LEHIGH UNIVERSITY

Introduction to Dimensionality Reduction

ISE – 364 / 464

DEPT. OF INDUSTRIAL & SYSTEMS ENGINEERING

GRIFFIN DEAN KENT

Overview of Dimensionality Reduction

Often when dealing with datasets with many features, several of those features will only contribute a marginal amount of predictive capability. This is one of many different types of problems that dimensionality reduction techniques aim to address.

Dimensionality Reduction (DR)

These are methods that attempt to condense the most amount of information contained within a dataset into a smaller number of new features. These methods are useful for getting rid of features that do not have much predictive capability, increase computational efficiency of working with the dataset, aid in data visualization, supplement and enable more effective use of unsupervised clustering methods, as well as address a problem known as "**the curse of dimensionality**".

- **The Curse of Dimensionality:** This is a term coined for a phenomenon that arises when dealing with datasets that have a **small number of datapoints relative to the number of features in the dataset** (i.e., when one is dealing with many more features than datapoints). Specifically, in ML problems with a high-dimensional feature space typically an enormous amount of training data is required to ensure that there are enough samples that are representative of each combination of possible values. Typically, as the **dimensionality grows**, the **amount of data that is required** to create models that can generalize accurately **grows exponentially**.

Overview of Embeddings

A concept of paramount importance for dimensionality reduction techniques, and many other areas of machine learning, is the idea of numeric embeddings.

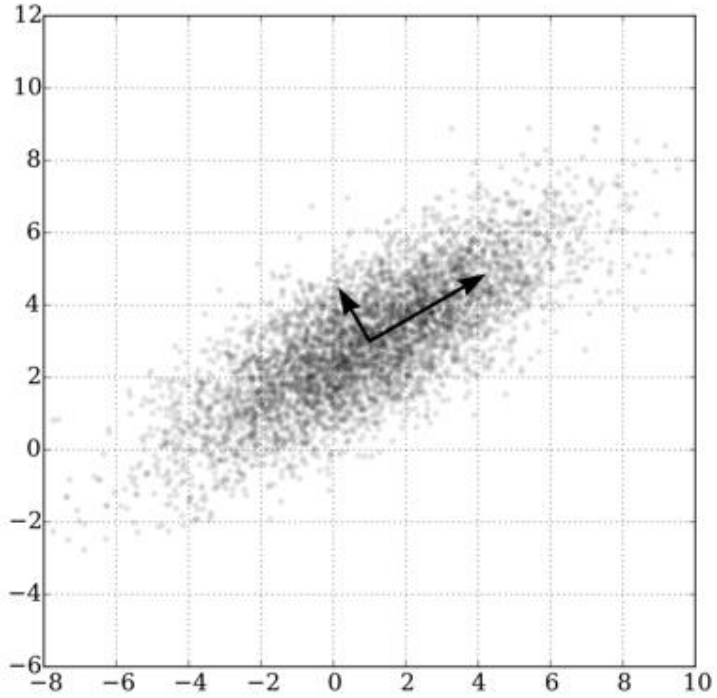
Embeddings

An embedding is a way of **representing some type of complex data** (such as high-dimensional tabular data, images, words, sounds, etc.) as a **numerical representation** in some new or different **vector space**. In terms of dimensionality reduction techniques, these embeddings are typically representations that "encode" some sort of key features, patterns, or relationships of the original data in a way that reduces its dimensionality. At a fundamental level, good embeddings are vector spaces with spatial regions that **represent some sort of semantic meaning**.

Popular Dimensionality Reduction Techniques

- **Principal Component Analysis (PCA):** The most fundamental of all DR techniques – generates the top vectors (principal components) in the directions of maximum variance of a dataset which are orthogonal. These vectors also correspond to the eigenvectors of the covariance with the largest eigenvalues
- **Linear Discriminant Analysis (LDA):** A supervised learning method that projects data into a new embedded space that maximizes the class separability of the target labels.
- **T-Distributed Stochastic Neighbor Embeddings (TSNE):** A nonlinear technique that maps high-dimensional data into (typically) 2D or 3D spaces while preserving local neighborhood relationships.
- **Uniform Maniform Approximation and Projection (UMAP):** A nonlinear technique that reduces high-dimensional data by preserving both global and local structure of data by utilizing manifold learning.

Principal Component Analysis



Example of the two principal component vectors of a 2-dimensional dataset.

Principal Component Analysis (PCA)

PCA is perhaps the most well-known dimensionality reduction technique. At a high level, PCA transforms a dataset into a lower-dimensional embedding where the new axis (i.e., the features of the newly embedded space) are orthogonal to each other and are ordered by the amount of variance that they explain in the original dataset. In this way, the "**first PCA**" corresponds to the direction vector (i.e., the axis or feature in the embedded space) that explains the most variance in the original data, the "**second PCA**" explains the second most variance, and so on and so fourth.

Formally, consider a dataset $\mathcal{D} = \{x^{(i)}\}_{i=1}^{m}$ with a design matrix given by $X \in \mathbb{R}^{m \times n}$ where each of the features are standardized (they have a mean of 0 and a standard deviation of 1), which will ensure that all the features are on the "same scale" (this is a **required assumption** when dealing with PCA).

PCA learns two linear transformations $W \in \mathbb{R}^{c \times n}$ and $U \in \mathbb{R}^{n \times c}$:

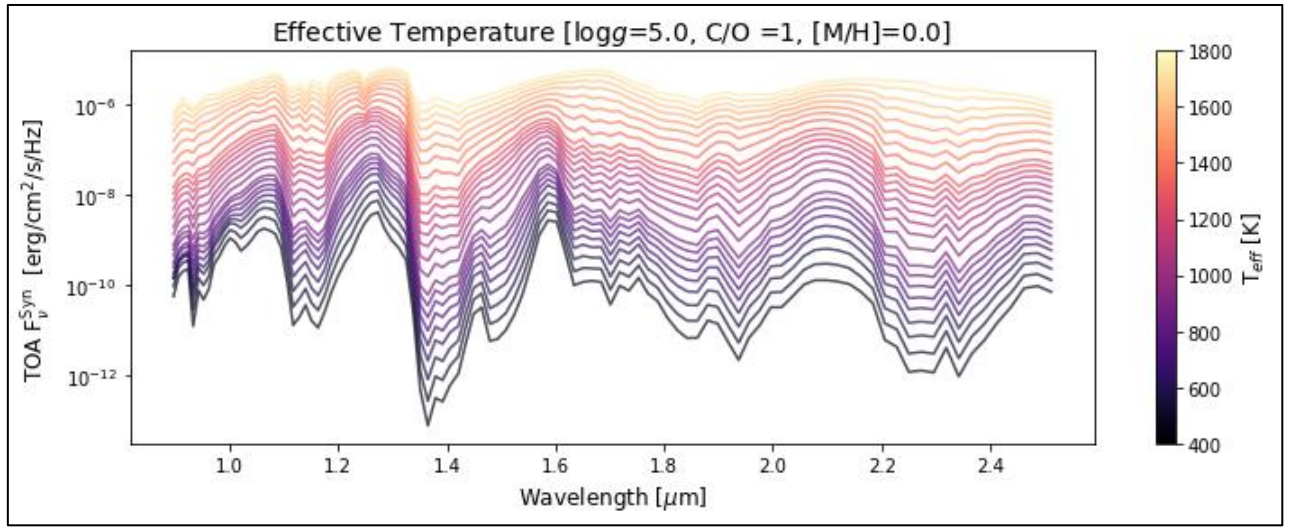
- The **embedding matrix** $W \in \mathbb{R}^{c \times n}$, which maps the dataset X into some lower dimensional dataset $Z = WX^T \in \mathbb{R}^{c \times m}$ such that $c < n$.
- The **recovery matrix** $U \in \mathbb{R}^{n \times c}$, which is used to approximately recover the original dataset X as the approximation $\tilde{X} = Z^T U^T \in \mathbb{R}^{m \times n}$.

PCA can be formulated as two different, yet equivalent problems: a **minimum error formulation (MEF)** and a **maximum variance formulation (MVF)**.

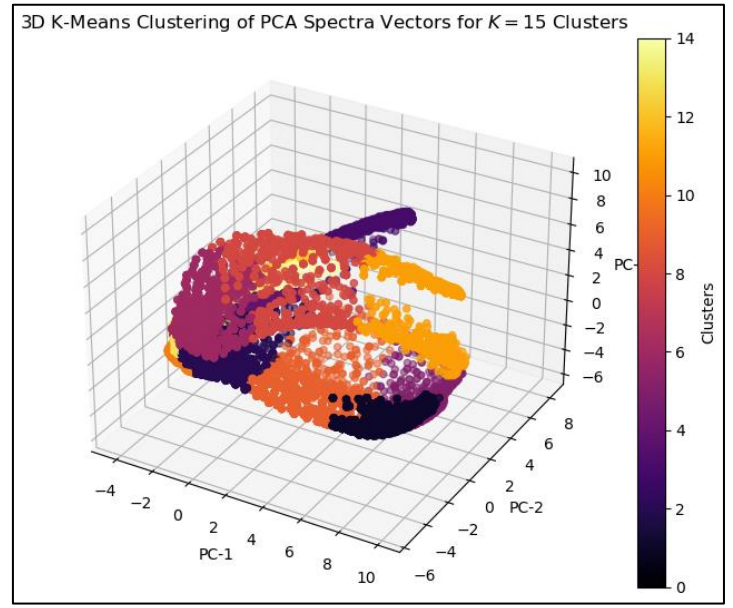
Illustration of PCA

The following is an example of how NASA can use dimensionality reduction to help analyze useful patterns about interstellar objects simply from the infrared spectra that are emitted from the objects. Specifically, this example visualizes a dataset that consists of 104 wavelengths (a dataset with 104 features) with the top 3 PCs.

Original synthetic "sequence" dataset of 104 spectral wavelengths emitted from Brown Dwarf celestial objects



Scatterplot of the top 3 Principal Components of the spectra dataset (Clustered by K-Means)



Minimum Error Formulation of PCA

Minimum Error Formulation (MEF) of PCA

The first formulation of PCA we will introduce is perhaps the most intuitive; namely, we wish to determine the matrices $W \in \mathbb{R}^{c \times n}$ and $U \in \mathbb{R}^{n \times c}$ that minimize the error between an original datapoint $x \in \mathbb{R}^n$ and its approximated recovery vector $\tilde{x} \in \mathbb{R}^n$. Naturally, one can formulate this problem by utilizing the MSE loss function over the entire dataset $\mathcal{D}$, yielding the following optimization problem:

$$\underset{W,U}{\operatorname{argmin}} \sum_{i=1}^{m} \left\| x^{(i)} - \tilde{x}^{(i)} \right\|_2^2 = \underset{W,U}{\operatorname{argmin}} \sum_{i=1}^{m} \left\| x^{(i)} - UWx^{(i)} \right\|_2^2 .$$

One could write this in matrix-vector notation as

$$\underset{W,U}{\operatorname{argmin}} \left\| X^T - \tilde{X}^T \right\|_2^2 = \underset{W,U}{\operatorname{argmin}} \| X^T - UWX^T \|_2^2 .$$

MEF Reformulation

It can be shown that the minimum error formulation (MEF) of PCA is equivalent to the following problem, posed solely in terms of the matrix $U \in \mathbb{R}^{n \times c}$:

$$\underset{U}{\operatorname{argmin}} \sum_{i=1}^{m} \left\| x^{(i)} - UU^T x^{(i)} \right\|_2^2 = \underset{U}{\operatorname{argmin}} \| X^T - UU^T X^T \|_2^2 .$$

It follows from this formulation that the matrix U is orthonormal (its columns are orthogonal to each other and are normalized) and $W = U^T$. An implication of this is that $U^T U = I$.

Maximum Variance Formulation of PCA

Under the assumption that the features of the dataset are standardized, i.e., for a feature x_j the mean of that random variable is $\mu_j = 0$, then the variance of that feature is given by

$$\mathrm{Var}(x_j) = \mathbb{E}\left[(x_j - \mu_j)^2\right] = \mathbb{E}[{x_j}^2].$$

- However, this is the **population variance**. Since we are dealing with a dataset that is drawn from some unknown underlying distribution, we would utilize the **sample variance** as

$$\mathrm{Var}(x_j) = \frac{1}{m}\sum_{i=1}^{m}(x_j - \mu_j)^2 = \frac{1}{m}\sum_{i=1}^{m}{x_j}^2,$$

It bears mentioning that when sample sizes are somewhat small, the denominator is typically set to $m-1$ (Bessel's correction) and aims to correct for the bias in estimating the population variance; however, for our purposes (and when the dataset is large) using m is sufficient.

Maximum Variance Formulation of PCA

Let $\Sigma = \frac{1}{m}X^TX = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}{x^{(i)}}^T \in \mathbb{R}^{n\times n}$ denote the covariance matrix corresponding to the dataset X. Then, one can also formulate the PCA problem as learning an **orthonormal** matrix $U \in \mathbb{R}^{n\times c}$ such that the variance of the newly embedded data $Z = U^TX^T$ is maximized. This can be written mathematically as the problem (notice that $(1/m)ZZ^T$ is the covariance matrix of the embedded dataset)

$$\max_{U:U^TU=I} \mathrm{trace}\left(\frac{1}{m}ZZ^T\right) = \max_{U:U^TU=I} \mathrm{trace}(U^T\Sigma U),$$

where the trace operation sums the elements along the diagonal of the resulting matrix.

- It can be shown that **this problem is equivalent to the MEF of PCA** and visa versa.

Solution to PCA

Now that we have proposed two different forms of the PCA problem, to demonstrate that they are equivalent all one must do is analyze the solutions to both, which are the same.

The Solution to the MEF and MVF for PCA

Let $\Sigma = \frac{1}{m} X^T X$ denote the covariance matrix of the original data $X \in \mathbb{R}^{m \times n}$ and let $\{p_k\}_{k=1}^{c}$ denote the set of eigenvectors of Σ with the c largest eigenvalues (such that $c < n$).

- Then, the **optimal solution to both the ME and MV formulations of PCA** is the orthonormal matrix $U \in \mathbb{R}^{n \times c}$ such that the columns of U are the **eigenvectors** that correspond to the c largest eigenvalues of the covariance matrix Σ, i.e., $U \coloneqq [p_1, p_2, \dots, p_c]$, and where $W = U^T$.
- Further, the **amount of variance** in the original dataset X that is **explained** by projecting the data onto the lower-dimensional ($c < n$) embedded space is equivalent to the sum of the top-c eigenvalues $\sum_{k=1}^{c} \lambda_k = \text{trace}(U^T \Sigma U)$ (where $\lambda_k \in \mathbb{R}$). Therefore, each eigenvalue λ_k defines the amount of variance that explained by each of the corresponding eigenvectors (which are the new features of the embedded dataset).
- Therefore, if one were to use all the eigenvectors to project onto, then the dimension of the embedded space would be $c = n$ (and one would not reduce the number of dimensions) but would retain 100% of the explained variance of the original data.

How Many PCs to Choose?

Ultimately, the solution to this question is dependent on the goal of performing PCA, but here are some considerations:

- Optimize the trade-off between explained variance and number of PCs by utilizing "elbow-plots" (see slides on K-means).
- If one is simply trying to visualize a high-dimensional dataset, you will typically just choose either the top 2 or 3 PCs to plot.
- Often, PCA is used as a pre-stage method before performing a more advanced nonlinear dimensionality reduction technique such as TSNE or UMAP (as these are much more computationally expensive methods).