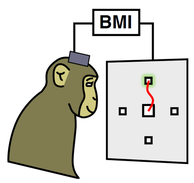


PCA & NMF on Neuronal Activity Recordings in the Brain

Yiyan Li

Imperial College London

Data Dimensionality Reduction

In many fields, scientists deal with high-dimensional data, such as DNA microarrays, molecules etc. With high-dimensional data, the number of features can exceed the number of observations, which makes calculations extremely difficult.

To develop more effective models that allow data analysis become computationally feasible and interpretable, we need to perform dimensionality reduction. **Data Dimensionality Reduction** *is about obtaining low-dimensional representations of the data.*

There are several techniques to perform dimensionality reduction. The following text will illustrate **PCA** and **NMF**.

Problem Setting

Data $\mathbf{X}_{N\times p}$ (rows $\{\mathbf{x}^{(n)}\}_{n=1}^{N}$ are data points)

Covariance matrix $C_{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}^{(n)} - \mu)(\mathbf{x}^{(n)} - \mu)^T$: describes information in the data

k: number of principal components (PCs), parameter that controls how well the data are approximated

Principal Component Analysis

Principal Component Analysis (PCA) is based on a projection onto a subspace of lower dimensions designed to retain maximal information of the original data.

PCA seeks directions of projection $\{\phi_j\}_{j=1}^{m}$ that capture the most variance.

Non-negative Matrix Factorisation

Non-negative Matrix Factorisation (NMF) approximates $\mathbf{X}_{N\times p}$ by the product of $\mathbf{W}_{N\times k}$ (non-negative coefficients) and $\mathbf{H}_{k\times p}$ (non-negative equivalent of PCs).

We find the best $\mathbf{W}$ and $\mathbf{H}$ by *Lee and Seung's Multiplicative Update Rule* [1], an iterative algorithm.

Monkey BMI Tensor Dataset

The Monkey Brain Machine Interface (BMI) tensor data [2] has been curated from a series of experiments, where a monkey moves a cursor to one of four targets (at 0, 90, 180 and -90 degrees).

The tensor is formatted as **43 neurons** $\times$ **200 time steps** $\times$ **88 trials**. To study neuron assemblies with correlated patterns of activity, we analyse the data as the following two tasks:

1. Fix a time step ($t = 50$) to get a matrix $\mathbf{X}_{88\times 43}$,
2. Average over the same-angle trials to get a matrix $\mathbf{X}_{200\times 43}$.

Then we implement PCA and NMF in Python to reduce their dimensions. In task 1, We visualise the data using projection onto 1st principal component (PC1) and 2nd principal component (PC2). In task 2, we use the coefficient of PC1 (a_1) to represent the activity of the major neuron at different time steps.

Python Plots

PCA results:

Fig 1.1: 2D Representation using PCA ($t = 50$)

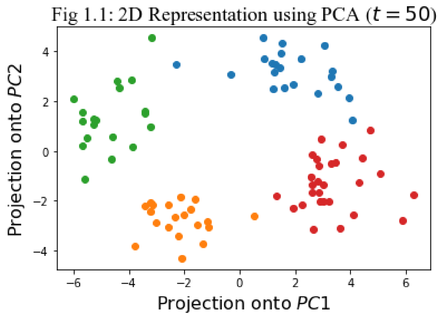


Fig 2.1: Fraction of Residual Variance

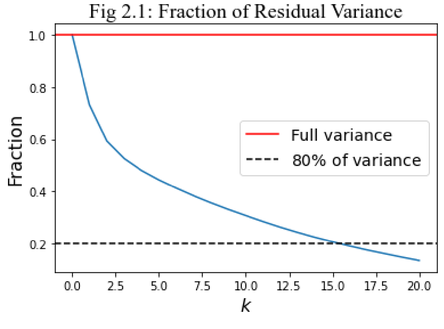
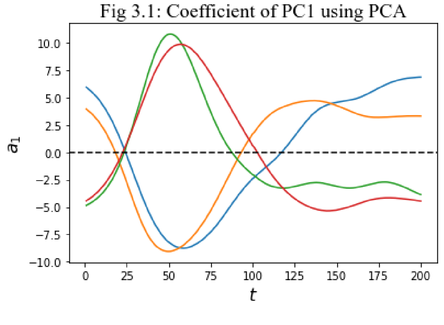


Fig 3.1: Coefficient of PC1 using PCA



NMF results:

Fig 1.2: 2D Rpresentation using NMF ($t = 50$)

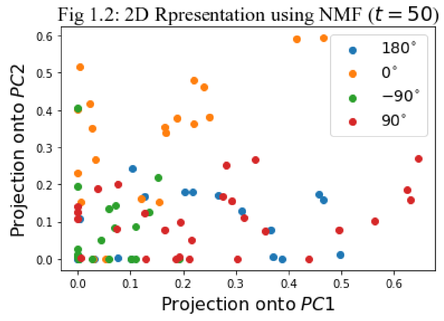


Fig 2.2: MSE of NMF ($t = 50$)

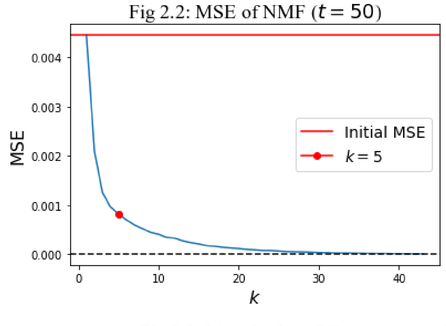
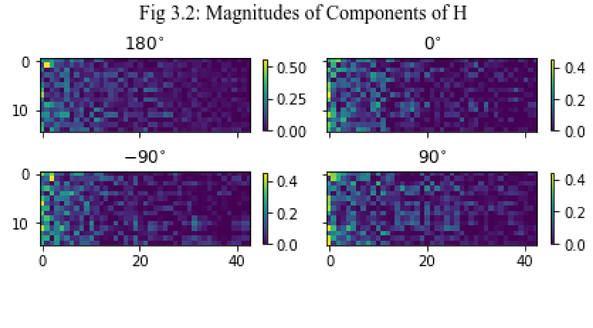


Fig 3.2: Magnitudes of Components of H



Compare PCA and NMF Results

Low dimensional representation:
In fig 1.1, we can see neurons with same angle group together in low dimensions, while there is no obvious clustering shown in NMF result (fig 1.2). This is because negative values compensate for positive values during the iterate process of NMF, resulting in losing local properties.

In fig 3.1, we observe a peak of neuronal activities at time step $t = 50$, therefore we have chosen $t = 50$ in task 1.

The choice of k:
In PCA, we calculate the fraction of residual variance (fig 2.1) by using the covariance matrix $C_{\mathbf{x}}$. Setting $k = 15$ allows 80% of variance to be captured. However, NMF is more complex, since all components are trained at same time. We use *mean squared error (MSE)* (fig 2.2) instead in NMF. If we set a threshold of 20% of initial MSE, then $k = 5$.

Note that the MSE of NMF is significantly smaller than the MSE of PCA, since the iterative algorithm ensures NMF is close to the mean. As shown in fig 3.2, magnitudes of components of $\mathbf{H}$ (NMF equivalent of PCs) are all close to zero.

Conclusions

PCA is useful for feature discovery since similar data cluster together. However, PCA has limitations as it relies on linear transforms.

NMF tends to describe global properties, the iterative algorithm causes loss on local features. The constraint of non-negativity can help the interpretability of the basis vectors and the low dimensional representation.

References

[1] Lee DD, Seung HS. Learning the parts of objects by non-negative matrix factorization. *Nature* 1999; 401: 788-791.

[2] T. G. Kolda. Monkey BMI Tensor Dataset. `https://gitlab.com/tensors/tensor_data_monkey_bmi`, 2021.

[3] Deisenroth MP, Faisal AA, Cheng SO. Dimensionality Reduction with Principal Component Analysis. In: *Mathematics for Machine Learning*. Cambridge University Press; 2020. p.317-347.