Factor Analysis and Independent Component Analysis

Factor Analysis

- In Factor Analysis, we assume that an observation x ∈ R^d is generated from a latent representation z ∈ R^p where $p \leq d$.
- We can consider Factor Analysis as a generative model, and this probabilistic model allows us to deal with missing values in the data set by running generatively to provide samples from the distribution.
- We can also consider Factor Analysis as a discriminative model since we can model class-conditional densities that can be applied to classification problems.
- We will focus on Probabilistic Principle Component Analysis (PPCA), which in one type of factor analysis, in the assignment .

Basis of Factor Analysis

In Factor Analysis, we assume a data point x_i is obtained by:

1. Linearly project a corresponding latent point z_i to a higher dimension space by matrix $\Lambda \in R^{\{d \times p\}}$
2. Apply some linear translation μ
3. Add Gaussian noise ϵ with covariance matrix Ψ

$$x_i = \Lambda z_i + \mu + \epsilon$$
$$P(z_i) = N(0, I)$$
$$P(\epsilon) = N(0, \Psi)$$

Note that N stands for a normal/Gaussian distribution. We assume that noises in different dimension are independent from each other, so Ψ is diagonal.

Conditional and Marginal Probability

From the above assumption, we can conduct that

$$p(x|z) = N(\Lambda z + \mu, \Psi)$$
$$p(x) = \int p(x|z)p(z)dz$$

Since both $p(x|z), p(z)$ are Gaussian, $p(x)$ should also be Gaussian:

$$E(x) = E(\Lambda z + \mu + \epsilon) = \ \mu$$
$$cov(x) = E(\big(\Lambda z + \mu + \epsilon - E(x)\big)\big(\Lambda z + \mu + \epsilon - E(x)\big)^T)$$
$$= E\big((\Lambda z + \epsilon)(\Lambda z + \epsilon)^T\big)$$
$$= E(\Lambda z z^T \Lambda^T) + \Lambda E(z\epsilon^T) + E(\epsilon z^T)\Lambda^T + E\ (\epsilon\epsilon^T)$$

Since we assume ϵ and z are independent, they are uncorrelated →

$E(z\epsilon^T) = E(\epsilon z^T) = 0 \rightarrow cov(x) = E(\Lambda z z^T \Lambda^T) + E\ (\epsilon\epsilon^T) = \Lambda\Lambda^T + \Psi \rightarrow$

$$p(x) = N(\mu, \Lambda\Lambda^T + \Psi)$$

- Now we have

$$p(x) = N(\mu, \Lambda\Lambda^T + \Psi)$$
$$p(x|z) = N(\Lambda z + \mu, \Psi)$$
$$p(z) = N(0, I)$$

We can use the Bayesian Theorem to get

$$p(z|x) = N(\Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}(x - \mu), I\ - \Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}\Lambda)$$

(Note that it can also be derived from the joint distribution of x and z)

Maximum Log Likelihood

To determine the value of μ, Λ, Ψ, we would like to set the partial derivatives of the MLE $(\log p(X) = \sum logp(x_i))$ with respect to μ, Λ, Ψ to be zero, and then solve the equations. Unfortunately, we don't have closed-form solution for the factor analysis.

We do have closed-form solution if we assume Ψ has identical diagonal entries-–this is called PPCA, and calculation of the closed-form solution would be shown later.

As a result, we need to implement an EM algorithm to determine the value of μ, Λ, Ψ

EM algorithm

We learned EM algorithm in Gaussian Mixture Model, here is a brief review of it:

We define Evidence Lower Bound (ELBO) as

$$L(q,\theta) := -KL(q(z)||p(x,z|\theta)),$$

and we can proof that $L(q,\theta) \leq \log(p(x|\theta))$

(Notice that $p(x,z|\theta) = p(x,z)$ and $p(x|\theta) = p(x)$ in my notation, θ is the parameter set--in this case, μ, Λ, Ψ. And $q(z)$ is the prior probability of the latent variable.)

Increase ELBO will increase $\log(p(x|\theta))$.

In E-step, we want to find the best distribution q(z) to maximize the lower bound. And in M-step, we find the best parameter(s) θ to maximize the lower bound.

E-step:

Since $L(q,\theta) = \log\big(p(x|\theta)\big) - KL(q|p(z|x))$, and $0 \leq KL$, we maximum $L(q,\theta)$ by letting $KL(q|p(z|x))$ = 0 $\leftrightarrow$ q = p$(z|x)$.

M-step:

We take the derivative with respect to θ and let it equals to zero. By solving the equation, we can find best parameter(s) θ.

EM algorithm for Factor analysis

Here is the summary of factor analysis algorithm by centering the X (i.e., letting μ = 0):

1. Initial Λ, Ψ
2. Iteratively update the following four quantities until the log likelihood converged.

$$Mean(z_i) = \Lambda^{(t)T}\left(\Lambda^t\Lambda^{(t)T} + \Psi^t\right)^{-1}x_i$$

$$Cov(z_i) = I\ - \Lambda^{(t)T}\left(\Lambda^t\Lambda^{(t)T} + \Psi^t\right)^{-1}\Lambda^t$$

$$\Lambda^{t+1} = (\sum_{i=1}^{n}(\Psi^t)^{-1}x_i Mean(z_i)^T)(\sum_{i=1}^{n}(\Psi^t)^{-1}\Lambda^t Cov(z_i))$$

$$\Psi^{t+1} = (1/n)diag(\sum_{i=1}^{n}[x_i x_i^T - 2\Lambda^{t+1}Mean(z_i)x_i^T + \Lambda^{t+1}Cov(z_i)\Lambda^{t+1}])$$

Probabilistic PCA

- PPCA is a special case of factor analysis by considering an isotropic noise in the original space: $P(\epsilon) = \sigma^2 I$
- The probabilistic PCA model expressed a high-dimensional vector **x** as a linear combinations of basis vectors + noise.
- This is accomplished by formulating PCA as a maximum likelihood of a probabilistic latent variable model.
- In PPCA we want to estimate $p(\boldsymbol{x})$**.**
- To do this we use a linear-Gaussian framework where all the marginal and conditional distributions are Gaussians.
- First part of the formulation is to define the latent variable **z** representing the principle component subspace.

Probabilistic PCA

- With the linear-Gaussian framework the prior $p(\boldsymbol{z})$ is defined as a Gaussian:

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\mathbf{0}, \boldsymbol{I})$$

- The conditional $p(\boldsymbol{x} \mid \boldsymbol{z})$ [observed **x** conditioned on latent **z**] is also a Gaussian defined as:

$$p(x|z) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \sigma^2\boldsymbol{I})$$

- Columns of **W** span a linear subspace within the principle subspace. Variable σ^2 controls the variance of the conditionals.
- Note that there is no loss of generality in the assumption of a zero mean and unit covariance Gaussian for $p(\boldsymbol{z})$**.**

Probabilistic PCA MLE

- A closed form exact solution can be derived for PPCA by defining the marginal distribution p(x) by using the sum and product rules of probability giving:

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$$

- Again, using the linear-Gaussian framework, p(x) is Gaussian and expressed as:

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{C})$$

- Here **C** is a (d x d) covariance matrix defined:

$$\boldsymbol{C} = \boldsymbol{W}\boldsymbol{W}^T + \sigma^2\boldsymbol{I}$$

Probabilistic PCA MLE

- We can determine closed form solution of the model parameters using maximum likelihood.

- To do this we want to maximize the probability of data set **X** expressed as:

$$p(X|\mu, W, \sigma^2) = \prod_{n=1}^{N} p(x_n|W, \mu, \sigma^2)$$

- The log likelihood is then:

$$log\ p(X|\mu, W, \sigma^2) = \sum_{n=1}^{N} log\ p(x_n|W, \mu, \sigma^2)$$

Probabilistic PCA MLE

MLE μ :

$$\nabla_{\mu}\left(-\frac{Nd}{2}log2\pi-\frac{N}{2}log|C|+\sum_{n=1}^{N}\frac{1}{2}(x_n-\mu)^T C^{-1}(x_n-\mu)\right)=\sum_{n=1}^{N}C^{-1}(x_n-\mu)$$

- Set = 0

$$0=\sum_{n=1}^{N}C^{-1}x_n-C^{-1}\mu$$

$$\sum_{n=1}^{N}x_n=\sum_{n=1}^{N}\mu\ =\ \mu N$$

$$\mu=\frac{1}{N}\sum_{n=1}^{N}x_n$$

Probabilistic PCA MLE

MLE **W**:

- Using the identity scalar $y^T A\, y = Tr(y^T A\, y)$ the log likelihood becomes:

$$logp(x_n|W, \mu, \sigma^2) = -\frac{Nd}{2} log2\pi - \frac{N}{2} log|C| + \frac{1}{2}\sum_{n=1}^{N} \mathrm{Tr}\big((x_n - \mu)^T C^{-1}(x_n - \mu)\big)$$

- Using the identity for traces that $Tr(ABC) = TR(CAB)$:

$$-\frac{Nd}{2} log2\pi - \frac{N}{2} log|C| + \frac{1}{2}\sum_{n=1}^{N} Tr(\,(x_n - \mu)(x_n - \mu)^T C^{-1}\,)$$

Probabilistic PCA MLE

MLE **W** continued :

- Taking further advantage of trace identities we can use $\sum_i Tr(A_i B) = Tr(\sum_i A_i B)$ to exchange the sum and trace to give:

$$-\frac{Nd}{2} log 2\pi - \frac{N}{2} log|C| + \frac{1}{2} Tr\left(\sum_{n=1}^{N} (x_n - \mu)(x_n - \mu)^T \mathrm{C}^{-1}\right)$$

$$= -\frac{Nd}{2} log 2\pi - \frac{N}{2} log|C| + \frac{N}{2} Tr\left(\frac{1}{N}\sum_{n=1}^{N} (x_n - \mu)(x_n - \mu)^T \mathrm{C}^{-1}\right)$$

Probabilistic PCA MLE

MLE **W** continued :

- To make things a bit easier lets define the sample covariances **S** as:

$$S = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)\,(x_n - \mu)^T$$

And this gives:

$$= -\frac{Nd}{2}log2\pi - \frac{N}{2}log|C| + \frac{N}{2}Tr(S\mathrm{C}^{-1})$$

- To solve for **W** we sub back in for **C** :

$$= -\frac{Nd}{2}log2\pi - \frac{N}{2}log|WW^T + \sigma^2 I| + \frac{N}{2}Tr(S(WW^T + \sigma^2 I)^{-1})$$

Probabilistic PCA MLE

MLE mean **W** continued :

- Now take the gradient:

$$\nabla_W = -\frac{NW}{WW^T + \sigma^2 I} + \frac{NSW}{(WW^T + \sigma^2 I)^2}$$

- Set to zero and solve for **W:**

$$1 = \frac{S}{WW^T + \sigma^2 I}$$

$$\sigma^2 I = S - WW^T$$

- This has a known solution:

$$W = U_m(\Lambda_m - \sigma^2 I)^{\frac{1}{2}}R$$

Probabilistic PCA MLE

MLE mean **W** continued :

- To get the known solution we express **W** as its SVD.

$$\boldsymbol{W} = \boldsymbol{ULV^T}$$

- Where **U** is a (d x q) matrix. **L** is a (q x q) matrix and **V** is a (q x q).

- Now sub this into the gradient equation and with some manipulation we arrive at:

$$\boldsymbol{W} = U_q\left(\mathrm{K}_q - \sigma^2 I\right)^{\frac{1}{2}} R$$

- Where U_q has columns of eigenvectors and $\mathbf{K}_q$ is the diagonal matrix of the eigenvalues.

$$k_j = \begin{cases} \lambda_j, & for\ corresponding\ u_j \\ \sigma^2, & otherwise \end{cases}$$

- When $k_j = \sigma^2$ we see that we will have zero entries on our diagonal due to $\boldsymbol{K}_q - \sigma^2 I$

- R can be taken as R = I. It is independent of **C**.

Probabilistic PCA MLE

MLE σ^2 :

- Starting from the previously stated log likelihood function using the identity for traces that $Tr(ABC) = TR(CAB)$:

$$logp(X|\mu, W, \sigma^2) = -\frac{Nd}{2} log2\pi - \frac{N}{2} log|C| + \frac{1}{2}\sum_{n=1}^{N} Tr(\ (x_n - \mu)(x_n - \mu)^T C^{-1}\ )$$

- Then subbing in the **S** for the sample covariances and the definition for **C**:

$$logp(X|\mu, W, \sigma^2) = -\frac{Nd}{2} log2\pi - \frac{N}{2} log|WW^T + \sigma^2 I| + \frac{N}{2} Tr(S(WW^T + \sigma^2 I)^{-1})$$

Probabilistic PCA MLE

MLE σ^2 continued:

- Tipping and Bishop (1999b) proved that MLE function is at a maximum when the q eigenvectors are coincident with the q largest eigenvalues.

- To arrive to the above statement, we sub in our previously derived MLE for **W** to get:

$$= -\frac{Nd}{2} log(2\pi) + \sum_{j=1}^{q'} log(\lambda_j) + \frac{1}{\sigma^2} \sum_{j=q'+1}^{d} \lambda_j + (d - q') log(\sigma^2) + q'$$

- Here q' is the number of non zeros eigenvalues $\boldsymbol{K}_q - \sigma^2 I$: $\lambda_1, \ldots, \lambda_{q'}$ corresponding to the retained eigenvectors in **W.** $\lambda_{q'}, \ldots, \lambda_d$ are the discarded eigenvectors.

Probabilistic PCA MLE

MLE σ^2 continued:

- Now calculate the gradient:

$$\nabla_{\sigma^2} = \frac{1}{(\sigma^2)^2} \sum_{j=q'+1}^{d} \lambda_j + \frac{d - q'}{\sigma^2}$$

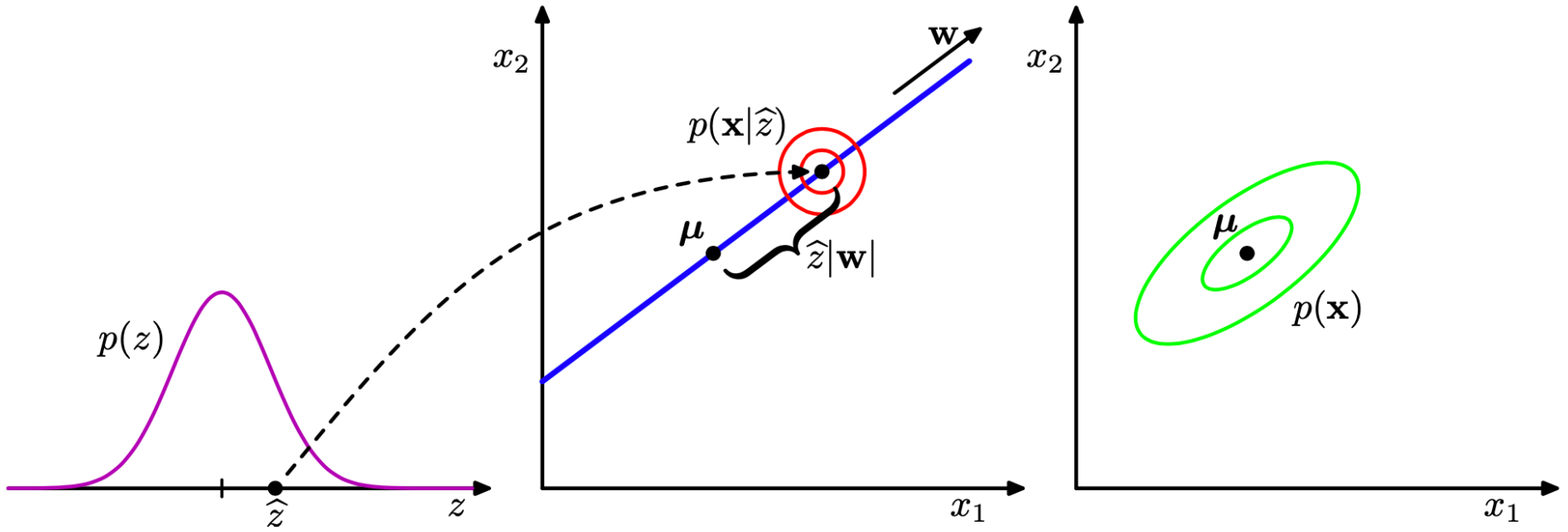
- Set to zero and solve:

$$\frac{1}{(\sigma^2)^2} \sum_{j=q'+1}^{d} \lambda_j = \frac{d - q'}{\sigma^2}$$

$$\sigma^2 = \frac{1}{d - q'} \sum_{j=q'+1}^{d} \lambda_j$$

- Note that the MLE for σ^2 is the average of the discarded dimensions eigen values.

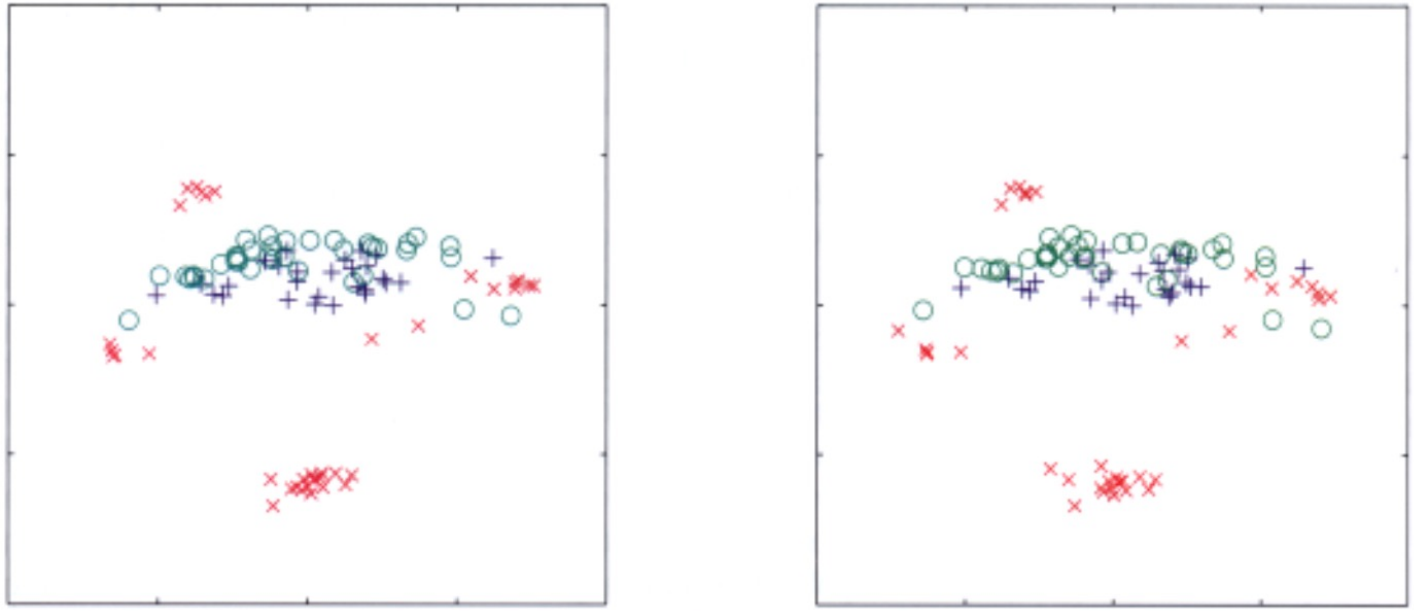
Graphic interpretation of PPCA



--Bishop, Pattern Recognition and Machine Learning

z is the latent variable with dimension 1 and x is the observe data with dimension 2. p(z), p(x|z), and p(x) are all Gaussians.

Application of PPCA



Probabilistic PCA visualization of a portion of the oil flow data set for the first 100 data points. The left-hand plot shows
the posterior mean projections of the data points on the principal subspace. The right-hand plot is obtained by first randomly
omitting 30% of the variable values and then using EM to handle the missing values. Note that each data point then has at least
one missing measurement but that the plot is very similar to the one obtained without missing values.

--Bishop, Pattern Recognition and Machine Learning

Independent Component Analysis

- As PCA, ICA assumes that the observed variables are linear combinations of latent variables, but the latent variables in ICA are hypothesized to follow non-Gaussian distributions and are statistically independent.
- ICA is a method to recover the original sources (called independent components) from the observed data by multiplying the observed data by a unmixing matrix; therefore, in principle, the number of independent components is equal to the number of observed data samples.

Intuitive explanation -- blind source separation

Consider the blind source separation in cocktail-party problem:

Two people are talking at the same time in different places in a party, and their voices are recorded in two microphones in different locations. In such scenario, the observed data is the two recordings from the two microphones, and the independent components (latent variables) are the signals each of which contains only the voice of just one person.

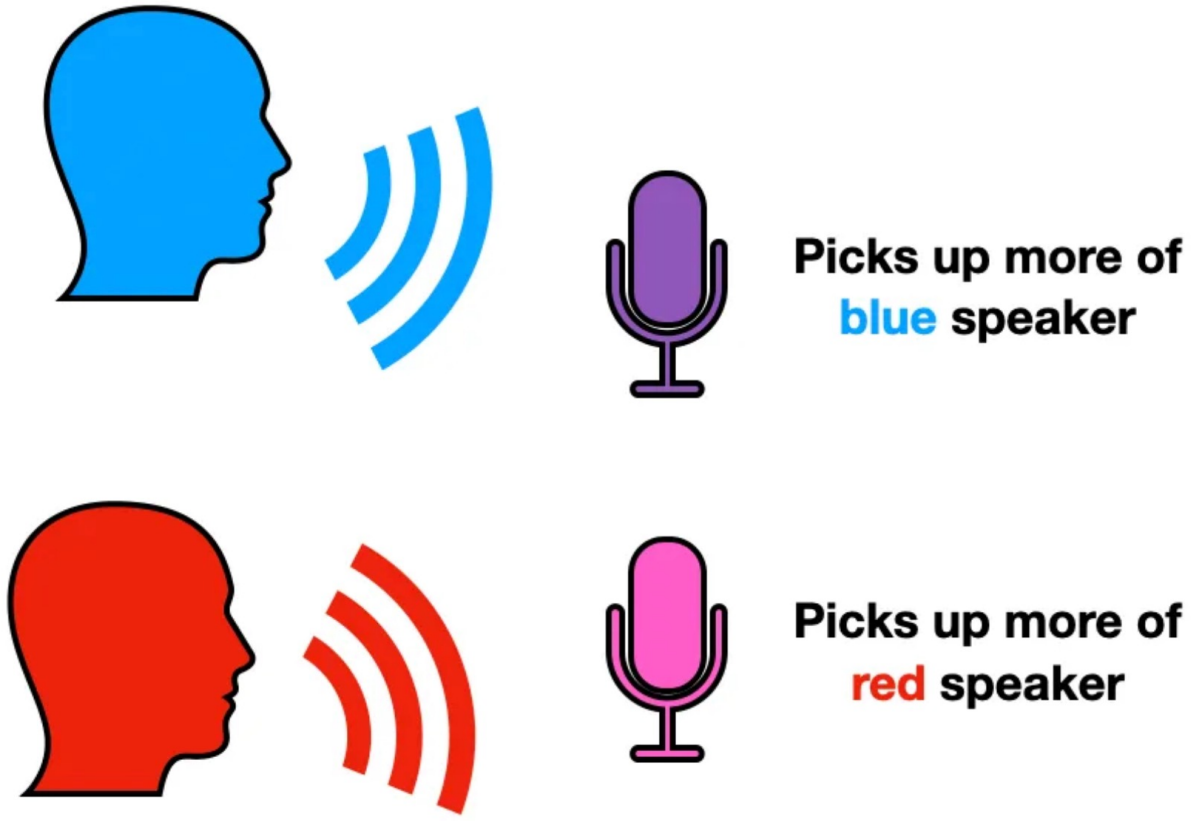


Image from https://towardsdatascience.com/independent-component-analysis-ica-a3eba0ccec35

Mathematics explanation

- Assume that we observe n linear mixtures $x_1, \ldots, x_n$ of n independent components and for all 0<j<n+1:

$$x_j = a_{j1}s_1 + \cdots + a_{jn}s_n$$

- Using the vector-matrix notation, we have x = As and s = Wx where $W = A^{-1}$. In other words, we want to find the unmixing matrix W.
- Let's consider a linear combination of the x_i, denoted as $y = w^T x = w^T As = z^T s$, where $z = A^T w$. If w is one row of the unmixing matrix W , then y is equal to one of the independent components. Namely, we want to find w such that z only has one non-zero entity. However, we have no knowledge of A, so we cannot directly compute w; therefore, we need to measure the non-Gaussianity of y to find w.

“ Nongaussian is independent”

Central Limit Theorem:

A sum of two independent random variables usually has a distribution that is closer to gaussian than any of the two original random variables.

Since a sum of even two independent random variables ($s_m, +s_j$) is more gaussian than the original variables (s_m, s_j), y becomes least gaussian when it in fact equals one of the independent random variables; therefore, we want to maximize the non-Gaussianity of y to find w.

Measure of Nongaussianity

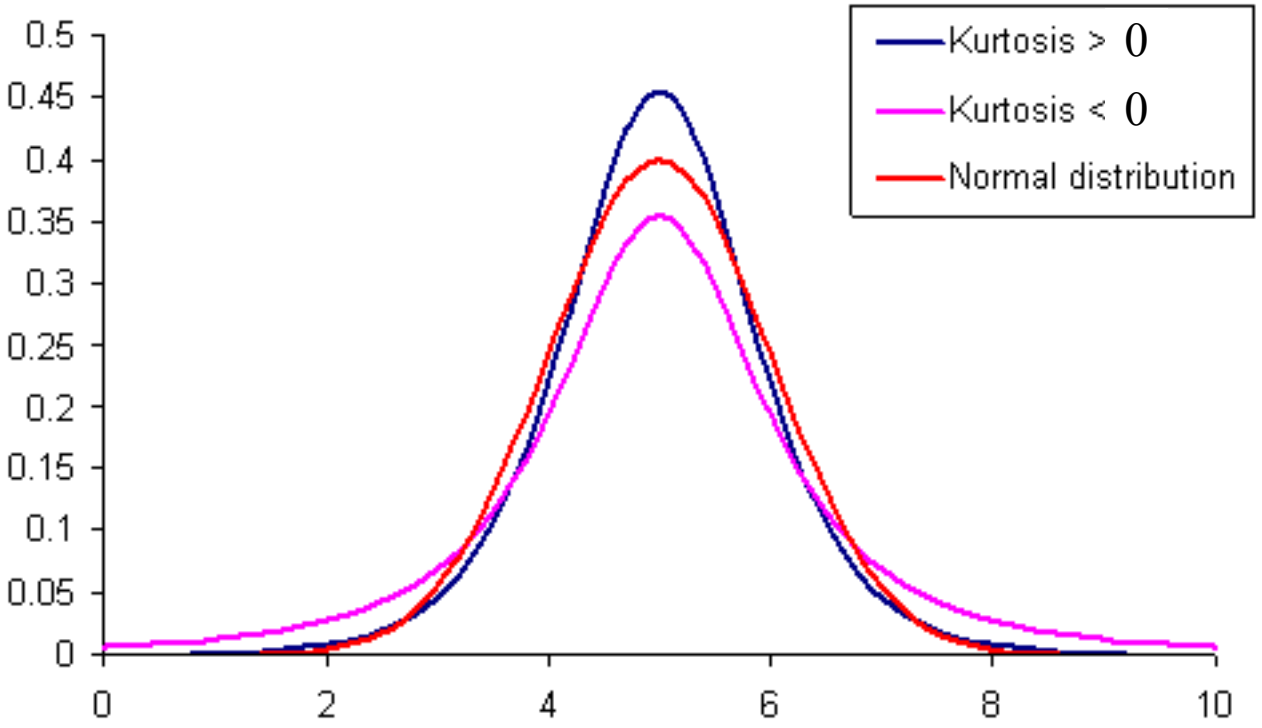
- Kurtosis

$$kurt(y) = E\{y^4\} - 3(E\{y^2\})^2$$

Kurtosis is zero if y is Gaussian;

Kurtosis is greater than zero if y is heavy-tailed (e.g., Laplacian)

Kurtosis is less than zero if y is light-tailed (e.g., uniform)



Negentropy

- Entropy

$$H(y) = -\int f(y) log f(y) dy$$

where f is the pdf of the random variable y. A gaussian distribution f has the largest entropy among all distribution of equal variance.

Entropy is small for distributions that are clearly concentrated on certain values, i.e., when the variable is clearly clustered, or has a pdf that is very "spiky".

- Negentropy

$$J(y) = H(y_{gauss}) - H(y)$$

where y_{gauss} is a Gaussian random variable of the same covariance matrix as y.

Negentropy is always non-negative, and it is zero if and only if **y** has a Gaussian distribution.

Approximation of negentropy

$$J(y) \approx [E\{G(y)\} - E\{G(v)\}]^2$$

where G is non-quadratic, v is a Gaussian variable of zero mean and unit variance.

Common choice of G:

$$G_1(u) = \left(\frac{1}{a_1}\right) logcosha_1 u, 1 \leq a_1 \leq 2$$

$$G_2(u) = -\exp\left(-\frac{u^2}{2}\right)$$

These approximations are less computational expensive than negentropy and more robust than kurtosis.

FastICA

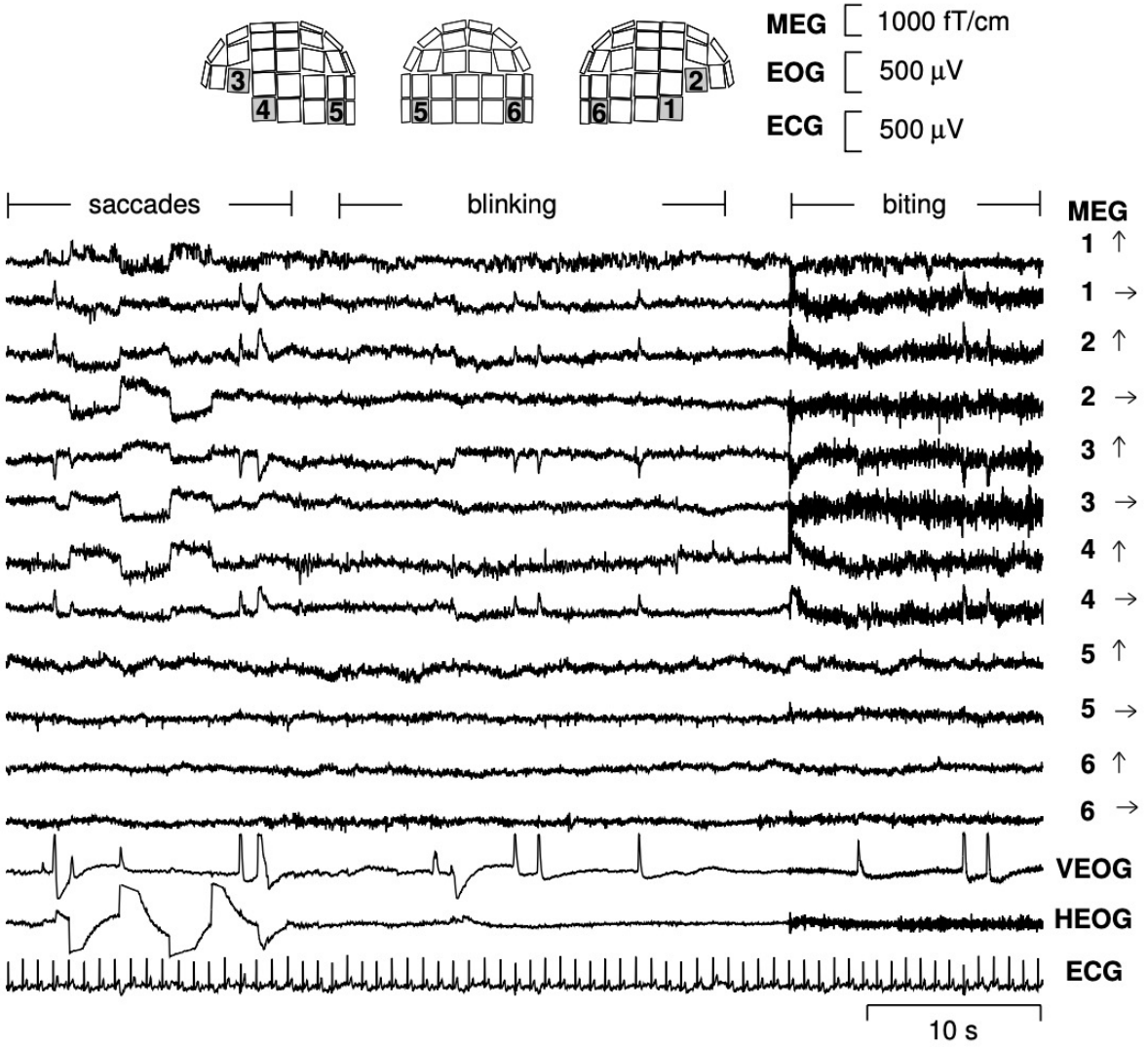
- The FastICA learning rule finds a direction, i.e., a unit vector w such that the projection $w^T x$ maximizes nongaussianity. Nongaussianity is measured by the approximation of negentropy in the previous slide.
- Assume that the data is prewhitened (components in X are uncorrelated and their variances equal unity), the we use the following iteration scheme for finding a maximum of the nongaussianity of $w^T x$:

1. Choose an initial weight vector w.
2. Let $w^+ = E\{xG'(w^T x)\} - E\{G''(w^T x)\}w$
3. Let $w = w^+ / \left|\left|w^+\right|\right|$
4. If the dot product of old and the new w is almost equal to 1 (converged), then return w; otherwise (not converged), go back to 2.

Ambiguities of ICA

- As PCA, multiplying an independent component by -1 will not affect the model.
- The order if the independent components can be be deterministic.
- The w (as well as the independent components) would change if we use different approximations ($G_1 \; or \; G_2$)
- The independent components can be estimated one by one. This is useful since it decreases the computational load of the method in cases where only some of the independent components need to be estimated; therefore, you may see the number of independent components is not equal to the number of sample data in certain case studies.

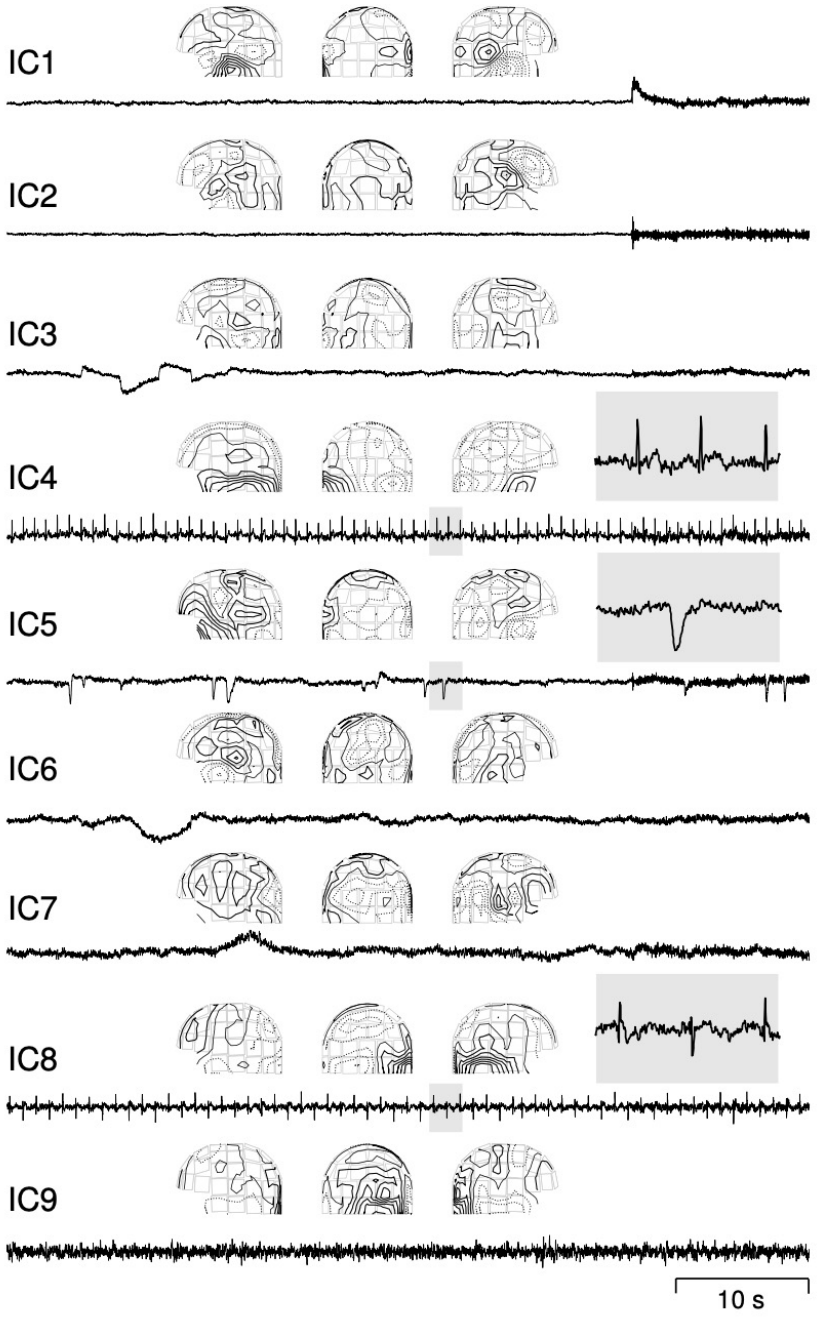
ICA application—MEG data



Magnetoencephalography (MEG) is a non-invasive technique where the activity or the cortical neurons can be measured with very good temporal resolution and moderate spatial resolution.

We can use ICA to separate brain activity from artifacts (e.g., eye movements or blinks).

The plot on the left shows 9 MEG signals, total of 122 signals were recorded and were pre-processed using PCA, then the authors used 9 independent components to illustrate that the signals generated by the brain activity and the artifacts are statistical independent.



The plot on the left shows 9 independent components (ICs). The first two ICs are clearly due to the muscular activity originated from the biting.

IC3 and IC5 show the horizontal eye movements and the eye blinks, respectively. IC4 represents the cardiac artifact that is very clearly extracted.

The results not only show that the FastICA algorithm is possible to isolate the brain activities and different types of artifacts, but also support the authors' hypothesis that the brain activities and the artifacts are physiologically separate processes.

Conclusion

- Factor analysis and PPCA are generative models that can deal with the missing data issue. The latent variables z can also be used in discriminant models.
- ICA is a dimension reduction method. As Factor analysis and PPCA , ICA also uses linear projection between the original space (with observed data in it) and latent space, the fundamental different among them is that ICA assumes that the independent components are non-Gaussian and statistically independent.

Reference

- Vigário, R. (1997). Extraction of ocular artifacts from EEG using independent component analysis. *Electroenceph. Clin. Neurophysiol.*, 103(3):395–404.
- Vigário, R., Jousmäki, V., Hämäläinen, M., Hari, R., and Oja, E. (1998). Independent component analysis for identification of artifacts in magnetoencephalographic recordings. In *Advances in Neural Information Processing Systems*, volume 10, pages 229–235. MIT Press.
- Bishop, Christopher M. (2006). Pattern recognition and machine learning*. New York :Springer*
- Ghojogh, B., Ghodsi, A., Karray, F., & Crowley, M. (2021). Factor analysis, probabilistic principal component analysis, variational inference, and variational autoencoder: Tutorial and survey. *arXiv preprint arXiv:2101.00734*.
- A. Hyvarinen and E. Oja, Independent Component Analysis: Algorithms and Applications, Neural Networks, 13(4-5), 2000, pp. 411-430.
- Tipping, Michael E., and Christopher M. Bishop. 'Probabilistic Principal Component Analysis'. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 61, no. 3 (1 September 1999): 611–22.
- Minka, Thomas. 'Automatic Choice of Dimensionality for PCA'. In *Advances in Neural Information Processing Systems*, edited by T. Leen, T. Dietterich, and V. Tresp, Vol. 13. MIT Press, 2000.