Contrastive Principal Component Analysis

Maths Behind Machine Learning (MA6100)
Project

Rishit Singh (CE20BTECH11032)
Sankalp Deshmukh (CE20BTECH11034)

Reference Paper: Contrastive Principal Component Analysis

1 Introduction

Contrastive principal component analysis (cPCA) is a technique that is designed to discover low-dimensional structure that is unique to a dataset, or enriched in one dataset relative to other data. The technique is a generalization of standard PCA, for the setting where multiple datasets are available – e.g. a treatment and a control group, or a mixed versus a homogeneous population – and the goal is to explore patterns that are specific to one of the datasets.

2 The cPCA Algorithm

Let $\{X_i\}_{i=1}^n$ and $\{Y_j\}_{j=1}^m$ be two datasets where $X_i, Y_j \in \mathbb{R}^d$. We are interested in finding patterns that are enriched in the X_i's relative to the Y_j's. We refer to $\{X_i\}$ as the target data and $\{Y_j\}$ as the background data. Without loss of generality, we assume the data have been centered and use C_X and C_Y to denote their respective empirical covariance matrices.

Let $\mathbb{R}^d_{\text{unit}}$ be the set of vectors in $\mathbb{R}^d$ with unit norm. For any direction $v \in \mathbb{R}^d_{\text{unit}}$, its corresponding variances in the target and in the background can be written as

$$\text{Target variance: } v^T C_X v, \text{Background variance: } v^T C_Y v.$$

The goal of cPCA is to identify directions v which account for large variances in the target and small variances in the background. Specifically, cPCA solves the following optimization problem:

$$\arg\max_{v \in \mathbb{R}^d_{\text{unit}}} v^T (C_X - \alpha C_Y) v,$$

where $\alpha \in [0, \infty]$ is a parameter discussed later.

The optimisation problem can be efficiently solved by conducting an eigenvalue decomposition on the matrix $C := (C_X - \alpha C_Y)$ and returning the eigenvectors corresponding to the leading eigenvalues. Analogously to PCA, we call the leading eigenvectors the contrastive principal components (cPCs), and we return the subspace spanned by the first few (typically two) orthogonal cPCs. For a suitable α, projecting $\{X_i\}$ onto this subspace provides insight into structure specific to data.

The contrast parameter α represents the trade-off between maximizing the target variance and minimizing the background variance. When $\alpha = 0$, cPCA selects the directions that only maximize the target variance, and hence reduces to PCA applied on $\{X_i\}$. As α increases, directions that reduce the background variance become more optimal, and the contrastive principal components are driven towards the null space of the covariance matrix of $\{Y_i\}$. When $\alpha = \infty$, any direction not in the null space of the background data receives an infinite penalty. In this case, cPCA is reduced to first projecting the target data on the null space of the background and then performing PCA on the projected data. Therefore, each value of α yields a direction with a different trade-off between target and background variance.

2.1 Choosing the background dataset

The additional insight that cPCA reveals about a target dataset is based on characteristics of the background, leading us to ask: how do we choose a good contrastive background dataset? In general, the background should be chosen to have the structure that we want to remove from the target data. This structure may correspond to variables that we are not interested in but may have significant variation in the target data.

3 Applications of Contrastive PCA

As a general unsupervised learning technique, cPCA (like PCA) has a variety of uses in data exploration and visualization. Here, we show two applications where cPCA can provide additional insight into the structure of a dataset that is missed by standard PCA.

Demographically-Diverse Cancer Patients. Suppose we have gene-expression measurements from individuals of different ethnicities and sexes. This data includes geneexpression levels of cancer patients $\{X_i\}$, which we are interested in analyzing. We also have control data, which corresponds to the gene-expression levels of healthy patients $\{Y_i\}$ from a similar demographic background. Our goal is to find trends and variations within cancer patients (e.g. to identify molecular subtypes of cancer).

If we directly apply PCA to $\{X_i\}$, however, the top principal components may correspond to the demographic variations of the individuals instead of the subtypes of cancer because the genetic variations due to the former are likely to be larger than that of the latter. As we show, we can overcome this problem by noting that the healthy patients also contain the variation associated with demographic differences, but not the variation corresponding to subtypes of cancer. Thus, we can search for components in which $\{X_i\}$ has high variance but $\{Y_i\}$ has low variance.

Handwritten Digits on Complex Backgrounds. As another example, consider a dataset $\{X_i\}$ that consists of handwritten digits on a complex background, such as different images of grass. A typical unsupervised learning task may be to cluster the data according to the digits in the image. However, if we perform standard PCA on these images, we find that the top principal components do not represent features related to the handwritten digits but reflect the dominant variation in features related to the image background.

We will show that it is possible to correct for this by using a reference dataset $\{Y_i\}$ that consists solely of images of the grass (not necessarily the same images used in $\{X_i\}$ but having similar covariance between features), and looking for the subspace of higher variance in $\{X_i\}$ compared to $\{Y_i\}$. By projecting onto this subspace, we can actually visually separate the images based on the value of the handwritten digit.

3.1 Simulation Results for Handwritten Digits on Complex Backgrounds

We create a target dataset of 12,665 synthetic images by randomly superimposing images of handwritten digits 0 and 1 from the MNIST dataset on top of images of grass taken from Kaggle. The images of grass are converted to grayscale, resized to be 100x100, and then randomly cropped to be the same size as the MNIST digits, 28x28. (Figure 1)

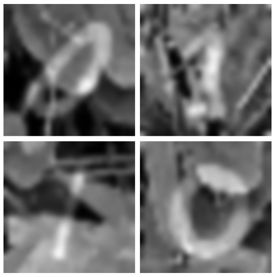

Figure 1: Target Dataset

A background dataset is then introduced consisting solely of images of grass belonging to the same set of kaggle dataset, but we use images that are different than those used to create the target dataset. (Figure 2)

Then, we plot the result of embedding the synthetic images onto their

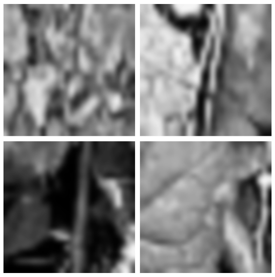

Figure 2: Background Dataset

first two principal components using standard PCA. We see that the lower-dimensional embeddings of the images with 0s and images with 1s are hard to distinguish. (Figure 3)

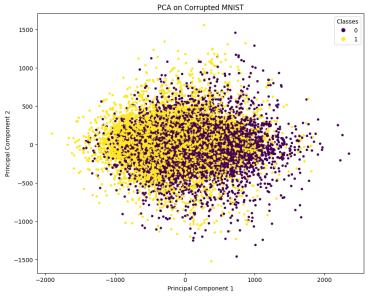

Figure 3: PCA

Using cPCA on the target and background datasets, (with a value of the contrast parameter α set to 2.0), two clusters emerge in the lower-dimensional representation of the target dataset, one consisting of images with the digit 0 and the other of images with the digit 1. (Figure 4)

Link to code implementation `https://colab.research.google.com/drive/1ztLJBM6zZFaJfM6CTErkNy6m2wHIG91D?usp=sharing`

4 Extensions: Kernel cPCA

We extend cPCA to Kernel cPCA, following the analogous extension of PCA to kernel PCA.

Consider the nonlinear transformation $\Phi : \mathbb{R}^d \to F$ that maps the data to some feature space F. We assume that the mapped data, $\Phi(X_1), \ldots, \Phi(X_n), \Phi(Y_1), \ldots, \Phi(Y_m)$, is centered, i.e., $\sum_{i=1}^{n} \Phi(X_i) = \sum_{j=1}^{m} \Phi(Y_j) = 0$. The covariance matrices for

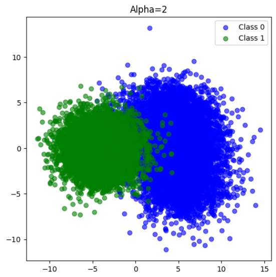

Figure 4: cPCA

the target and the background can be written as

$$C_{\bar{X}} = \frac{1}{n}\sum_{i=1}^{n}\Phi(X_i)\Phi(X_i)^T, \quad C_{\bar{Y}} = \frac{1}{m}\sum_{j=1}^{m}\Phi(Y_j)\Phi(Y_j)^T.$$

Contrastive PCA on the transformed data solves for the eigenvectors of $(C_{\bar{X}} - \alpha C_{\bar{Y}})v$, where the k-th eigenvector is the k-th contrastive component, but this is inefficient if the dimensionality of F is large.

We next describe the kernel cPCA algorithm, which allows us to efficiently perform contrastive analysis on the transformed data.

Let $N = n{+}m$ and denote the data as $(Z_1, \ldots, Z_N) = (X_1, \ldots, X_n, Y_1, \ldots, Y_m)$. Define the kernel matrix K to have the ij-th element $K_{ij} = \Phi(Z_i)\cdot\Phi(Z_j)$, and write it in the form of a block matrix as

$$K = \begin{bmatrix} K_X & K_{XY} \\ K_{YX} & K_Y \end{bmatrix},$$

where $K_X \in \mathbb{R}^{n\times n}$, $K_Y \in \mathbb{R}^{m\times m}$ are the sub-kernels corresponding to $X_1, \ldots, X_n$, and $Y_1, \ldots, Y_m$, respectively.

As derived in Appendix E, instead of directly calculating the eigenvectors of $(C_{\bar{X}} - \alpha C_{\bar{Y}})v$, we can consider its dual representation $v = \sum_{i=1}^{N} a_i\Phi(Z_i)$, and solve a_i's via the following eigenvalue problem for non-zero eigenvalues:

$$\lambda a = \widetilde{K}a,$$

where the first eigenvector $a^{(1)}$ corresponds to the first contrastive component, and

$$\widetilde{K} = \begin{bmatrix} \frac{1}{n}K_X & \frac{1}{n}K_{XY} \\ -\frac{\alpha}{m}K_{YX} & -\frac{\alpha}{m}K_Y \end{bmatrix}.$$

To make $\|v\| = 1$, we require $a^Ta = 1$. Finally, we can project the data onto the k-th contrastive component by

$$[v^{(k)}\cdot\Phi(Z_1), \ldots, v^{(k)}\cdot\Phi(Z_N)] = K^{(k)}a^{(k)}.$$

Note that in the above calculation, the kernel can be constructed via some kernel function $h(\cdot, \cdot)$ as $K_{ij} = h(Z_i, Z_j)$, and the projected data can be computed as $K^{(k)}a^{(k)}$. As a result, by Kernel cPCA, we can actually perform cPCA in the feature space without explicitly computing the nonlinearly transformed data.

5 Conclusion

The main advantages of cPCA is its generality and its easiness of use. Computing a particular contrast PCA takes essentially the same amount of time as computing a regular PCA. This computational efficiency enables cPCA to be useful for interactive data exploration, where each operation should ideally be almost immediate. Moreover any data where PCA can be usefully applied, cPCA can also be applied.