Independent Component Analysis

ECE 532

Department of Computer Science

University of Wisconsin-Madison

Project Team 15: Haonan SHEN, Zeyu TAN, Xinyu ZENG

April, 2020

Abstract

In the machine learning area, it is important for researchers to find the pattern of data points. When a set of data points is provided, researchers are commonly asked to separate component(s) that best represent the data set. Among those various component extraction techniques, Principal Component Analysis(PCA) is one of the most frequently used approaches. It finds the principal component, i.e. the first column of U(matrix U is from the singular value decomposition of A) and the correlated stretching factor σ1. This method works well for single pattern models, and it is easy to compute, as it compresses the data set onto a lower dimension feature. However, in real world circumstances, there are many cases that are not suitable to be represented by a single principal component; in those cases, a new method is needed so that multiple components could be separated from the data set. Independent component analysis (ICA) is one of the solutions for those cases. Invented in the early 1980s, first used to solve a neurophysiological area issue, and further expanded its usage during the 1990s, the ICA isolates multiple components in the data set by decorrelating the components (in many cases, components are individual signals). In the following tutorial, the entire procedure of applying the ICA will be provided step by step from the beginning of how to preprocess data to the end of finding components through FastICA algorithm. Students are expected to review the PCA learnt in previous lectures, understand the limitations of PCA, determine when to apply PCA or ICA, master how to find components through FastICA algorithm, and know how to implement these algorithms in Python after completing this tutorial with the assistance from classmates and instructors.

Learning Objectives:

1. Students will learn the mechanism of Independent Component Analysis.
2. Students will understand the difference between Independent Component Analysis and other analysis approaches such as Principal Component Analysis.
3. Students will be able to process signals using the FastICA algorithm.
4. Students will have a concrete understanding about Singular Value Decomposition as they explore PCA and ICA.
5. Students will be able to appropriately apply Independent Component Analysis in real life data analysis cases.

Background

Problem: Component extraction

Introduction

When analyzing a set of data points, it is essential to find the vector components that can represent the data points, because vector components can reflect the major pattern of the data points.

Principal Component Analysis(PCA)

What is PCA

One of the most frequently used unsupervised machine learning methods to extract principal components is Principal Component Analysis(PCA). By getting the singular value decomposition of the data matrix A, PCA can find the principal component, which is the first column of matrix U, and the stretching factor σ1. Similarly, the second principal component, which is orthogonal to the first principal component, is the second column of U and the stretching is σ2. The figure 1.1 below visually illustrates how PCA works.

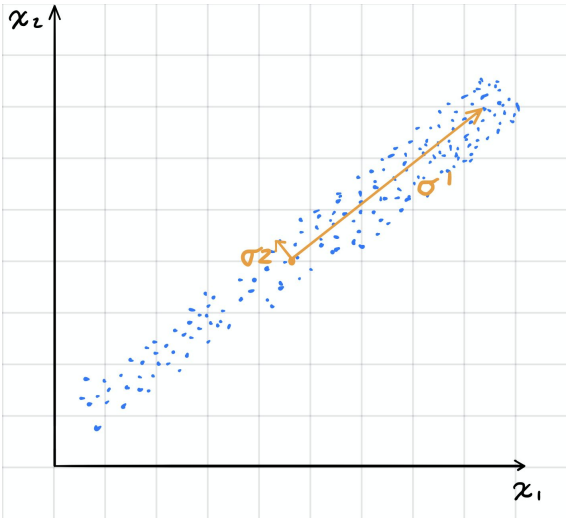


Figure1.1 PCA

Limitation of PCA

However, if there are more than one pattern hidden within data points, which is highly possible in real life cases, can PCA be applied to extract principal components in those circumstances? The answer is no (Technically, a component still can be extracted, but it is not

‘principal’ at all). The following example will show why PCA is not reliable in extracting a dataset that contains more than one major pattern.

Assume the figure 2.1 below represents a set of data points collected. There are clearly two main patterns shown by the data points. So the expectation for the components is very likely to be the two components shown in figure 2.2. However, if PCA is directly used to exact the principal component, the result will look similar to the one shown in figure 2.3. As shown in the figure, the principal component computed can reflect neither of the two patterns. That is caused by the fact that PCA is taking and using the SVD of all the data points. In other words, the principal component found by PCA is defective because PCA is using one component vector to represent two patterns. Fortunately, an alternative method, which is known as individual component analysis (ICA), can analyze data points distributed like this and separate components with better accuracy.

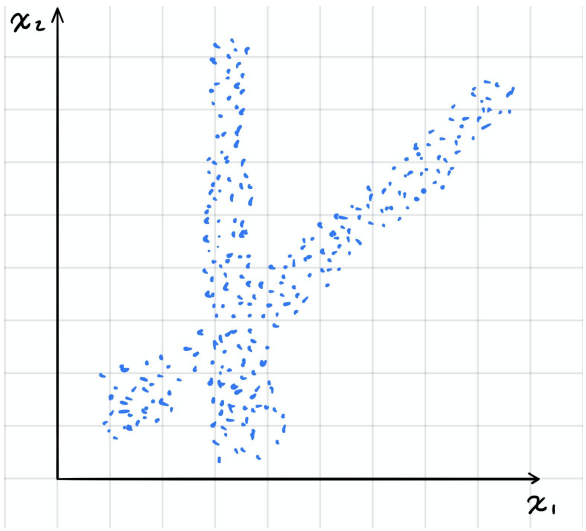


Figure 2.1

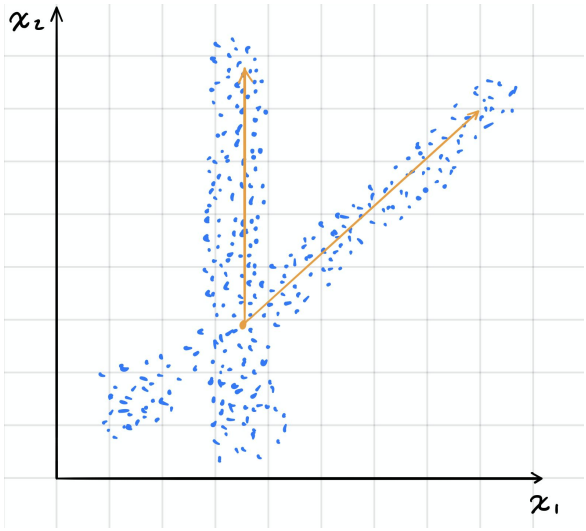


Figure 2.2

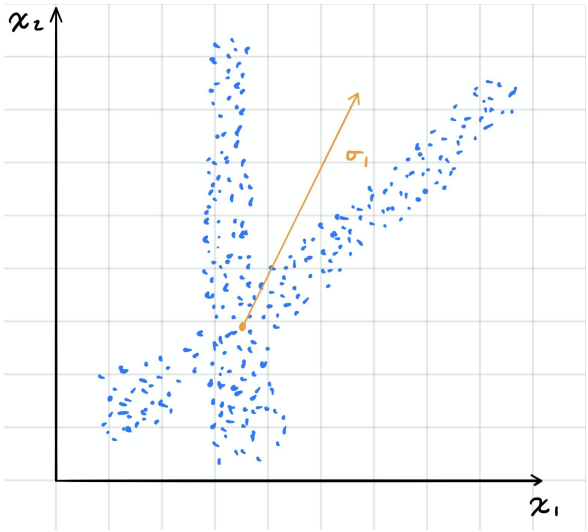


Figure 2.3

Individual Component Analysis(ICA)

History of ICA

The ICA technique, though not yet this name, was first introduced by J. Herault, C. Jutten, and B. Ans in the early 1980s to solve a problem in the neurophysiological area. During the 1980s, the ICA was mostly known among French researchers, with limited global influence. ICA was popularized in the mid-1990s, after Tony Bell and Terry Sejnowski published the paper about a fast and efficient ICA algorithm based on informax. Among many approaches that implement ICA, one of the most commonly used is FastICA, invented by Aapo Hyvärinen in the late 1990s.[2] Since then, ICA has become a widely applied technique to separate a multivariate signal into additive subcomponents in various areas, such as face recognition, cell phone communication, and etc.

Introduction to ICA

One of the most famous examples for ICA might be the “cocktail party problem”. Imagine that we are in a very crowded cocktail party shown in Figure 3.1, and three microphones, located at different positions of the room, are collecting sound signals from people. Every microphone collects mixed sound because many people are talking at the same time. If a data scientist is asked to extract an individual’s sound from the mixed sound provided by three microphones, ICA,an unsupervised machine learning method, can play an important role in solving such kinds of problems.

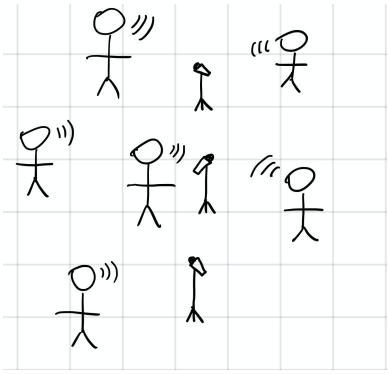


Figure 3.1

Insight from cocktail party problem

The cocktail party problem mentioned above can give insight to the application of ICA. Assume now there are only three people in the cocktail party and there are three microphones in the room. Three people can be considered as three original source of sound: $s_1(t), s_2(t), s_3(t)$, and the three collected mixed sound: $x_1(t), x_2(t), x_3(t)$. It is also assumed that the observed signal follows the model: $X = AS$. Then the linear combination can be represented as shown in Figure 3.2. In these linear equations, $x_1(t), x_2(t), x_3(t)$ are known because they are the mixed sound collected by three microphones. $s_1(t), s_2(t), s_3(t)$ are the unknown variables are clear and individual sources of sound we are looking for. A is an unknown matrix.

cocktail-party problem

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t) + a_{13}s_3(t)$$
$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t) + a_{23}s_3(t)$$
$$x_3(t) = a_{31}s_1(t) + a_{32}s_2(t) + a_{33}s_3(t)$$

Figure 3.2

Generally, what we expect ICA to do is to approximate unknown mixed matrix A and different original source matrix(column vector in some cases) S, given only the measured data matrix(column vector in some cases) X.

Step 0. Before using ICA, check prerequisites.

As the introduction and the insight given by “cocktail party problem” stated, ICA is a powerful tool to approximate the original data based on measured data. However, there are two important prerequisites for applying ICA:

1. Individual components must be ‘independent’.
2. Independent components cannot be Gaussian distributed.

ICA was not so popular at an earlier time partially because it can only be used to find individual components that are not Gaussian distributed. To understand why the individual components cannot be Gaussian distributed, it is important to visualize the multivariate distribution of two independent Gaussian variables x_1, x_2, which is shown in Figure 4.1. The figure shows that the density is symmetric so no information about direction of mixed matrix A can be obtained. So matrix A is not likely to be approximated.

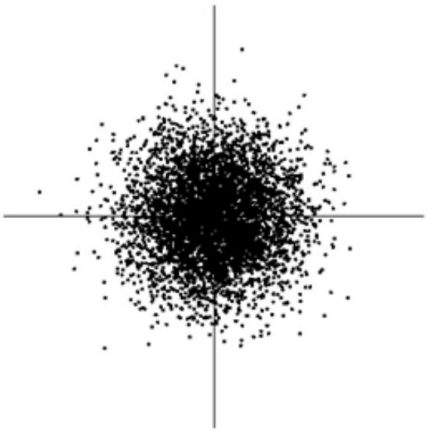


Figure 4.1[3]

$$p(y_1, y_2) = p_1(y_1)p_2(y_2)$$

Figure 4.2

The word ‘independent’ in the first prerequisite mentioned above means the information provided by one variable can not be affected by the information provided by the other variable. This prerequisite can also be represented by the formula shown in Figure 4.2.

Step 1. Preprocessing the data

After checking the feasibility of ICA through checking if the two assumptions are met (i.e. independence and non-Gaussian distribution of individual components), the ICA can be used to help find individual components. Before using the algorithm, the collected data need to be preprocessed. The preprocessing includes two two parts:

1. Centering
2. Whitening

Centering:

When we utilize the PCA algorithm, it is necessary to center the data first. This centering process remains the same when we utilize the ICA algorithm. By subtracting X with mean value of X, we make X a zero-mean variable.

Whitening:

It is another important step if preprocessing the data. The purpose of whitening is to linearly transform x to obtain a new vector z which has components that are uncorrelated and has unitary variance (variance = 1). In other words, the covariance matrix of z equals the identity matrix.

The actual way to do whitening is by eigendecomposition. The final result is $x_{whiten} = ED^{-1/2}E^T x$ where E is the eigendecomposition of the covariance matrix of X and D is the diagonal matrix of eigenvalues.

Step 2. Applying FastICA algorithm.

To find the individual components, there are many different approaches available. FastICA is one of the powerful algorithms that can approximate the individual component.

The final purpose of FastICA is to find a unit vector W, which represents a direction such that the projection $W^T X$ maximizes nongaussianity. After centering and whitening the data points, the steps of FastICA algorithm is shown below[4]:

1. Choose an initial (e.g. random) weight vector **w**.
2. Let $\mathbf{w}^+ = E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - E\{g'(\mathbf{w}^T\mathbf{x})\}\mathbf{w}$
3. Let $\mathbf{w} = \mathbf{w}^+ / \|\mathbf{w}^+\|$
4. If not converged, go back to 2.

E in step 2 gets the mean value and norm in step 3 is the Frobenius Norm.[4]

g is some non quadratic function, and it is important not to choose the function g that grows fast. Two suggested functions are: [4]

$$g_1(u) = \tanh(a_1 u),$$
$$g_2(u) = u\exp(-u^2/2)$$

The algorithm converges when W^+ and W are the same or inverse. That means the directional unit vector W are pointing to the same directing(if W and W^+ point to the opposite direction, the FastICA algorithm still converges).[3]

It is also worth mentioning that the order of the independent component may not be the same.

Another approach to find the individual component

Except for the FastICA algorithm, which is a powerful tool for ICA, Singular Value Decomposition(SVD) can also find the individual component analysis. SVD, which is taught in chapter 3 and chapter 4, is a very useful tool in machine learning and it is useful both in ICA and PCA.

As we mentioned before, we model the problem with $X = AS$ and this can be represented by the linear combination shown in Figure 3.2. We want to approximate unknown matrix A and different original source matrix(column vector in some cases) S, given only the measured data matrix(column vector in some cases) X. Normally, we will get S by finding XA^{-1}. However, because A is an unknown mixed matrix, we have to approximate A first.

Before approximating A, it is nice to review how SVD works:

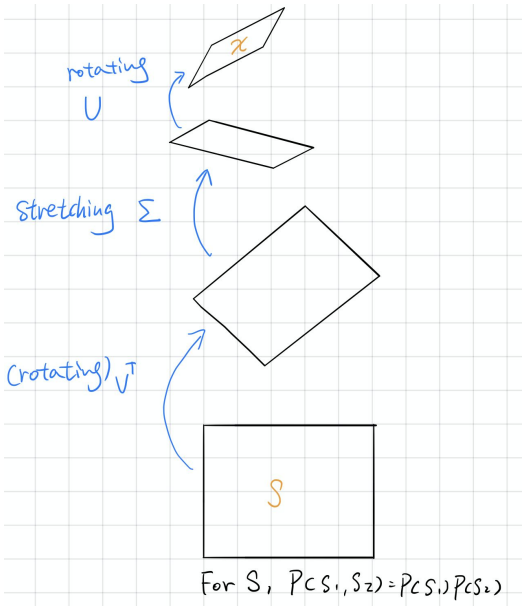
For any known matrix A, the matrix has the SVD shown below:

$$A = U\Sigma V^T$$

When we are considering AS, we are actually computing:

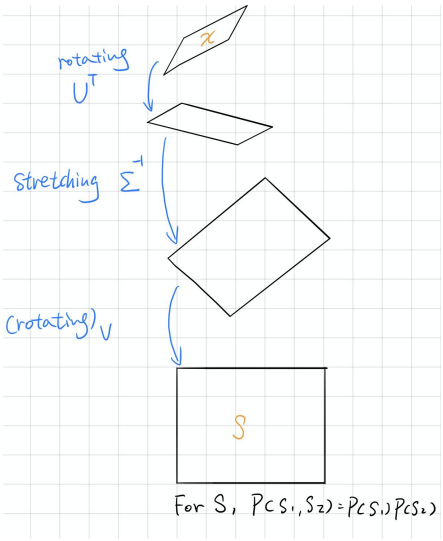
$$AS = U\Sigma V^T S$$

Therefore, $X = AS$ can also be visualized as the figure shown below:



A actually rotate S, stretch S and rotate S again. After these operation, we get X.

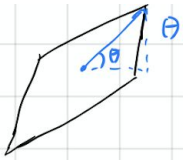
Now, let us consider this inversely, if we are given X and we are asked to find A and S, what should we do?



To sum up, it takes three steps to approximate A and S:

Step 1. Get the variance function of angle and maximize or minimize that function, U^T can be obtained

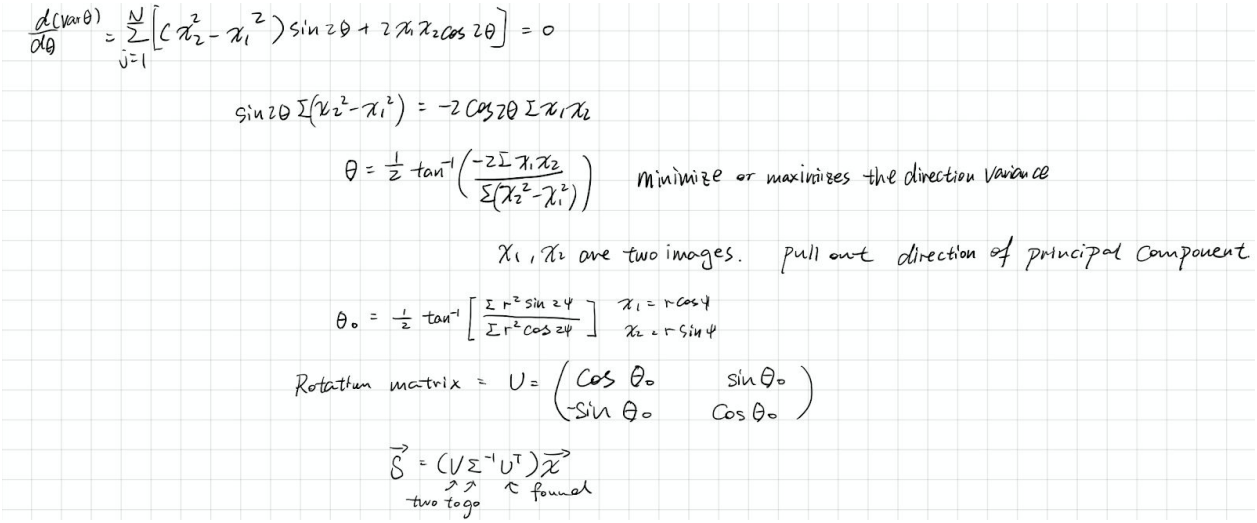
get the variance function of θ



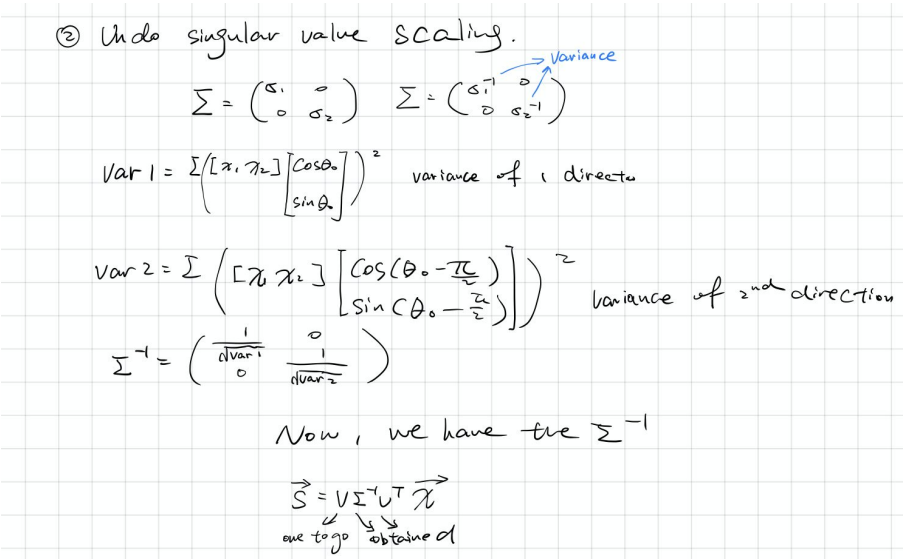
$$Var(\theta) = \sum_{j=1}^{N} \left\{ [x_1(j) \; x_2(j)] \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \right\}^2$$

$$= \sum_{j=1}^{N} (x_1 \cos\theta + x_2 \sin\theta)^2$$

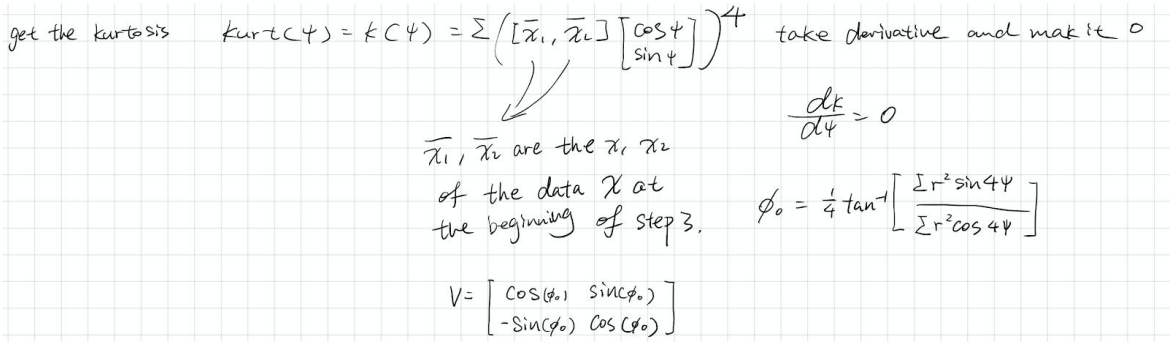
$$= \sum_{j=1}^{N} (x_1^2 \cos^2\theta + 2 x_1 x_2 \sin\theta \cos\theta + x_2^2 \sin^2\theta)$$



Step 2. Undo the singular value scaling. Σ^{-1} can be obtained.



Step3. Approximate S with sperable(independant) elements maximizing or minimizing the kurtosis function. V can be obtained.



After getting Σ^{-1}, V, and U^T, individual components S can be obtained by computing:

$$S = VU^T\Sigma^{-1}X$$

Ethical Issues

One potential ethical issue may be to monitor some specific person's voice in a noisy environment because by using ICA we can differentiate different people's voices just like the cocktail party problem. This indicates that this algorithm can be used to hurt privacy which may need further regulation.

Conclusion

PCA and ICA are both unsupervised methods to find the component. PCA finds the principal component, the component that reflects the main pattern. ICA finds and extracts the individual component from the overall data. Many different approaches can be used to implement ICA. For example, FastICA, an iterative method, can find and using SVD can both find the individual component. There are also other available approaches waiting to be discussed.

Warm-up Questions

1. T/F ICA is widely used to find the basis representing the variance of a set of data.
2. Bucky Badger has two images, and he decides to mix them up with different proportions.

Here are the two mixtures, and Bucky uses a technique to separate these two images.

The following two images are the results Bucky gets after separation.

(Images are adapted from Quora author Luis Argerich)[1]

Which of the following techniques is most likely used by Bucky?

A. PCA
B. Neural Network
C. ICA
D. K-means Clustering

3. Both used to find components of a set of data, PCA is _____, and ICA is _____.
    A. supervised learning; supervised learning
    B. supervised learning; unsupervised learning
    C. unsupervised learning; supervised learning
    D. unsupervised learning; unsupervised learning
4. Select all that apply. Which of the following assumptions are needed before using the ICA algorithm.
    a. The components have to be dependent on each other
    b. The components have to be independent
    c. Data could be distributed in any pattern
    d. Data cannot be Gaussian distributed
    e. Signal could only be sound collected by microphones

5. T/F The purpose of using whitening to process data before applying ICA is to ensure the unitary variance and uncorrelatedness of components.

Main Activity: Independent Component Analysis

6. First you are going to apply ICA on the mixture of two well-defined signals----A periodic square-wave waveform and a periodic sawtooth or triangle waveform. For simplicity, we will use the similar mixed sound model in the cocktail party problem. However instead of $X = AS$ we are going to use $X = SA$ to represent the linear transformation, where X is the mixed signal, A is the linear transformation matrix, and S is the matrix for original signals. You can find the skeleton code at *activity.ipynb.* [5][6]

a) First try to use PCA to get the principal components of real signals. Plot the components you get as predicted signals, do they correctly reflect the original signals?

b) Finish the code to get the independent component matrix by using FastICA provided by sklearn to have a first taste of the power of ICA. Hint: look at the documentation of FastICA here.

c) What is your observation from the predicted signals? Any difference between your predicted signal, the original signals and real signals?

7. Using the same data provided in question 6, now instead of using the library provided by python, you are going to implement the FastICA algorithm by yourself.
   a) We will first start by preprocessing the data. Complete the code in centering and whitening. Hint: loot at the background again to see the mathematical formula for whitening. Centering is the same as subtracting the mean in PCA.
   b) Next, fill out the code in `update_w()`. Refer to the background to see the FastICA algorithm.
   c) Compare the result of our FastICA with the FastICA provided by sklearn. Does our algorithm seem good enough?
   d) Change the parameter $alpha$ in `g(x)` and `g_prime(x)`, what can you observe in the predicted signals?
   e) Change the iteration times and tolerance, what can you observe in the predicted signals?

References

[1] Argerich, L. (2015, December 17). What is the difference between PCA and ICA? Retrieved April 28, 2020, from https://www.quora.com/What-is-the-difference-between-PCA-and-ICA

[2] Hyvarinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, *10*(3), 626–634. doi: 10.1109/72.761722

[3] Hyvärinen, A., & Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural Networks*, *13*(4-5), 411–430. doi: 10.1016/s0893-6080(00)00026-5

[4] Hyvärinen Aapo, Karhunen, J., & Oja, E. (2001). *Independent component analysis*. Chichester: Wiley-Interscience.

[5] Pedregosa *et al.*, (2011). Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830. https://scikit-learn.org/stable/auto_examples/decomposition/plot_ica_blind_source_separation.html

[6] Marklin, C. (2019). *Independent Component Analysis (ICA) In Python.*

https://towardsdatascience.com/independent-component-analysis-ica-in-python-a0ef0db0955e

Acknowledgement

We would like to thank all the teaching assistants and Prof. Malloy for their instructions and efforts to help us study at this challenging moment.

Appendix: Warm-up and main activity solution

1. False. In fact, PCA is used to find the basis representing the variance of a set of data.
2. C. ICA is most likely used to separate components and get results like this.
3. D. Both PCA and ICA are unsupervised machine learning algorithms, as data set input for both algorithms are not labeled.
4. B & D. ICA needs two assumptions to ensure components can be isolated.
5. True. Whitening is used for purposes mentioned above, and is important for ICA implementation.

6.

a).

```
U, s, VT = np.linalg.svd(X, full_matrices=False)
plot_signal_matrix(U, title="predicted signals")
```

b).

```
ica = FastICA(n_components=2)
X_hat = ica.fit_transform(X)
```

c).

Real signal is the linear transformation of the mixture of original signals.
Predicted signal finds the wavelength well, but gets the wrong amplitude.

7.

a).

```
def centering(x):
    return x - x.mean(axis=1, keepdims=True)
```

```
def whitening(x):
    d, E = np.linalg.eig(np.cov(x))
    D = np.diag(d)
    D_inv_sqrt = np.sqrt(np.linalg.inv(D))

    x_whiten = E@D_inv_sqrt@E.T@x
    return x_whiten
```

b).

```
def update_w(w, X):
    w_new = np.mean((X * g(w.T@X)), 1) - np.mean(g_prime(w.T@X), 0) *
w
    w_new /= np.linalg.norm(w_new)

    return w_new
```

c).
It should be almost the same as FastICA by sklearn if the algorithm is implemented correctly.

d).
The predicted signal becomes significantly different from the original signal when alpha gets bigger(~50).

e).
When the iteration number is small or tolerance is big, the final predicted signal is more different than the original signal. Also the result is inconsistent because of the numerical computation of python.