**CCA Introduction to Can Corr An in Python**

December 25, 2020 by [cmdlinetips](https://cmdlinetips.com/author/cmdline/)

<https://cmdlinetips.com/2020/12/canonical-correlation-analysis-in-python/>

Increasingly, we have multiple high dimensional datasets from from the same samples. Canonical Correlation Analysis aka CCA is great for scenarios where you two high dimensional datasets from the same samples and it enables learning looking at the datasets simultaneously. A classic example is audio and video datasets from the same individuals. One can also think of CCA is another dimensionality reduction technique like Principal Component Analysis. Unlike PCA, you deal with two datasets in CCA instead of one.

If you are interested in a bit of history CCA is originally developed by the same [Hotelling who developed PCA](https://cmdlinetips.com/2020/12/canonical-correlation-analysis-in-r/) in the 1930s.

What is Canonical Correlation Analysis?

In this tutorial, we will see examples of how to perform CCA using Palmer Penguins data set. We will use scikit-learn to perform Canonical Correlation Analysis (CCA). We will not go into the math behind CCA, instead we will see a practical example of doing CCA and understand the main intuition behind the results.

As we explained in the previous post on [performing CCA with R](https://cmdlinetips.com/2020/12/canonical-correlation-analysis-in-r/), one can understand the idea behind CCA as follows

Let us say, there is one or more variables generating two high-dimensional data sets X and Y. Here, the data sets X and Y are observables. And we don’t know about the latent variable(s) behind the two data sets. Since both the datasets are from the latent variable, there will be a lot shared or common variation in the two datasets. By doing CCA, we can identify the common variation, the canonical variates that are highly correlated to the unknown latent variable.

Typically, the two datasets might have other variation in addition to the variation due to the main latent variable. And, CCA helps us remove the data specific variation or noise in the two datasets and gets to the canonical variable that captures the hidden variable.

Palmer Penguin Dataset for Canonical Correlation Analysis

To get started, let us load Pandas, Matplotlib, numpy, and Seaborn.

|  |  |
| --- | --- |
| 1234 | import pandas as pdimport matplotlib.pyplot as pltimport seaborn as snsimport numpy as np |

Let us load Palmer Penguin datasets and do a a bit bit of cleanup by removing any rows with missing values.

|  |  |
| --- | --- |
| 1234 | link2data = "<https://raw.githubusercontent.com/mwaskom/seaborn-data/master/penguins.csv>"df = pd.read\_csv(link2data)df =df.dropna()df.head() |

Since Canonical Correlation Analysis involve two high dimensional data set sets, we will split the Penguin’s data into two datasets X and Y. The dataset X has two variables corresponding to bill length and depth.

|  |  |
| --- | --- |
| 12 | X = df[['bill\_length\_mm','bill\_depth\_mm']]X.head() |
| 123456 | bill\_length\_mm  bill\_depth\_mm0   39.1    18.71   39.5    17.42   40.3    18.04   36.7    19.35   39.3    20.6 |

We also need to [standardize the variables](https://cmdlinetips.com/2020/12/compute-standardized-values-z-score-python/) by subtracting with mean and dividing by standard deviation.

|  |  |
| --- | --- |
| 123456789 | X\_mc = (X-X.mean())/(X.std())X\_mc.head()     bill\_length\_mm  bill\_depth\_mm0   -0.894695   0.7795591   -0.821552   0.1194042   -0.675264   0.4240914   -1.333559   1.0842465   -0.858123   1.744400 |

The second dataset Y contains flipper length and body weight.

|  |  |
| --- | --- |
| 12 | Y = df[['flipper\_length\_mm','body\_mass\_g']]Y.head() |

Let us standardize the Y data set as well.

|  |  |
| --- | --- |
| 12345678 | Y\_mc = (Y-Y.mean())/(Y.std())Y\_mc.head()    flipper\_length\_mm   body\_mass\_g0   -1.424608   -0.5676211   -1.067867   -0.5055252   -0.425733   -1.1885724   -0.568429   -0.9401925   -0.782474   -0.691811 |

Canonical Correlation Analysis with Scikit-learn in Python

Now we have two datasets from the same Penguins. As we know, the features of a Penguin differ strongly from another based on what species it is. And in our toy example, species is a latent variable that is common behind the two datasets X and Y.

To perform CCA in Python, We will use CCA module from sklearn.cross\_decomposition.

|  |  |
| --- | --- |
| 1 | from sklearn.cross\_decomposition import CCA |

First, we instantiate CCA object and use fit() and transform() functions with the two standardized matrices to perform CCA.

|  |  |
| --- | --- |
| 123 | ca = CCA()ca.fit(X\_mc, Y\_mc)X\_c, Y\_c = ca.transform(X\_mc, Y\_mc) |

And our result is two canonical correlate matrices.

|  |  |
| --- | --- |
| 1234 | print(X\_c.shape)print(Y\_c.shape)(333,2)(333,2) |

Understanding the results from Canonical Correlation Analysis

Let us dig deeper to understand the results from canonical correlation analysis and get an intuition. At first, we will use the pair of canonical covariates and see how are they correlated to each other.

Let us add the results from CCA with the meta data corresponding to Penguins data in to a Pandas dataframe.

|  |  |
| --- | --- |
| 1234567 | cc\_res = pd.DataFrame({"CCX\_1":X\_c[:, 0],                       "CCY\_1":Y\_c[:, 0],                       "CCX\_2":X\_c[:, 1],                       "CCY\_2":Y\_c[:, 1],                       "Species":df.species.tolist(),                      "Island":df.island.tolist(),                      "sex":df.sex.tolist()}) |
| 1234567 | cc\_res.head()    CCX\_1   CCY\_1   CCX\_2   CCY\_2   Species Island  sex0   -1.186252   -1.408795   -0.010367   0.682866    Adelie  Torgersen   MALE1   -0.709573   -1.053857   -0.456036   0.429879    Adelie  Torgersen   FEMALE2   -0.790732   -0.393550   -0.130809   -0.839620   Adelie  Torgersen   FEMALE3   -1.718663   -0.542888   -0.073623   -0.458571   Adelie  Torgersen   FEMALE4   -1.772295   -0.763548   0.736248    -0.014204   Adelie  Torgersen   MALE |

Let us check how correlated the the first pair of canonical covariates are. We use NumPy’s corrcoef() function to compute the correlation. And we can see that the first of pair of canonical covariates are highly correlated.

|  |  |
| --- | --- |
| 1234 | import numpy as npnp.corrcoef(X\_c[:, 0], Y\_c[:, 0])array([[1.        , 0.78763151],       [0.78763151, 1.        ]]) |

We can also compute the correlation between the second pair of covariates and we see that the correlation is not that high.

|  |  |
| --- | --- |
| 1234 | np.corrcoef(X\_c[:, 1], Y\_c[:, 1]) array([[1.        , 0.08638695],       [0.08638695, 1.        ]]) |

To further understand the relationship between the pairs of canonical covariates, let us make scatter plot with the first pair of canonical covariates.

|  |  |
| --- | --- |
| 1234567 | sns.set\_context("talk", font\_scale=1.2)plt.figure(figsize=(10,8))sns.scatterplot(x="CCX\_1",                y="CCY\_1",                 data=cc\_res)plt.title('Comp. 1, corr = %.2f' %         np.corrcoef(X\_c[:, 0], Y\_c[:, 0])[0, 1]) |

We can see the high correlation between the first pair of canonical covariates.

Scatter plot of First Pair of Canonical Covariates

Interpreting Canonical Covariates with Heatmap

In this toy example, we know already that the species variable in the dataset is a latent variable or hidden variable. Let us see how the latent variable is correlated with the first pair of canonical covariates. We first make a boxplot between the latent variable and each of the first pair of canonical covariates.

|  |  |
| --- | --- |
| 1234567 | plt.figure(figsize=(10,8))sns.boxplot(x="Species",                y="CCX\_1",                data=cc\_res)sns.stripplot(x="Species",                y="CCX\_1",                  data=cc\_res) |

Boxplot of Canonical Correlate from X and Latent Variable

|  |  |
| --- | --- |
| 1234567 | plt.figure(figsize=(10,8))sns.boxplot(x="Species",                y="CCY\_1",                  data=cc\_res)sns.stripplot(x="Species",                y="CCY\_1",                  data=cc\_res) |

Boxplot of Canonical Correlate from Y and Latent Variable
By coloring the scatterplots between the first pair of canonical covariates with the species variable and see how the canonical covariates have capture the underlying variable behind our datasets.

|  |  |
| --- | --- |
| 123456 | plt.figure(figsize=(10,8))sns.scatterplot(x="CCX\_1",                y="CCY\_1",                 hue="Species", data=cc\_res)plt.title('First Pair of Canonical Covariate, corr = %.2f' %         np.corrcoef(X\_c[:, 0], Y\_c[:, 0])[0, 1]) |

Scatter plot of First Pair of Canonical Covariates colored by Latent Variable

From the correlations between the first pair of canonical covariates and species variable in the data set, we can see that our Canonical Correlation Analysis has captured the shared variation between the two datasets. In this example the shared variation or the latent variable behind the first pair of canonical covariates is the species variable.

By carefully looking at the association between CCs and a variable here the latent variable, we can definitely understand the results from our CCA results. Let us take a step further and make a heatmap of correlations between canonical covariates from each of the dataset against our input dataset including the meta data associated with it,

Let us repeat ourselves by creating dataframe with original data and the canonical covariates from the first dataset X. To compute correlation, we convert the character variables into categorical variables and convert them to 0/1/2 codes.

|  |  |
| --- | --- |
| 1234567 | ccX\_df = pd.DataFrame({"CCX\_1":X\_c[:, 0],                       "CCX\_2":X\_c[:, 1],                       "Species":df.species.astype('category').cat.codes,                      "Island":df.island.astype('category').cat.codes,                      "sex":df.sex.astype('category').cat.codes,                       "bill\_length":X\_mc.bill\_length\_mm,                      "bill\_depth":X\_mc.bill\_depth\_mm}) |

With Pandas’ corr() function we can compute correlation of all variables in the dataframe.

|  |  |
| --- | --- |
| 12 | corr\_X\_df= ccX\_df.corr(method='pearson') corr\_X\_df.head() |

Let us make a [heatmap with lower triangular correlation matrix](https://cmdlinetips.com/2020/02/lower-triangle-correlation-heatmap-python/). To do that we subset the original correlational matrix with Numpy’s tril() function.

|  |  |
| --- | --- |
| 12 | plt.figure(figsize=(10,8))X\_df\_lt = corr\_X\_df.where(np.tril(np.ones(corr\_X\_df.shape)).astype(np.bool)) |

We can make make lower triangular correlation heatmap using Seaborn’s heatmap function.

|  |  |
| --- | --- |
| 12345 | sns.heatmap(X\_df\_lt,cmap="coolwarm",annot=True,fmt='.1g')plt.tight\_layout()plt.savefig("Heatmap\_Canonical\_Correlates\_from\_X\_and\_data.jpg",                    format='jpeg',                    dpi=100) |

The heatmap showing correlations of the canonical covariates from the dataset X reveals a lot of interesting details. We can see that as expected there is nor correlation between the first and second canonical covariates from the dataset X. Note that the covariate is highly correlated to the two variables in the dataset X, positively with bill length and negatively with depth.

As we saw before, the first canonical co-variate is highly correlated with Species variable, the latent or hidden variable in this example. The first canonical covariate from X is also correlated with the other hidden variable, island, associated with the dataset. We can also see that the first canonical covariate is not correlated to sex variable. However, the second canonical covariate is correlated with sex variable moderately.

Correlation Heatmap of Canonical Correlates from Data X and dataset

Let us make a similar lower triangular correlation heatmap with the canonical covariates from Y dataset Y and the dataset Y itself including the meta data associated with it.

|  |  |
| --- | --- |
| 12345678910111213141516171819202122 | # second pair of canonical covariates with the dataset ccY\_df = pd.DataFrame({"CCY\_1":Y\_c[:, 0],                       "CCY\_2":Y\_c[:, 1],                       "Species":df.species.astype('category').cat.codes,                      "Island":df.island.astype('category').cat.codes,                      "sex":df.sex.astype('category').cat.codes,                       "flipper\_length":Y\_mc.flipper\_length\_mm,                      "body\_mass":Y\_mc.body\_mass\_g}) # compute correlation with Pandas corr()corr\_Y\_df= ccY\_df.corr(method='pearson')  # Get lower triangular correlation matrixY\_df\_lt = corr\_Y\_df.where(np.tril(np.ones(corr\_Y\_df.shape)).astype(np.bool)) # make a lower triangular correlation heatmap with Seabornplt.figure(figsize=(10,8))sns.heatmap(Y\_df\_lt,cmap="coolwarm",annot=True,fmt='.1g')plt.tight\_layout()plt.savefig("Heatmap\_Canonical\_Correlates\_from\_Y\_and\_data.jpg",                    format='jpeg',                    dpi=100) |

We see a similar pattern as we saw with the correlation map of canonical covariates from dataset X. One of the highlights is as we saw before the first canonical covariate from dataset Y is highly correlated with Species variable. And the second canonical covariate from dataset is Y is correlated with sex variable, suggesting that we can capture the effect of two distinct hidden variables using canonical correlation analysis.

Correlation Heatmap of Canonical Correlates from Data Y and dataset

Second pair of Canonical covariates Reveal a Second Latent Variable

The two heatmaps showing the correlations of canonical covariates and datasets show that sex is another variable that has an effect on both the datasets and CCA could capture that. To see the effect of sex, we can make a scatter plot with the second pair of canonical covariates and color it by sex variable.

|  |  |
| --- | --- |
| 123456 | plt.figure(figsize=(10,8))sns.scatterplot(x="CCX\_2",                y="CCY\_2",                 hue="sex", data=cc\_res)plt.title('Second Pair of Canonical Covariate, corr = %.2f' %         np.corrcoef(X\_c[:, 1], Y\_c[:, 1])[0, 1]) |

Scatter plot of Second pair of Canonical Covariates

Summary of CCA Example in Python

Summarising what we saw so far, Canonical Correlation Analysis is a great toolkit to understand tow high dimensional datasets. As a toy example with Penguins data, the post showed how to perform CCA with Scikit-learn in Python. We also saw how to to interpret and understand pairs of canonical covariates obtained by looking at the two datasets at the same time.

In this post we did not get to the math or algorithm behind CCA, it will be fun to look at that in a future post. Also, Penguin dataset was perfect to illustrate CCA and it will be lot more fun to apply CCA to a more complex/realistic high dimensional datasets. Definitely great ideas for another few posts.