Data Preprocessing Tasks

1. Data Cleaning
2. Data Transformation
3. Data Reduction
   We’re here.
4. Discretization
Dimensionality Reduction

• Dimensionality reduction is a commonly used approach for generating fewer features.
• Typically used because too many features can degrade performance.
• Key idea:

  Create a new subset features that are representative of the original features.
Data Preprocessing

**OptDigits Dataset**
Optical Recognition of Handwritten Digits Dataset

- 5,620 instances
  - Samples from 43 people
- 64 features or dimensions
  - $8 \times 8$ input matrix
  - Elements are integers $[0 \ldots 16]$  
- 1 class or target variable
  - A handwritten digit $[0 \ldots 9]$
OptDigits Dataset: Preprocessing

Raw Data

- Normalization
- Convert to bitmap
- Blurring
- Downsampling

Preprocessed Data
OptDigits Dataset: Random Projection

- Notice that the data is *not* well-separated.
- In general, we would prefer for the data to be better-separated, particularly by class (digit).
Key to Principal Component Analysis

A projection should maximize the variation between data, as this captures the “uniqueness” of these data.

Each projection should be made in the direction that captures the most variance—that is, the direction where the data is most spread out—as this variability reflects the internal structure of the data.
Principal Component Analysis (PCA)

• PCA addresses two objectives:
  1. Generally, it is desirable to reduce the number of features used to represent the data.
  2. Generally, it is desirable for the set of features to describe a large amount of “information.”

• PCA attempts to create features or dimensions sequentially based upon the amount of information they capture.
PCA: A Visual Description

- To maximize the amount of information or structure retained, we can try to maximize the variance retained by a projection.
PCA: A Visual Description

- We can observe this variance based upon the relative sparsity or density of the data.
  - Sparser data has greater variance.
  - Denser data has lesser variance.
PCA: A Visual Description

- Our initial projection, $v_1$, should maximize the variance of the data.
PCA: A Visual Description

- The next projection should maximize the remaining variance, while being orthogonal (perpendicular) to the previous projection(s).
  - If the projection is not orthogonal, it will be capturing the same variability already captured by previous projection(s).
PCA: A Visual Description

Original Data

Transformed Data

3D → 2D
PCA: A Visual Description

Transformed Data

Rotated (for display purposes)

New Features ($v_1$ and $v_2$)
PCA: A Mathematical Description

*Let’s begin with how to represent the data.*
Data and Vectors

Feature data is often as a vector, a quantity that has magnitude and direction. For example, the vectors \( \mathbf{a}_1 \) and \( \mathbf{a}_2 \), shown below, each have a length of 3. We’ve illustrated \( \mathbf{a}_1 \) with a direction parallel to the \( y \) axis and \( \mathbf{a}_2 \) with a direction \( 45^\circ \) with respect to the \( x \) axis.

\[
\mathbf{a}_1 = \begin{bmatrix} 2 & 6 & 1 \end{bmatrix} \quad \mathbf{a}_2 = \begin{bmatrix} 7 & 5 & 2 \end{bmatrix}
\]
Data and Matrices

Data from several features are often represented as a *matrix*, a tabular representation of a set of numbers as a collection of rows and columns. For example, the matrix $\mathbf{A}$, shown below, is a 2 by 3 ($2 \times 3$) matrix. The transpose of $\mathbf{A}$ is written as $\mathbf{A}^T$, and is produced by interchanging the rows and columns of $\mathbf{A}$.

$$
\mathbf{A} = \begin{bmatrix}
2 & 6 & 1 \\
7 & 5 & 2
\end{bmatrix}
$$

$$
\mathbf{A}^T = \begin{bmatrix}
2 & 6 & 1 \\
7 & 5 & 2
\end{bmatrix}
$$
Vectors and Matrices

Notes:

• Notationally, it is traditional to represent matrices as bold upper case (i.e., $\mathbf{A}$) and row vectors as bold lower case (i.e., $\mathbf{a}$), with all vectors representing row vectors unless explicitly transposed (i.e., $\mathbf{a}^T$ or $\mathbf{a}^{-1}$) to represent column vectors.

• Also, by convention, a matrix in which all non-diagonal entries are 0 is termed a diagonal matrix.
Eigenvalues and Eigenvectors

The eigenvalues and eigenvectors of an $m$ by $n$ matrix are, respectively, the scalar values $\lambda$ and the vectors $\mathbf{x}$ that are solutions to the following equation:

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

In other words, eigenvectors are the vectors that are unchanged, except for magnitude, when multiplied by $A$.

The basic equation is $A\mathbf{x} = \lambda \mathbf{x}$. The number $\lambda$ is an eigenvalue of $A$. 
Consider matrix $\mathbf{A}$ and vector $\mathbf{x}_1$:

$$
\mathbf{A} = \begin{bmatrix} .8 & .3 \\ .2 & .7 \end{bmatrix}, \quad \mathbf{x}_1 = \begin{bmatrix} .6 \\ .4 \end{bmatrix}
$$

Notice that:

$$
\mathbf{A}\mathbf{x}_1 = \begin{bmatrix} .8 & .3 \\ .2 & .7 \end{bmatrix} \begin{bmatrix} .6 \\ .4 \end{bmatrix} = \mathbf{x}_1 \quad \text{(Ax = x means that $\lambda_1 = 1$)}
$$

$$
\mathbf{A}\mathbf{x}_2 = \begin{bmatrix} .8 & .3 \\ .2 & .7 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} .5 \\ -.5 \end{bmatrix} \quad \text{(this is $\frac{1}{2}\mathbf{x}_2$, so $\lambda_2 = \frac{1}{2}$)}
$$

Note that if $\mathbf{x}_1$ is multiplied again by $\mathbf{A}$, we still get $\mathbf{x}_1$. 

Eigenvalues and Eigenvectors
Eigenvalues and Eigenvectors

$A^2$ has eigenvalues $1^2$ and $(.5)^2$

$\lambda = 1$

$A x_1 = x_1 = \begin{bmatrix} .6 \\ .4 \end{bmatrix}$

$\lambda^2 = 1$

$A x_1 = x_1 = \begin{bmatrix} .6 \\ .4 \end{bmatrix}$

$\lambda = .5$

$A x_2 = \lambda_2 x_2 = \begin{bmatrix} .5 \\ -.5 \end{bmatrix}$

$\lambda^2 = .25$

$A^2 x_2 = (.5)^2 x_2 = \begin{bmatrix} .25 \\ -.25 \end{bmatrix}$

$x_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

The eigenvectors keep their directions, whereas all other vectors change direction. But note that all other vectors are combinations of the (two) eigenvectors.
Recall that the variance is the most common measure of the spread of a set of points:

\[ Var(x) = s_x^2 = \frac{1}{m - 1} \sum_{i=1}^{m} (x_i - \bar{x})^2 \]
Covariance

Recall that the covariance is a measure of the degree to which two variables vary together, and is given by:

\[ \text{Cov}(x_i, x_j) = \frac{1}{m - 1} \sum_{i=1}^{m} (x_{ki} - \bar{x_i})(x_{kj} - \bar{x_j}) \]

where \( x_{ki} \) and \( x_{kj} \) are the values of the \( i^{th} \) and \( j^{th} \) feature vectors for the \( k^{th} \) object.
Covariance Matrices

The covariance of two feature vectors $X$ and $Y$ is given by a covariance matrix

$$c_{XY} = \begin{bmatrix} Cov(X, X) & Cov(X, Y) \\ Cov(Y, X) & Cov(Y, Y) \end{bmatrix}$$

where $Cov(X, X) = Var(X)$, $Cov(Y, Y) = Var(Y)$, and $Cov(X, Y) = Cov(Y, X)$. 
Covariance Matrices

The covariance matrix of $n$ feature vectors is given as

$$C_X = \begin{bmatrix}
\text{Cov}(X_1, X_1) & \text{Cov}(X_1, X_2) & \ldots & \text{Cov}(X_1, X_n) \\
\text{Cov}(X_2, X_1) & \text{Cov}(X_2, X_2) & \ldots & \text{Cov}(X_2, X_n) \\
\vdots & \vdots & \ddots & \vdots \\
\text{Cov}(X_n, X_1) & \text{Cov}(X_n, X_2) & \ldots & \text{Cov}(X_n, X_n)
\end{bmatrix}$$

where $\text{Cov}(X_i, X_j) = \text{Cov}(X_j, X_i)$ and $\text{Cov}(X_i, X_i) \geq 0$. 
Covariance Matrix Decomposition

The covariance matrix $\mathbf{C}_X$ can be decomposed as follows:

$$\mathbf{C}_X = \mathbf{\Phi} \mathbf{\Lambda} \mathbf{\Phi}^{-1} \quad \mathbf{\Phi}^{-1} = \mathbf{\Phi}^T$$

Thus, $\mathbf{C}_X$ can be decomposed as the product of three matrices. $\mathbf{\Phi}$ is known as the eigenvector matrix and $\mathbf{\Lambda}$ as the eigenvalue matrix. Specifically, the columns of $\mathbf{\Phi}$ are the eigenvectors of $\mathbf{C}_X$, while the diagonal elements of $\mathbf{\Lambda}$ are the eigenvalues of $\mathbf{C}_X$. 
From Linear Algebra to PCA

So what does this all have to do with PCA?
The Goal of PCA

• A goal of PCA is to find a transformation of the data that satisfies the following properties:
  – Each pair of new features has 0 covariance.
  – The features are ordered with respect to how much of the variance of the data each feature captures.
  – The first feature captures as much of the variance of the data as possible.
  – Each successive feature captures as much of the remaining variance as possible, as long as it is orthogonal to the previous components.
Covariance Matrix for PCA

Given a dataset represented as an \( m \) by \( n \) matrix \( \mathbf{D} \):

- Centralize the data (subtract the mean).
- Calculate the covariance matrix \( \mathbf{C} \), where each entry of \( \mathbf{C} \) is:
  \[
  c_{ij} = Cov(d_{*i}, d_{*j})
  \]
  where \( i \) and \( j \) are features.
- Then, with each feature of \( \mathbf{D} \) normalized to have a mean of 0, \( \mathbf{C} = \mathbf{D}^T \mathbf{D} \). In words, if the feature means of \( \mathbf{D} \) are 0, then the covariance matrix \( \mathbf{C} \) is equal to the product of \( \mathbf{D} \) and its transpose, \( \mathbf{D}^T \).
PCA: A Mathematical Description

As a covariance matrix, \( C \) can be decomposed into an eigenvector matrix and an eigenvalue matrix. Thus, with \( C = D^T D \):

- Let \( \lambda_1 \ldots \lambda_n \) be the eigenvalues of \( C \). The eigenvalues can be ordered such that \( \lambda_1 \geq \lambda_2 \ldots \geq \lambda_m \).
- Let \( \Phi = [\phi_1, \ldots, \phi_n] \) be the matrix of eigenvectors of \( C \), ordered so that the \( i^{\text{th}} \) eigenvector corresponds to the \( i^{\text{th}} \) largest eigenvalue.
PCA: A Mathematical Description

Then PCA computes:

\[ D' = D\Phi \]

- The data matrix \( D' \) is the set of transformed data that satisfies the posed goals of PCA.
- In words, \( D' \) is the matrix that results from taking the product of the original data matrix, \( D \), with \( \Phi \), the eigenvector matrix of \( C \). Recall that \( C \) is the covariance matrix of \( D \) (the product of \( D \) and its transpose, \( D^T \)).
PCA: A Mathematical Description

• The eigenvectors represent linear combinations of the original features.
  – Each is a new axis or dimension.

• The eigenvalues are measures of the amount of variance captured by the eigenvectors.
  – Each eigenvector has a corresponding eigenvalue.
OptDigits Dataset: PCA Projection

- Notice that the data is well-separated.
- In general, each class (digit) is well-separated from the others.
OptDigits Dataset: A Comparison

Random Projection (2D)  
PCA Projection (2D)
How Many Principal Components?

- If all of the principal components are kept, then there is no data reduction.
- Rather, the objective is to *minimize* the number of retained principal components.
- *Scree Plots* can help determine the number of components to retain.
Scree Plots

• A Scree Plot is a simple line segment plot that shows the fraction of total variance in the data as explained or represented by each principal component.
• The point where there is a significant drop in the explained variance is sometimes called the “knee” or “elbow” point of the plot.
OptDigits Dataset: PCA Scree Plot
OptDigits Dataset: PCA Scree Plot

```
Principal Component | Variance Explained (%) |
--- | --- |
1 | 19% |
2 | 16% |
3 | 11% |
4 | 8%  |
5 | 5%  |
6 | 4%  |
7 | 3.5%|
8 | 3.1%|
```

"Knee point"
OptDigits Dataset: A Fun PCA Scree Plot

"Knee point"
Advantages of PCA

• By generating new features that maximize the variance in the data, the data points can often remain well-separated.

So what’s the catch? Or is class over?
Limitations of PCA

- PCA does not necessarily help segmenting or separating data.
- PCA generates principal components, which are linear combinations of the original features.
  - Non-linear structure may not be captured.
Limitations of PCA: Separating Data

PCA does *not* necessarily help segmenting or separating data, and thus does not necessarily improve modeling.

Well-separated Data

First Principal Component

Resulting Transformation (Not well-separated data)
Limitations of PCA: Separating Data

In contrast, a better projection might look like the following, which maintains separation of the data.

Well-separated Data

A Potential Projection

Resulting Transformation (Well-separated data)
Swiss Roll Dataset
A Synthetic Non-linear Dataset

• 1,500 instances
• 3 features or dimensions
  – $X$ is based on cosine function
  – $Y$ is random real value [0 ... 21]
  – $Z$ is based on sine function
• Looks delicious
Limitations of PCA: Non-linear Data

Swiss Roll Data Projection (2D)

- Notice that as a result of PCA, the projected data is *not* well-separated.
- The principal components maximize the variance, but not necessarily the separation of the data.
Local Linear Embedding (LLE)

Reduce dimensionality by analyzing overlapping local neighborhoods to determine local structure:

1. Find the nearest neighbors of each data point.
2. Express each point $x_i$ as a linear combination of the other points, i.e., $x_i = \sum_j w_{ij} x_j$, where $w_{ij} = 0$ if $x_j$ is not a near neighbor of $x_i$.
3. Find the coordinates of each point in lower-dimensional space of specified dimension $p$ by using weights found in step 2.
LLE Projection (Swiss Roll Dataset)

Swiss Roll Data Projection (2D)

- Notice that as a result of LLE, the projected data is well-separated.
- LLE attempts to retain local structure.
Summarizing Data
Preprocessing
Summarizing Data Preprocessing

• Preprocessing is informed by data understanding.

• *Data cleaning* involves the correction of data quality problems, including imputing missing data, smoothing noisy data, and removing outliers and inconsistencies.

• *Data transformation* involves the aggregation, generalization, and normalization of data, and the possible construction of new features from old ones.

• *Data reduction* involves the sampling of data, selection of features, and reduction of dimensionality.
And Now...

Let’s see some data reduction!