Introduction

Assuming some data $s \in \mathbb{R}^n$ are generated from n independent sources. These are the n independent hidden variables. These can be, for example, n independent speakers talking in a party. The observed signal can be written as $x = As$, where A is an unknown square matrix called the mixing matrix. An example for x is the signal recorded by n microphones. The signal recorded by each microphone can be regarded as a linear combination of the voices from n independent speakers.

Repeated observations gives a dataset $\{x^{(i)}; i = 1, \ldots, m\}$. Our goal is to recover the sources $s^{(i)}$ that had generated our data ($x^{(i)} = As^{(i)}$). Let $W = A^{-1}$ be the unmixing matrix, then we are to find W so that given the microphone recordings $x^{(i)}$, we can recover the sources by computing $s^{(i)} = Wx^{(i)}$.

In the equation $x^{(i)} = As^{(i)}$, we only know the observed signal x. Obviously there is not enough information to solve for s. However, in ICA we will use the condition of **independent** source signals to extract s from x. Different ways of constraining this independence gives different algorithms for ICA. A few of them will be explained briefly below.

Ambiguity of ICA

- The permutation of the original sources s^i is ambiguous to ICA.
- ICA cannot recover the correct scaling of the source, neither the sign.
- Normally these ambiguities do not matter in typical applications.

Non-Gaussianity requirement in ICA

http://cs229.stanford.edu/notes/cs229-notes11.pdf (http://cs229.stanford.edu/notes/cs229-notes11.pdf)

Consider an example in which $n = 2$, and $s \sim N(0, I)$. Here, I is the 2×2 identity matrix. The density for the source is rotationally symmetric. Suppose we observe some $x = As$, where A is our mixing matrix. The distribution of x will also be Gaussian, with zero mean and covariance $E[xx^T] = E[Ass^TA^T] = AA^T$. Now, let R be an arbitrary orthogonal matrix, so that $RR^T = R^TR = I$, and let $A' = AR$. Then if the data had been mixed according to A' instead of A, we would have instead observed $x' = A's$. The distribution of x' is also Gaussian, with zero mean and covariance $E[x'(x')^T] = E[A'ss^T(A')^T] = E[ARss^T(AR)^T] = ARR^TA^T = AA^T$. Hence, whether the mixing matrix is A or A', we would observe data from a $N(0, AA^T)$ distribution. Thus, there is no way to tell if the sources were mixed using A and A'. So, there is an arbitrary rotational component in the mixing matrix that cannot be determined from the data, and we cannot recover the original sources.

Comments:

- How about if two sources are both Gaussian but with different variance? A linear combination of of Gaussian random variables is always Gaussian random variable. However, a combination of square of Gaussian random variables can be, in special cases, χ^2 distribution. See details in statistics notes.
- The key here is for Gaussian data, in the transformed data (after As) we cannot identify an axis where either variance is maximum, or non-Gaussianity is maximum. Therefore, we cannot identify the 'angle' we 'rotate' the data, and hence cannot rotate it back with the same angle. See details later about using SVD and 4th moment (kurtosis) maximization to find rotated angles.

Numerical methods

ICA with maximum likelihood estimation (MLE)

ICA with sigmoid function http://cs229.stanford.edu/notes/cs229-notes11.pdf (http://cs229.stanford.edu/notes/cs229-notes11.pdf)

Note this is very different from the matrix form of linear regression, where we have only one vector of parameter. Here we have multiple columns of parameters stored in W. The derivation of ICA takes matrix W as the parameters and uses the standard MLE to find the W updating formula. This is same as deriving other formula for logistic or linear regressions. Below are the key points,

- In linear regression we have $y = \theta^T x$. The we calculate the log likelihood using $p(y|\theta)$ and maximize it w.r.t. theta. The we obtain the update equation for θ.
- In ICA here, it is exactly same process except the random variable is x, and the parameter is not just one column but a whole matrix. Here we use $x = W * s$ where parameter is a matrix, different from the $y = \theta^T x$. However, the idea is exactly same.
- The key point employed in the derivation is assuming the independence of signals. This method is different from the ways of maximizing Non-Gaussianity as shown below.
- In Andrew's note of ICA (page 6), it remarks when we have large amount of data, then even the $x_i, i = 1, 2, \ldots m$are correlated (e.g. for speech time series), we can still assume they are independent and use the multiplication rule in MLE. It also says that using special form of SGD can help with convergence if the data are correlated.
- The data should be centered. See cs229 notes for the reason. In other cases such as the data for linear/logistic regression, the samples (different x_i) are independently sampled and thus independent condition is satisfied as long the sample size is much smaller than the population size. From this example, we know that even there is correlation, we can have ways to handle this.

ICA with maximum non-Gaussianity: FastICA algorithm

https://en.wikipedia.org/wiki/FastICA (https://en.wikipedia.org/wiki/FastICA)
http://labs.seas.wustl.edu/bme/raman/Lectures/Lecture14_ICA.pdf (http://labs.seas.wustl.edu/bme/raman/Lectures/Lecture14_ICA.pdf)

- Note scikit learn has implementation for FastICA algorithm.
- In FastICA, we will find an optimal vector w such that $y = w^T X$ is equal to one of the sources signal. Here X is a matrix including all the signals observed at different time points.
- According to central limit theorem the extracted signal from X, $y = w^T X$, is more like a Gaussian than the source s, as the y is a linear combination of source s ($y = w^T X = w^T As$). When y is exactly equal to one of the source s, then it become least Gaussian. Therefore, we should find a vector w^T that maximizes the non-Gaussianity of $w^T X$, since this will make y equal to one of the sources. **From above, the column of X should be the data at different time points?**
- An appropriate measure of non-Gaussianity is the fourth moment (kurtosis). Therefore to obtain the optimal w^T, we just maximize the kurtosis and set it to be zero. Unlike updating the whole W matrix in each numerical iterative step in MLE approach, here we update only one vector w^T by maximizing the non-Gaussianity.
- The above way is actually very standard. For example, in PCA, LDA, we also projecting data X to some vector w (equivalent to projecting or transforming to another basis). Then in this new basis we optimize some quantity (PCA:variance, LDA:separation...) and thus determine w, or the new basis vector.
- Before the maximization process, it is necessary to do the **centering and whitening** of the data. Centering consists of subtracting the mean of the observation vector. Whitening consists of applying a linear transform to the observations so that its components are uncorrelated and have unit variance. Whitening makes the mixing matrix orthogonal which has the advantage of halving the number of parameters that need to be estimated, since an orthogonal matrix only has $\frac{n(n-1)}{2}$ free parameters. This is working in an eigenbasis.

ICA with singular value decomposition (SVD)

Video: https://www.youtube.com/watch?v=olKgmOuAvrc (https://www.youtube.com/watch?v=olKgmOuAvrc)
The approach below is strongly connected to the method in the previous cell, also involving maximizing Kurtosis.

Using SVD, decompose the mixing matrix as $A = U\Sigma V^T$. Because U, V are both unitary operator, and Σ are diagonal, we are essentially do three steps to the source data in the following order:

- Rotate the original image_0 (data) with the matrix V^T to a target image_1.
- Stretch the image_1 with the diagonal elements of Σ to a target image_2.
- Rotate the image_2 with U to a target image_3.

Now we are going to take a reverse order to obtain image_0 from image_3. The image_3 corresponds to our observed signal x.

- First we try to calculate the angle we rotate image_2 to image_3 (by unitary matrix U). Assuming the initial side of this angle is at zero degree, then the terminal side of this angle should be along the direction with maximum variance. Calculating variance of x with a function of angle θ and then setting first derivative to be zero, we could obtain the angle θ_0 at which variance is maximized. Using θ_0 we can construct the unitary (rotational) matrix U, with which we rotate image_2 to image_3. Thus we can use U^T to rotate image_3 back to image_2.
- After going back to image_2, we can un-stretch it back to image_1. We first calculate the variance along θ_0 and $\theta_0 - \frac{\pi}{2}$. Taking square roots of the calculated two variances we obtain σ_1 and σ_2. Using them to construct a diagonal matrix, we then obtain the 'stretching' matrix Σ that rotates image_1 to image_2. Finally, using its inverse Σ^{-1}, we can unstretch image_2 back to image_1.
- To rotate image_1 back to image_0, we calculate a angle that maximize the kurtosis of image_1, and then we can use the calculated angle to rotate image_1 to obtain the original image_0. The idea of maximizing kurtosis is similar to the previous section where we try to constrain the independence condition as much as possible. This is similar to the previously introduced fastICA algorithm.

ICA with entropy or information

This is another quantity used to achieve independence. Check details later.

Comparison of PCA and ICA

- PCA construct mutually uncorrelated and orthogonal new axes by maximizing variance. This is not the mutually independent new axes in ICA. However, in one particular case where the data is Gaussian, then the uncorrelated axes obtained by maximizing variance in PCA is same as the independent axes as in ICA. This is because Gaussian data just has no beyond 2nd-order moments (variance). In this particular case, PCA amounts to find independent Gaussians (new features).
- In general case when the data is not Gaussian, PCs in PCA are just uncorrelated but not necessarily independent. ICA looks for maximally independent components. Here the independence is a stronger concept than uncorrelated components as in PCA. Uncorrelation only measures linear relationship, while independence measures the existence of any relationship.
- Blind source separation (BSS), ICA does well while PCA does bad.
- When rotating the data from $m \times n$ to $n \times m$, PCA give same results while ICA is not and therefore highly directional.
- The first component of PCA gives brightness (high variance), then second is average face,... while ICA gives nose, eyes, etc (ICA gives high spatial frequency stuff?). For natural scenes ICA gives edges. For documents, ICA gives topics.
- **PCA maximize second moment (variance) to kill correlation and maximize uncorrelatedness. One way of of ICA is to maximize fourth moment (kurtosis) to kill dependence and maximize independence (non-Gaussianity).**
- Draw a cartoon showing how PCA find directions (eigenvectors) that maximize variance and also let the directions orthogonal (and thus components uncorrelated). **Then draw a cartoon showing ICA find directions maximize independence, or non-Gaussianity**. Explain this with a picture showing two clouds of data along different directions, and showing why PCA is not as good as ICA in this

case.

- ICA components usually has more real-life meanings such as the independent speaker voice, and therefore ICA is more like an unsupervised learning algorithm than PCA, even though PCA is also an unsupervised learning algorithm. The components of both algorithms can be regarded as controlled by hidden variables.

Downside of ICA

- ICA works well in problems where we wish to un-mix truly independent signals. For instance, separating distinct audio signals in the "cocktail party" problem. There is a downside to ICA, as the original Comon paper points out. **If one finds K components with ICA, it does not rank the importance of the components or the amount of variance of the data explained by the component consistently.** If you compared ICA with K+1 components vs. K components these can look extremely different. Whereas, with PCA the first K components are always ranked by how much of the signal/variance of the data they explain. Moreover, these components are reproducible should you choose to find a larger number of components (If you apply PCA twice to similar data, the first K components are the same). **If you have no basis for believing the your signal components come from "independent" sources, PCA or its variations are still good or at least capture information upto the second moment in your signal.** https://www.quora.com/Are-there-implicit-Gaussian-assumptions-in-the-use-of-PCA-principal-components-analysis (https://www.quora.com/Are-there-implicit-Gaussian-assumptions-in-the-use-of-PCA-principal-components-analysis)
- For Gaussian signal, ICA cannot restore the source signals. However, earlier we mention that for Gaussian signals PCA and ICA are equivalent, then if we do PCA, what we will get?