher dimensional space, it instead projects the data into this feature space so that we can classify the data.

3.1 Important Definitions

Before going into the kernel PCA, here are some important terms and their definitions:

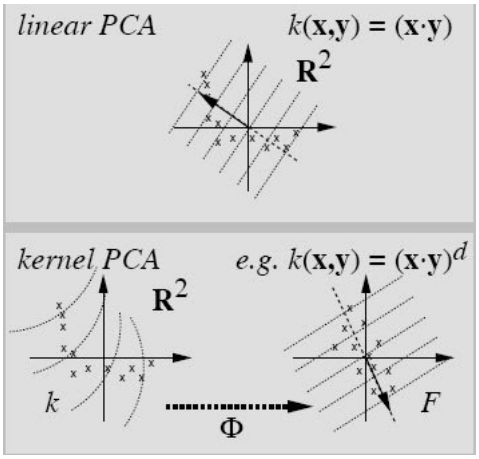

Figure 2: Kernel PCA

Inner Product
An inner product on a vector space V is a function that, to each pair of vectors $\mathbf{u}$ and $\mathbf{v}$ in V, associates a real number $\langle \mathbf{u}, \mathbf{v} \rangle$ and satisfies the following axioms, for all $\mathbf{u}, \mathbf{v}, \mathbf{w}$ in V and all scalars c:

1. $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{v}, \mathbf{u} \rangle$
2. $\langle \mathbf{u} + \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{u}, \mathbf{w} \rangle + \langle \mathbf{v}, \mathbf{w} \rangle$
3. $\langle c\mathbf{u}, \mathbf{v} \rangle = c\langle \mathbf{u}, \mathbf{v} \rangle$
4. $\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$ and $\langle \mathbf{u}, \mathbf{u} \rangle = 0$ if and only if $\mathbf{u} = \mathbf{0}$.

Feature Space
The feature space is a space spanned by Φ. The kernel trick is used to select, from the data, a relevant subset forming a basis in a feature space $\mathcal{F}$. Thus, the selected vectors define a subspace in $\mathcal{F}$. In other words, they map data from original input space into a (usually high-dimensional) feature space where linear relations exist among data.

Positive Semi-Definite Matrix
A positive semi-definite matrix is a symmetric $n \times n$ matrix A such that its eigenvalues are all non-negative. This holds if and only if

$$\mathbf{v}^T A \mathbf{v} \geq 0$$

for all vectors $\mathbf{v}$.

Kernel Function
A kernel is a function κ that for all $\mathbf{x}, \mathbf{z} \in X$ satisfies

$$\kappa(\mathbf{x}, \mathbf{z}) = \big\langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{z}) \big\rangle,$$

where $\boldsymbol{\phi}$ is a mapping from X to an (inner product) feature space $\mathcal{F}$

$$\boldsymbol{\phi} :\to \boldsymbol{\phi}(\mathbf{x}) \in \mathcal{F}.$$

3.2 Formulating the Standard PCA With Dot Products

Before going into the Kernel PCA, we will formulate the standard PCA with dot products which we will then use when deriving the Kernel PCA.

Given a set of mean subtracted observations (the process of mean subtracting observations can be found in the Appendix), $\mathbf{x}_k \in \mathbb{R}^n$, $k = 1, \ldots, p$, $\sum_{k=1}^{p} \mathbf{x}_k = 0$, the PCA diagonalizes the covariance matrix,

$$C = \frac{1}{p} \sum_{j=1}^{p} \mathbf{x}_j \mathbf{x}_j^T. \tag{3}$$

To do this, we solve

$$\lambda \mathbf{v} = C\mathbf{v}, \tag{4}$$

where $\lambda \geq 0$ and the eigenvectors, $\mathbf{v}$, are in $\mathbb{R}^n \backslash \{0\}$.

All solutions, $\mathbf{v}$, where $\lambda \neq 0$, must lie in the span of $\mathbf{x}_1 \ldots \mathbf{x}_p$ because

$$\lambda \mathbf{v} = C\mathbf{v} = \frac{1}{p} \sum_{j=1}^{p} (\mathbf{x}_j \cdot \mathbf{v}) \mathbf{x}_j. \tag{5}$$

In this case, (5) is equivalent to

$$\lambda (\mathbf{x}_k \cdot \mathbf{v}) = (\mathbf{x}_k \cdot C\mathbf{v}) \quad \text{for all } k = 1, \ldots, p. \tag{6}$$

Now, we will be doing these computations in the feature space $\mathcal{F}$, which is related to the input space by the nonlinear map:

$$\Phi : \mathbb{R}^n \to \mathcal{F}, \quad \mathbf{x} \to \mathbf{X}. \tag{7}$$

Note that for the rest of this section, upper case letters are used for elements of $\mathcal{F}$ and lower case letters are used for elements of $\mathbb{R}^n$.

Also, assume we are working with mean subtracted data, which means that

$$\sum_{k=1}^{p} \Phi(\mathbf{x}_k) = 0.$$

In the feature space, $\mathcal{F}$, the covariance matrix looks like:

$$\bar{C} = \frac{1}{p} \sum_{j=1}^{p} \Phi(\mathbf{x}_j) \Phi(\mathbf{x}_j)^T. \tag{8}$$

Now, we want to find the eigenvalues and eigenvectors that satisfy

$$\lambda \mathbf{V} = \bar{C}\mathbf{V}, \tag{9}$$

where $\lambda \neq 0$ and $\mathbf{V} \in \mathcal{F}\backslash\{0\}$. As in (4), all solutions, $\mathbf{V}$, where $\lambda \neq 0$, lie in the span of $\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_p)$. Because of this, we can use:

$$\lambda(\Phi(\mathbf{x}_k) \cdot \mathbf{V}) = (\Phi(\mathbf{x}_k) \cdot \bar{C}\mathbf{V}) \tag{10}$$

for all $k = 1, \ldots, p$. Also, because all solutions, $\mathbf{V}$, where $\lambda \neq 0$, lie in the span of $\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_p)$, there exist coefficients α_i $(i = 1, \ldots, p)$ such that

$$\mathbf{V} = \sum_{i=1}^{p} \alpha_i \Phi(\mathbf{x}_i). \tag{11}$$

Combining (10) and (11),

$$\lambda \sum_{i=1}^{p} \alpha_i (\Phi(\mathbf{x}_k) \cdot \Phi(\mathbf{x}_i)) = \frac{1}{p} \sum_{i=1}^{p} \alpha_i \left(\Phi(\mathbf{x}_k) \cdot \sum_{j=1}^{p} \Phi(\mathbf{x}_j)(\Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_i)) \right) \tag{12}$$

for all $k = 1, \ldots, p$.

Defining a $p \times p$ kernel matrix K by

$$K_{ij} := (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)) \tag{13}$$

and subbing this into (12),

$$p\lambda K\boldsymbol{\alpha} = K^2\boldsymbol{\alpha}. \tag{14}$$

In this equation, $\boldsymbol{\alpha}$ is the column vector with entries $\alpha_1, \ldots, \alpha_p$.

Now, to find solutions of $p\lambda K\boldsymbol{\alpha} = K^2\boldsymbol{\alpha}$, we need to solve

$$p\lambda\boldsymbol{\alpha} = K\boldsymbol{\alpha} \tag{15}$$

for the nonzero eigenvalues.

Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$ be the eigenvalues of K and let $\boldsymbol{\alpha}^1, \ldots \boldsymbol{\alpha}^p$ be the corresponding eigenvectors, where λ_{m+1} is the first zero eigenvalue. Next, we normalize $\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^m$ by requiring that the corresponding vectors in $\mathcal{F}$ are normalized. That is,

$$(\mathbf{V}^k \cdot \mathbf{V}^k) = 1 \quad \text{for all} \quad k = 1, \ldots, m. \tag{16}$$

Because of (11) and (15), this is a normalization condition for $\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^p$:

$$1 = \sum_{i,j=1}^{p} \alpha_i^k \alpha_j^k (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)) = \sum_{i,j=1}^{p} \alpha_i^k \alpha_j^k K_{ij} \tag{17}$$

$$= (\boldsymbol{\alpha}^k \cdot K\boldsymbol{\alpha}_j^k) = \lambda_k(\boldsymbol{\alpha}^k \cdot \boldsymbol{\alpha}^k). \tag{18}$$

Finally, we want to compute the principal component projections onto the eigenvectors $\mathbf{V}^k$ in $F(k = 1, \ldots, m)$. Let $\mathbf{x}$ be a test point with an image $\Phi(\mathbf{x})$ in $\mathcal{F}$, then

$$(\mathbf{V}^k \cdot \Phi(\mathbf{x})) = \sum_{i=1}^{p} \alpha_i^k (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})) \tag{19}$$

is called its nonlinear principal components corresponding to Φ.

In conclusion, to compute the principal components in $\mathcal{F}$, we first compute the matrix K (equation (13)) and then compute its eigenvectors and normalize them in $\mathcal{F}$ (equations (14) to (18)). Then, we compute the projections of a test point onto the eigenvectors (equation (19)).

3.3 The Kernel PCA Algorithm

Summarizing the previous section and using the kernel function κ, in order to perform the kernel PCA, the following steps must be followed:

1. Compute the kernel matrix $K_{ij} = (k(\mathbf{x}_i, \mathbf{x}_j))_{ij}$, where κ is the kernel function, as defined above.

2. Solve $p\lambda\boldsymbol{\alpha} = K\boldsymbol{\alpha}$ by diagonalizing K and normalize the eigenvector expansion coefficients α^n by requiring $\lambda_n(\boldsymbol{\alpha}^n \cdot \boldsymbol{\alpha}^n) = 1$.

3. To extract the principal components that correspond to the kernel k of a test point $\mathbf{x}$, compute the projections onto the eigenvectors:

$$(\mathbf{V}^n \cdot \Phi(\mathbf{x})) = \sum_{i=1}^{p} \alpha_i^n k(\mathbf{x}_i, \mathbf{x}),$$

   where $\mathbf{V}^n$ is the set of eigenvectors in $\mathcal{F}$ and $\Phi(\mathbf{x})$ is the image of $\mathbf{x}$ in $\mathcal{F}$.

   This algorithm shows that we exclusively compute the dot products between mappings, and never need to compute the mappings explicitly.

3.4 Novelty Detection Using the Kernel PCA

This section outlines applying the kernel PCA method to novelty detection.

Notation

We assume that the original data is given as n points, $\mathbf{x}_i \in \mathbb{R}^d$. Then there is some mapping that goes to the feature space, $\mathcal{F}$, so that

$$\mathbf{x}_i \longrightarrow \boldsymbol{\phi}(\mathbf{x}_i) \in \mathcal{F}.$$

If the context is clear, we will use $\boldsymbol{\phi}(\mathbf{x}_i) = \boldsymbol{\phi}_i$. Also, if $\mathbf{x}$ is a general point, then $\boldsymbol{\phi}(\mathbf{x}) = \boldsymbol{\phi}_x$ will be used. Similarly, we will use the following notation for the mean, and mean subtracted data points:

$$\boldsymbol{\phi}_0 = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\phi}_i \qquad \text{and} \qquad \tilde{\boldsymbol{\phi}}_i = \boldsymbol{\phi}_i - \boldsymbol{\phi}_0.$$

Using the appropriate inner product space, we define the kernel function κ:

$$\kappa(\mathbf{x}, \mathbf{y}) = \langle \boldsymbol{\phi}_x, \boldsymbol{\phi}_y \rangle$$

with the norm in the feature space defined with the inner product as:

$$\|\boldsymbol{\phi}_x\|^2 = \langle \boldsymbol{\phi}_x, \boldsymbol{\phi}_x \rangle = \kappa(\mathbf{x}, \mathbf{x}).$$

As noted previously, the inner product in the feature space can be computed directly in $\mathbb{R}^d$ using κ.

We define the kernel matrix using our n points in $\mathbb{R}^d$ and the kernel function κ:

$$K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$$

so we can compute the mean subtracted kernel matrix using the previous computation for the mean subtracted inner product. That is,

$$\begin{aligned}\tilde{K}_{ij} &= \langle \tilde{\boldsymbol{\phi}}_i, \tilde{\boldsymbol{\phi}}_j \rangle \\ &= \kappa(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{n}\sum_{r=1}^{n} \kappa(\mathbf{x}_i, \mathbf{x}_r) - \frac{1}{n}\sum_{s=1}^{n} \kappa(\mathbf{x}_j, \mathbf{x}_s) + \frac{1}{n^2}\sum_{r=1}^{n}\sum_{s=1}^{n} \kappa(\mathbf{x}_r, \mathbf{x}_s).\end{aligned}$$

These values can thus be computed by using the original kernel matrix K.

The Eigenvalues and Eigenvectors of the Covariance Matrix

To perform principal components analysis in the feature space $\mathcal{F}$, define the covariance matrix, as usual, in the feature space as:

$$C = \frac{1}{n}\sum \tilde{\boldsymbol{\phi}}_i \tilde{\boldsymbol{\phi}}_i^T = \frac{1}{n}\Phi\Phi^T$$

where Φ is a matrix whose columns are given by the mean subtracted data in feature space:

$$\Phi = \left[\tilde{\boldsymbol{\phi}}_1, \tilde{\boldsymbol{\phi}}_2, \cdots, \tilde{\boldsymbol{\phi}}_n\right].$$

Using this notation, we can write:

$$\tilde{K} = \Phi^T\Phi.$$

The data in the feature space is usually of very high dimension. The feature space is infinite dimensional when using the Gaussian kernel (defined below), so we do not want to construct the actual eigenvectors $\mathbf{v}$, instead we will only compute the inner product involving $\mathbf{v}$.

Now, suppose that $\mathbf{v}$ is the an eigenvector of the covariance matrix C and λ its associated eigenvalue (λ and $\mathbf{v}$ are called eigenpairs), so that

$$\lambda \mathbf{v} = C\mathbf{v}.$$

If we expand $C\mathbf{v} = \frac{1}{n}\Phi(\Phi^T\mathbf{v})$, we see that $\mathbf{v}$ is a linear combination of the columns of Φ, so $\mathbf{v}$ lies in the span of the vectors $S = \left\{\tilde{\boldsymbol{\phi}}_1, \tilde{\boldsymbol{\phi}}_2, \cdots, \tilde{\boldsymbol{\phi}}_n\right\}$. Thus, we can consider the equivalent system of equations:

$$\lambda \Phi^T \mathbf{v} = \Phi^T C \mathbf{v}. \tag{20}$$

We define $\boldsymbol{\alpha}$ as the coordinates of $\mathbf{v}$ with respect to the feature vectors. That is,

$$\mathbf{v} = \sum_{i=1}^{n} \alpha_i \tilde{\boldsymbol{\phi}}_i = \Phi\boldsymbol{\alpha}. \tag{21}$$

By the definition of the covariance C, and by substituting (21) into (20), we have:

$$\lambda \Phi^T \Phi\boldsymbol{\alpha} = \frac{1}{n}\Phi^T\Phi\Phi^T\Phi\boldsymbol{\alpha}.$$

From this we get our main equation:

$$n\lambda\tilde{K}\boldsymbol{\alpha} = \tilde{K}^2\boldsymbol{\alpha}. \tag{22}$$

If we compare the solutions to the above with the solutions to the following eigenvalue, eigenvector problem:

$$n\lambda\boldsymbol{\alpha} = \tilde{K}\boldsymbol{\alpha}, \tag{23}$$

we see that any solution to (23) will clearly also be a solution to (22). Conversely, we may have solutions to (22) that are not solutions to (23), but these vectors would necessarily be in the null space of $\tilde{K}$, which is not of interest to us. Thus, we have our main result thus far- we have made $\boldsymbol{\alpha}$ and λ computable from the kernel matrix $\tilde{K}$.

Scaling the Eigenvectors

We need to make sure that we have unit eigenvectors $\mathbf{v}$, so that the projection of a new vector in $\mathcal{F}$ will give the formula in (23). If we use (21) and the definition of $\tilde{K}$, then:

$$\|\mathbf{v}\|^2 = \langle\Phi\boldsymbol{\alpha}, \Phi\boldsymbol{\alpha}\rangle = \boldsymbol{\alpha}^T\Phi^T\Phi\boldsymbol{\alpha} = \boldsymbol{\alpha}^T\tilde{K}\boldsymbol{\alpha} = n\lambda\boldsymbol{\alpha}^T\boldsymbol{\alpha} = n\lambda\|\boldsymbol{\alpha}\|^2.$$

Therefore, we will scale $\boldsymbol{\alpha}$ so that

$$\|\boldsymbol{\alpha}\| = \frac{1}{\sqrt{n\lambda}}.$$

Residual Variance
We can use the eigenvalues of the covariance matrix to determine what percent of the total variation is explained by taking k out of n possible eigenvectors, similar to what is explained in 2.3. This is given by the sum of the first k normalized eigenvalues:

$$\sum_{i=1}^{k} \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j}.$$

One way to compute this is using the trace of a matrix (the sum of its diagonal elements). For an $n \times n$ matrix A with real eigenvalues,

$$\sum_{j=1}^{n} \lambda_j = \mathrm{tr}(A) = \sum_{j=1}^{n} A_{jj}.$$

In particular, if the eigenvectors are coming from the symmetric matrix $\tilde{K}$ and we use p eigenpairs for our approximate subspace, then the residual variance is the percent of variation explained by using p dimensions in feature space.

Measure of the Novelty of a New Point
Given a new vector $\mathbf{z} \in \mathbb{R}^d$, we set the measure of novelty as a slightly modified reconstruction error in feature space:

$$F(\mathbf{z}) = \|\tilde{\boldsymbol{\phi}}_z\|^2 - \|\mathrm{Proj}_Q(\tilde{\boldsymbol{\phi}}_z)\|^2$$

where Q is a $q-$dimensional subspace of $\mathcal{F}$ whose basis is given by the eigenvectors $\mathbf{v}_1, \cdots, \mathbf{v}_q$.

We will show how to compute the value of F without explicitly computing $\tilde{\boldsymbol{\phi}}_z$, or any of the eigenvectors. First, we look at $\|\tilde{\boldsymbol{\phi}}_z\|^2$:

$$\begin{aligned}
\|\tilde{\boldsymbol{\phi}}_z\|^2 &= \langle \boldsymbol{\phi}_z - \boldsymbol{\phi}_0, \boldsymbol{\phi}_z - \boldsymbol{\phi}_0 \rangle \\
&= \langle \boldsymbol{\phi}_z, \boldsymbol{\phi}_z \rangle - 2\langle \boldsymbol{\phi}_z, \boldsymbol{\phi}_0 \rangle + \langle \boldsymbol{\phi}_0, \boldsymbol{\phi}_0 \rangle \\
&= \langle \boldsymbol{\phi}_z, \boldsymbol{\phi}_z \rangle - \frac{2}{n} \sum_{i=1}^{n} \langle \boldsymbol{\phi}_z, \boldsymbol{\phi}_i \rangle + \frac{1}{n^2} \sum_{r=1}^{n} \sum_{s=1}^{n} \langle \boldsymbol{\phi}_r, \boldsymbol{\phi}_s \rangle \\
&= \kappa(\mathbf{z}, \mathbf{z}) - \frac{2}{n} \sum_{i=1}^{n} \kappa(\mathbf{z}, \mathbf{x}_i) + \frac{1}{n^2} \sum_{r=1}^{n} \sum_{s=1}^{n} \kappa(\mathbf{x}_r, \mathbf{x}_s).
\end{aligned}$$

Each expression is now computable simply by using the kernel function in $\mathbb{R}^d$. Going to the next projection, if we expand the projection as:

$$\mathrm{Proj}_Q(\tilde{\boldsymbol{\phi}}_z) = \beta_1 \mathbf{v}_1 + \beta_2 \mathbf{v}_2 + \cdots + \beta_q \mathbf{v}_q,$$

then since the eigenvectors are orthonormal in the feature space,

$$\|\mathrm{Proj}_Q(\tilde{\boldsymbol{\phi}}_z)\|^2 = \beta_1^2 + \cdots + \beta_q^2 = \langle \tilde{\boldsymbol{\phi}}_z, \mathbf{v}_1 \rangle^2 + \langle \tilde{\boldsymbol{\phi}}_z, \mathbf{v}_2 \rangle^2 + \cdots + \langle \tilde{\boldsymbol{\phi}}_z, \mathbf{v}_q \rangle^2.$$

Typically, extracting a "feature" from new data point $\tilde{\boldsymbol{\phi}}_z$, using eigenvector $\mathbf{v}$, means computing the scalar projection:

$$\langle \tilde{\boldsymbol{\phi}}_z, \mathbf{v} \rangle.$$

Using the definition of $\mathbf{v}$ in (21), this "feature" (or scalar projection) can be computed without computing $\mathbf{v}$ directly. From (23), the coordinates of $\mathbf{v}$ in the vector $\boldsymbol{\alpha}$ have been computed from the mean subtracted kernel matrix $\tilde{K}$. Thus,

$$\langle \tilde{\boldsymbol{\phi}}_z, \mathbf{v} \rangle = \langle \tilde{\boldsymbol{\phi}}_z, \sum_{i=1}^{n} \alpha_i \tilde{\boldsymbol{\phi}}_i \rangle = \sum_{i=1}^{n} \alpha_i \langle \tilde{\boldsymbol{\phi}}_z, \tilde{\boldsymbol{\phi}}_i \rangle. \tag{24}$$

This last inner product has already been computed as:

$$\langle \tilde{\boldsymbol{\phi}}_z, \tilde{\boldsymbol{\phi}}_i \rangle = \kappa(\mathbf{z}, \mathbf{x}_i) - \frac{1}{n} \sum_{r=1}^{n} \kappa(\mathbf{x}_i, \mathbf{x}_r) - \frac{1}{n} \sum_{s=1}^{n} \kappa(\mathbf{z}, \mathbf{x}_s) + \frac{1}{n^2} \sum_{r=1}^{n} \sum_{s=1}^{n} \kappa(\mathbf{x}_r, \mathbf{x}_s). \tag{25}$$

Novelty Detection
For each new point z, we can compute the "novelty," $F(z)$, by looking at the maximum value of F over the z values. From this, we see which z values are "normal" to get a threshold value ν. A value z is said to be novel if

$$F(z) \geq \nu.$$

4 Application: Spiral Results and the Kernel PCA

For our application, we used the Gaussian kernel which is defined as $\kappa(\mathbf{x}, \mathbf{y}) = e^{\frac{-||\mathbf{x}-\mathbf{y}||}{2\sigma^2}}$. The σ is the width of the Gaussian kernel and is chosen to maximize the number of noisy points outside the decision boundary and the number of regular points inside the boundary. The spiral results show the effect of changing the value of σ and the number of feature vectors, p. The kernel PCA section applies the kernel PCA to the breast cancer data that was used in 2.5.

4.1 Spiral Results

In MATLAB, the code is kpcabound(data,sigma,numev,outlier) where "data" is the array of data points, where one row is assigned for each data point; σ is the width of Gaussian kernel; numev is the number of eigenvalues to be extracted, or features, denoted q; and outlier is the number of points outside the decision boundary. The kpcabound demonstrates 'Kernel PCA for novelty detection' and plots the reconstruction error in feature space into the original space and plots the decision boundary enclosing all data points. We used a contour plot of the reconstruction error for a selection of points in the plane. The red curve denotes the contour at the maximum error of the "normal" data. Below are several figures with varying

σ and q values. As you can see, when $\sigma = 0.25$ and $q = 50$, the contour line more closely follows the data.

If we use Gaussian kernels, then the data in feature space all lie on a hyperdimensional sphere. That is,

$$\|\boldsymbol{\phi}_x\|^2 = \langle \boldsymbol{\phi}_x, \boldsymbol{\phi}_x \rangle = \kappa(\mathbf{x}, \mathbf{x}) = \mathrm{e}^0 = 1.$$

Therefore, a subspace cutting through the sphere in feature space will show itself by including neighborhoods about each point in $\mathbb{R}^d$. We can then choose a small σ if we want to enclose the data tightly, or we can choose larger σ to allow the boundary some room.

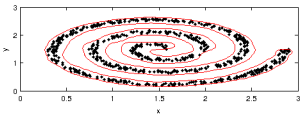

Figure 3: $\sigma = 0.25$, $q = 40$

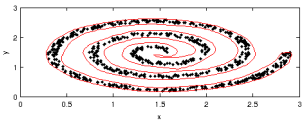

Figure 4: $\sigma = 0.25$, $q = 50$

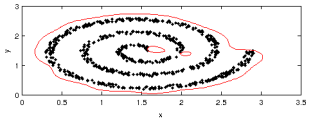

Figure 5: $\sigma = 0.25$, $q = 20$

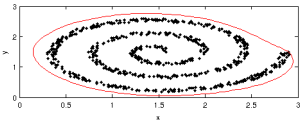

Figure 6: $\sigma = 0.50$, $q = 40$

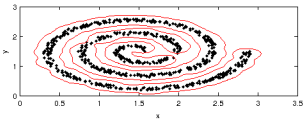

Figure 7: $\sigma = 0.10$, $q = 40$

4.2 The Kernel PCA

As in 2.5, we used the breast cancer data but instead used the kernel PCA. Again, $\frac{2}{3}$ of the data was used for building the model and $\frac{1}{3}$ was used for testing. To get the best results, we changed the σ value until the highest percentage of correct classification was achieved. An interesting result of this data was that when $0.5 \leq \sigma \leq 100$, the percentage of correct classification stayed relatively the same, around 96% and around 19 feature vectors. The graphs below show where the lines for benign and malignant intersect, which gives us the number of features and the percent classification for that particular σ value.

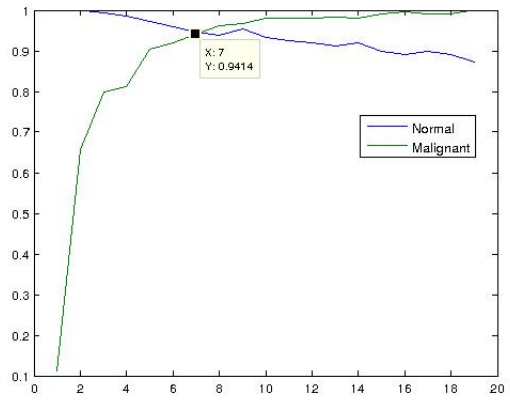

Figure 8: $\sigma = 2, \quad q = 14$

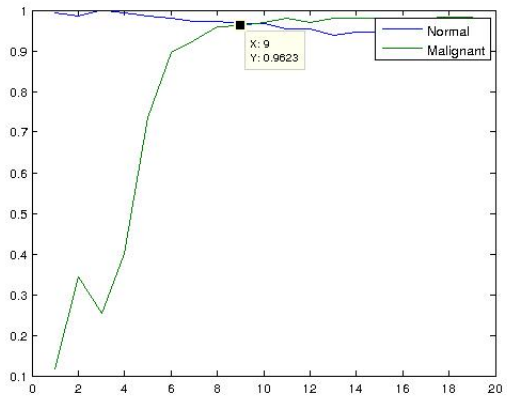

Figure 9: $\sigma = 12, \quad q = 18$

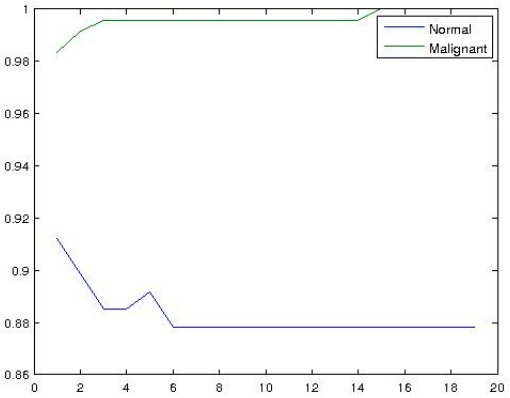

Figure 10: $\sigma = .20, \quad q =$N/A

The confusion matrix for Figure 8 is $C = \begin{bmatrix} 0.872 & 0.128 \\ 0 & 1.00 \end{bmatrix}$.

The confusion matrix for Figure 9 is $C = \begin{bmatrix} 0.932 & 0.068 \\ 0.017 & 0.983 \end{bmatrix}$.

The confusion matrix for Figure 10 is $C = \begin{bmatrix} 0.878 & 0.122 \\ 0 & 1.00 \end{bmatrix}$.

Thus, when $\sigma = 12$, we get the highest correct classification, at 93% correct benign classification and 98% correct malignant classification. These results are similar to the linear classifier in 2.5, except with a benign correct classification of 98% and malignant classification of 93% for the linear classifier.

5 Appendix

5.1 Mean Subtracting in High-Dimensional Space

Using mean subtracted data is necessary when performing the kernel PCA, so this section shows how to mean subtract data when working in a high-dimensional space.

Given any Φ and any set of observations $\mathbf{x}_1, \ldots, \mathbf{x}_p$, the points

$$\tilde{\Phi}(\mathbf{x}_i) := \Phi(\mathbf{x}_i) - \frac{1}{p}\sum_{i=1}^{p}\Phi(\mathbf{x}_i) \tag{26}$$

are mean subtracted. Thus, the assumption that the data is mean subtracted from 3.2 holds.

Now, define the covariance matrix, $K_{ij} = (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j))$, and $\tilde{K}_{ij} = (\tilde{\Phi}(\mathbf{x}_i) \cdot \tilde{\Phi}(\mathbf{x}_j))$ in $\mathcal{F}$. We are back to the familiar eigenvalue problem

$$\tilde{\lambda}\tilde{\boldsymbol{\alpha}} = \tilde{K}\tilde{\boldsymbol{\alpha}}$$

where $\tilde{\boldsymbol{\alpha}}$ is the expansion coefficients of an eigenvector (in $\mathcal{F}$) in terms of the points in (26), $\tilde{\boldsymbol{V}} = \sum_{i=1}^{p}\tilde{\alpha}_i\tilde{\Phi}(\mathbf{x}_i)$. Because we do not have the mean subtracted data (26), we cannot compute $\tilde{K}$ directly, but we can express it in terms of the non-mean subtracted K. The notation in the following will be: $K_{ij} = (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j))$, and $1_{ij} = 1$ for all $i, j, (1_p)_{ij} := \frac{1}{p}$. To compute $\tilde{K}_{ij} = (\tilde{\Phi}(\mathbf{x}_i) \cdot \tilde{\Phi}(\mathbf{x}_j))$,

$$\tilde{K}_{ij} = \left((\Phi(\mathbf{x}_i) - \frac{1}{p}\sum_{l=1}^{p}\Phi(\mathbf{x}_l)) \cdot (\Phi(\mathbf{x}_j) - \frac{1}{p}\sum_{n=1}^{p}\Phi(\mathbf{x}_n))\right) \tag{27}$$

$$= K_{ij} - \frac{1}{p}\sum_{l=1}^{p}1_{il}K_{lj} - \frac{1}{p}\sum_{n=1}^{p}K_{in}1_{nj} + \frac{1}{p^2}\sum_{l,n=1}^{p}1_{il}K_{ln}1_{nj} \tag{28}$$

$$= (K - 1_pK - K1_p + 1_pK1_p)_{ij}. \tag{29}$$

Thus, we can compute $\tilde{K}$ from K and then solve the eigenvalue problem $\tilde{\lambda}\tilde{\boldsymbol{\alpha}} = \tilde{K}\tilde{\boldsymbol{\alpha}}$. The solutions, $\boldsymbol{\alpha}^k$, are normalized by normalizing the corresponding vectors $\tilde{\mathbf{V}}^k$ in F, which translates into $\tilde{\lambda}_k(\tilde{\boldsymbol{\alpha}}^k \cdot \tilde{\boldsymbol{\alpha}}^k) = 1$.

5.2 The Classifier in MATLAB

Given n data points in $\mathbb{R}^d$ that are considered examples of "normal" behavior, the number of features to retain, q, and parameters needed for the kernel (like σ for the Gaussian), we construct the classifier:

1. Build the $n \times n$ mean subtracted kernel matrix: $\tilde{K}$.

   - Compute the kernel matrix K first:

     $$\texttt{K(i,j)} = \kappa(\mathbf{x}_i, \mathbf{x}_j), \quad \text{for all } 1 \leq i \leq n, \quad 1 \leq j \leq n.$$

   - Compute the mean subtracted kernel matrix. The Matlab code, once K is computed is:

     ```
     Rsum=sum(K,2);
     Asum=sum(sum(K));
     Kt=K-(1/n)*repmat(Rsum',n,1)-(1/n)*repmat(Rsum,1,n)-(1/n^2)*Asum;
     ```

2. Perform an eigenvalue/eigenvector decomposition on $\tilde{K}$. In Matlab, the `eigs` command will compute the first q eigenvectors corresponding to the largest q eigenvalues. Thus, we have q solutions to:

   $$n\lambda^r \boldsymbol{\alpha}^r = \tilde{K}\boldsymbol{\alpha}^r, \qquad \text{for } r = 1, 2, \ldots, q$$

   given in Matlab by the following, where `alpha` is an $n \times q$ matrix holding the eigenvectors, and `lambda` is a diagonal matrix holding the eigenvalues (which are actually denoted as $n\lambda$ in the notes above).

   ```
   opts.disp=0;  %Turn off debugging
   [alpha, lambda]=eigs(Kt,q,'lm',opts);
   ```

   We also need to "normalize" `alpha` so that the eigenvectors have unit length:

   ```
   alpha = alpha * diag( 1./sqrt(diag(lambda)) );
   ```

3. (Optional) Compute the percentage of variance left over after using q feature vectors. In Matlab, this is given by:

   ```
   % residual variance:
    resvar = (trace(Kt)-trace(lambda));
    fprintf('residual variance: %f %%\n',100*resvar/trace(Kt));
   ```

4. Finally, we need a cutoff value for novelty detection. This is the maximum reconstruction error using data classified as "normal".

Closing Remarks

I would like to thank Professor Keef and Professor Hundley at Whitman College for editing and advising me while writing this journal. I would also like to thank Nathan Fisher for editing this journal.

References

[1] B. Lkopf, *Kernel Principal Component Analysis* Advances in Kernel Methods Support Vector Learning, Cambridge, UK, 1999.

[2] D. Hundley, Class notes for Mathematical Modelling, Walla Walla, WA, 2015.

[3] D. Lay, *Linear Algebra and Its Applications,* College Park, MD, 2012.

[4] H. Hoffmann, *Kernel PCA for Novelty Detection,* Pattern Recognition, Munich, Germany, 2007, pp. 863-74.

[5] J. Taylor, N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge UK, 2004.

[6] Machine Learning- Confusion Matrix, Computer Science Source, 2010, `https://computersciencesource.wordpress.com/2010/01/07/year-2-machine-learning-confusion-matrix/`.