Solutions for week 2

Hebbian rules and ICA

Exercise 1: Optimality criteria for PCA: variance and optimal reconstruction

In lecture 1 we found that PCA is a result of Hebbian learning. We now ask whether PCA (and Hebbian learning rules!) can be derived from optimality criteria.

As usual we have a set of P input patterns $\mathbf{x}^\mu$ with $1 \leq \mu \leq P$. We assume that $\mathbb{E}_{data}[\mathbf{x}] = 0$.

A (i) Define an output $y = \sum_k w_k x_k$.

Derive the batch update rule then the single-sample update rule to maximize the variance $\mathbb{E}_{data}[y^2]$ by gradient ascent.

Solution:

We want to minimize the loss $\mathcal{L} = \mathbb{E}_{data}[y^2]$.

$$\begin{aligned}\frac{\partial}{\partial w_k}\mathbb{E}_{data}[y^2] &= \mathbb{E}\left[\frac{\partial}{w_k}\sum_{i,j} w_i w_j x_i x_j\right] \\ &= \mathbb{E}\left[\sum_{i\neq k} w_i x_i x_k + \sum_{j\neq k} w_j x_j x_k + 2w_k x_k^2\right] \\ &= \mathbb{E}\left[2x_k \sum_i w_i x_i\right]\end{aligned}$$

In vector form, this is written $\frac{\partial \mathcal{L}}{\partial w} = \mathbb{E}[2xw^T x] = \mathbb{E}[2xx^T]w = 2Cw$. Therefore the update rule is $w^{new} = w^{old} + 2\gamma C w^{old}$.

The online rule is obtained by dropping the average and using a single sample estimator of $\frac{\partial \mathcal{L}}{\partial w}$, namely $\frac{\partial \mathcal{L}}{\partial w}^{online} = 2xw^T x$.

(ii) Turn the update equation of gradient ascent into a differential equation. Compare your result to the equation we found in Lecture 1.

Solution:

For small enough γ, the batch and online rule respectively become the following differential equations: $\dot{w} = 2Cw$ and $\dot{w}^{online} = 2xw^T x$.

(iii) Assume now that the weight vector is normalized, i.e., we maximize variance for a normalized vector. Express the weight vector in terms of the Eigenvectors of the correlation matrix. Convince yourself that the variance is maximal if the only nonzero component is the projection on the first Eigenvector.

Solution:

We assume that $||w||^2 = 1$ at each step (through renormalization). The eigen decomposition is $w = \sum_k \langle w, e_k\rangle e_k$ where $e_1, e_2, ...e_n$ are orthonormal because C is symmetric, with eigenvectors $e_1, e_2, ...e_n$ corresponding to eigenvalues $\lambda_1 \geq ... \geq \lambda_n$. Then,

$$\mathbb{E}[y^2] = \mathbb{E}\left[(w^T x)^2\right] = \mathbb{E}\left[w^T x x^T w\right] = w^T C w = w^T \sum_k \lambda_k \langle w, e_k\rangle e_k = \sum_k \lambda_k \langle w, e_k\rangle^2$$

Since $\sum_k \langle w, e_k \rangle^2 = ||w||^2 = 1$, the variance $\mathbb{E}[y^2]$ is maximized when $\langle w, e_1 \rangle^2 = 1$, *i.e.* when w is colinear with e_1 the eigenvector with maximal eigenvalue.

(iv) Go back to point (i) and switch to a presentation in terms of the vector component. Intepret the result as a Hebbian learning rule and identify the presynaptic and postsynaptic terms.

Solution:

We write the result from (i) $\frac{\partial \mathcal{L}}{w}^{online} = 2xw^T x$ for each vector component i:

$$\frac{\partial \mathcal{L}}{w_i}^{online} = 2 \underbrace{x_i}_{\text{pre}} \underbrace{w^T x}_{\text{post}}$$

B. The aim of an autoencoder is to compress a set of high-dimensional data points into a low-dimensional representation such that a reconstruction of the input is possible at minimal loss. Assume a linear autoencoder consisting of one hidden layer of a single neuron $y = \sum_k w_k x_k$. The weights from the hidden layer to the output are $w_k^{\text{out}} = w_k$.

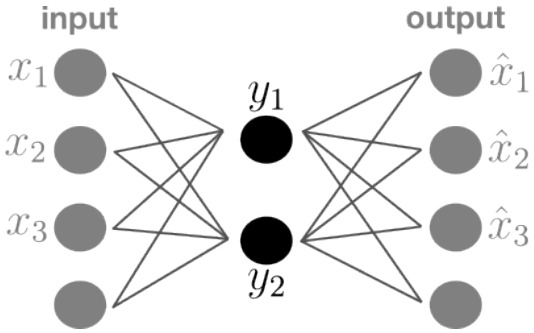

Figure 1: Architecture of an autoencoder with one hidden layer of two hidden neurons.

(i) Minimize the reconstruction error

$\frac{1}{2P} \sum_\mu \|\mathbf{x}^\mu - \hat{\mathbf{x}}^\mu\|^2$

where $\hat{x}_k^\mu = w_k^{\text{out}} y$.

Derive first two separate batch rules, one for the output weights and one for the input weights.

Solution:

In vector notation, we have $\mathcal{L} = \frac{1}{2}\mathbb{E}\left[\|x - \hat{x}\|_2^2\right]$ with $\boldsymbol{x} = y\boldsymbol{w}^{out}$ and $y = \boldsymbol{w}^T\boldsymbol{x}$. Now we differentiate this loss with respect to weights (omitting the expectation sign):

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}^{out}} = (\boldsymbol{x} - \hat{\boldsymbol{x}})(-y) = -(\boldsymbol{x} - \boldsymbol{w}^{out} y)y = -\left(\boldsymbol{x}y - \boldsymbol{w}^{out} y^2\right) \tag{1}$$

This has the form of Oja's rule but notice that for the weights that are being updated $\boldsymbol{w}^{out}$, y is the input and x is not the input. Therefore the update based on (1) does not have the *pre* $\times$ *post* form of a Hebbian learning rule.

Now for the input weights, using the fact that $\boldsymbol{w}^{out} = \boldsymbol{w}$:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} &= -(\boldsymbol{x} - \hat{\boldsymbol{x}})^\top \frac{\partial \hat{\boldsymbol{x}}}{\partial \boldsymbol{w}} \\
&= -(\boldsymbol{x} - \hat{\boldsymbol{x}})^\top \boldsymbol{w}^{out} \frac{\partial y}{\partial \boldsymbol{w}} \\
&= -(\boldsymbol{x} - \hat{\boldsymbol{x}})^\top \boldsymbol{w}^{out} \boldsymbol{x} \\
&= -\boldsymbol{x}^\top \boldsymbol{w}^{out} \boldsymbol{x} + y\boldsymbol{w}^{out\top} \boldsymbol{w}^{out} \boldsymbol{x} \\
&= -y\boldsymbol{x} + y\|\boldsymbol{w}^{out}\|^2 \boldsymbol{x}
\end{aligned}$$

Here we can impose that the weights are normalized, so $\|\boldsymbol{w}^{out}\|^2 = 1$ and the two terms cancel out: $\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0$.

Since we impose that the input and output weights are equal, they need to follow the same update rule to remain equal. Therefore they will both follow the learning rule

$$\Delta \boldsymbol{w}^{in/out} = \eta \left(\boldsymbol{x} y - \boldsymbol{w} y^2 \right) \tag{2}$$

This is truly Oja's rule for the input weights ($\boldsymbol{x}$ and y are the input and output respectively); we know from Week 1 that the corresponding batch update rule is $\Delta \boldsymbol{w} = C\boldsymbol{w} - \left(\boldsymbol{w}^\top C \boldsymbol{w} \right) \boldsymbol{w}$.

(ii) Then turn to a presentation in terms of the vector component. Intepret the result as a Hebbian learning rule and identify the presynaptic and postsynaptic terms. What is the difference between the two rules? How are they related to the Oja rule?

Solution:

We write the update of (2) per component:

$$\Delta w_i = \eta \left(x_i y - w_i y^2 \right)$$

As we said earlier, this is Oja's rule where x_i is the presynaptic term and y is the postsynaptic term.

(iii) Repeat the same calculation but assuming that there are two neurons in the hidden layer. Interpret the resulting online-rule as an interaction between the two hidden neuron. What is this interaction?

Solution:

We have $\hat{x}_k = \sum_i w^{out}_{ki} y_i$, *i.e.* $\hat{\boldsymbol{x}} = W^{out} \boldsymbol{y}$ and $y_i = \sum_j w_{ji} x_j$, *i.e.* $\boldsymbol{y} = W \boldsymbol{x}$. The loss to be minimized is $\mathcal{L} = \frac{1}{2} \mathbb{E} \left[\|x - \hat{x}\|_2^2 \right]$.

$$\frac{\partial \mathcal{L}}{\partial w^{out}_{ki}} = \mathbb{E} \left[\frac{1}{2} \frac{\partial}{\partial w^{out}_{ki}} \sum_j (x_j - \hat{x}_j)^2 \right] = \mathbb{E} \left[- \sum_j (x_j - \hat{x}_j) \frac{\left(\partial \sum_l w^{out}_{jl} y_l \right)}{\partial w^{out}_{ki}} \right] = \mathbb{E} \left[-(x_k - \hat{x}_k) y_i \right]$$

so

$$\frac{\partial \mathcal{L}}{\partial W^{out}} = \mathbb{E} \left[-(\boldsymbol{x} - \hat{\boldsymbol{x}}) \boldsymbol{y}^T \right] = \mathbb{E} \left[-\boldsymbol{x}\boldsymbol{x}^T W^T + W^{out} W \boldsymbol{x}\boldsymbol{x}^T W^T \right] = -CW^T + W^{out} W C W^T$$

and the online version can simply be written

$$\frac{\partial \mathcal{L}}{\partial w^{out}_{ki}} = y_i (-x_k + \boldsymbol{w}_{.k}^\top \boldsymbol{y}) \tag{3}$$

This yields a learning rule like in the one-neuron case (2) except that both hidden neurons appear.

For the input weights W:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w_{ji}} &= - \sum_k (x_k - \hat{x}_k) \sum_{l,m} w^{out}_{kl} \frac{\partial (w_{ml} x_m)}{\partial w_{ji}} \\ &= - \sum_k (x_k - \hat{x}_k) w^{out}_{ki} x_j \\ &= - \sum_k x_k w^{out}_{ki} x_j + \sum_{k,l} w^{out}_{kl} y_l w^{out}_{ki} x_j \end{aligned}$$

Here we identify the expression of y_i in the left term (using that $w_{ki}^{out} = w_{ik}$) and in the second term, since since $l \in \{1, 2\}$ we suppose without loss of generality that $i = 1$. Moreover, we impose again that each line of W is normalized.

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial w_{j1}} &= -\sum_k x_k w_{ik} x_j + \sum_k w_{k2}^{out} y_2 w_{ki}^{out} x_j + \sum_k \left(w_{k1}^{out}\right)^2 y_1 x_j \\ &= -x_j y_1 + x_j y_2 \sum_k w_{k2}^{out} w_{k1}^{out} + x_j y_1 \\ &= x_j y_2 \boldsymbol{w}_{2\cdot}^\top \boldsymbol{w}_{1\cdot}\end{aligned}$$

Again we see that y_2 appears in the expression of the update rule of w_{j1}. With the same reasoning we can develop the result of (3) into the same form; this allows the two weights to remain equal in the two-neuron case too. So for both weights, the update rule is:

$$\Delta w_{1j} = -\eta x_j y_2 \boldsymbol{w}_{2\cdot} \boldsymbol{w}_{1\cdot}^\top \tag{4}$$

We can look into the type of interaction that results from this expression. If both hidden neurons have correlated features, $\boldsymbol{w}_{2\cdot}\boldsymbol{w}_{1\cdot}^\top \geq 0$, and the interaction will be repulsive: w_{1j} will follow an inverse Hebbian rule when y_2 is active, and vice-versa. Therefore this learning rule will drive decorrelation of correlated neurons, and inversely drive increasing alignement of anti-correlated neurons. Overall, this interaction results in decorrelation of the two units.

(iv) Think about the relevance of these results for the interpretation of the learning rule. Would you agree with the following statement: "Hebbian learning rules are able to find 'good representations', in the sense that an optimal reconstruction of the stimulus WOULD be possible (even if we do not implement the reconstruction process)'.

Solution:

The results above show that the Hebbian learning rule actually arises as the solution (gradient descent) when trying to optimize stimulus reconstruction with an auto-encoder. So yes, Hebbian learning rules are intrinsically able to find 'good representations'.

Exercise 2: Some important fun facts about independence and correlation

2.1 Prove that if x_1 and x_2 are statistically independent, they are also uncorrelated:

$$p(x_1, x_2) = p_1(x_1)p_2(x_2) \Rightarrow \langle (x_1 - \langle x_1 \rangle)(x_2 - \langle x_2 \rangle) \rangle = 0.$$

Solution:

The definition of the expectation value[1] of a random variable x with distribution probability $p(x)$ is

$$\langle x \rangle := \int dx\, p(x)\, x,$$

[1] Here we use the notation $\langle \cdot \rangle$ instead of the more formally correct $E\{\cdot\}$ for the expectation value.

so that we can rewrite the correlation term:

$$\begin{aligned}\langle (x_1 - \langle x_1\rangle)(x_2 - \langle x_2\rangle)\rangle &= \int dx_1 \int dx_2\, p(x_1, x_2)\, (x_1 - \langle x_1\rangle)(x_2 - \langle x_2\rangle) \\ &= \int dx_1 \int dx_2\, p(x_1)\, (x_1 - \langle x_1\rangle)\, p(x_2)\, (x_2 - \langle x_2\rangle) \\ &= \left(\int dx_1\, p(x_1)\, (x_1 - \langle x_1\rangle)\right) \left(\int dx_2\, p(x_2)\, (x_2 - \langle x_2\rangle)\right) \\ &= (\langle x_1\rangle - \langle x_1\rangle)(\langle x_2\rangle - \langle x_2\rangle) = 0.\end{aligned}$$

In the last step, we used the fact that $\langle x\rangle$ is independent of x, so that $\int dx\, p(x)\, \langle x\rangle = \langle x\rangle \int dx\, p(x) = \langle x\rangle$ (since $\int dx\, p(x) = 1$).

2.2 Prove: Given two functions h_1, h_2 and two independent random variables x_1 and x_2, the expectation value of the product of h_1 and h_2 factorizes in the product of the expectation values :

$$p(x_1, x_2) = p_1(x_1)p_2(x_2) \;\Rightarrow\; E\{h_1(x_1)h_2(x_2)\} = E\{h_1(x_1)\}E\{h_2(x_2)\}$$

Solution:

Again using the definition of the expectation value:

$$\langle f(x)\rangle := \int dx\, p(x)\, f(x),$$

we can rewrite the expectation value of the product:

$$\begin{aligned}\langle h_1(x_1)h_2(x_2)\rangle &= \int dx_1 \int dx_2\, p(x_1, x_2)\, h_1(x_1)\, h_2(x_2) \\ &= \int dx_1 \int dx_2\, p(x_1)\, h_1(x_1)\, p(x_2)\, h_2(x_2) \\ &= \int dx_1\, p(x_1)\, h_1(x_1) \int dx_2\, p(x_2)\, h_2(x_2) \\ &= \langle h_1(x_1)\rangle \langle h_2(x_2)\rangle.\end{aligned}$$

2.3 For N-dimensional data, the Gaussian distribution has the form:

$$p(\vec{x}) = \frac{1}{\sqrt{(2\pi)^N det(C)}} \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu})^T C^{-1} (\vec{x} - \vec{\mu})\right),$$

where and $\vec{\mu}$ is the mean of the data, C their covariance matrix, and $det(C)$ its determinant.

Suppose that C has elements $C_{ij} \neq 0$ for all i, j indicating that the variables x_i and x_j are correlated. First, convince yourself (without calculation) that after transformation to the coordinate system of Eigenvectors, the new coordinates $\tilde{x}$ are uncorrelated even if the Eigenvalues are not identical $\lambda_n \neq \lambda_m$ for all n, m.

Show with a short calculation (one line) that the variables are not just uncorrelated but also statistically independent. Hence, what you need to show is:

$$\forall i, j : \langle (x_i - \langle x_i\rangle)(x_j - \langle x_j\rangle)\rangle = 0 \;\Rightarrow\; p(\vec{x}) = \prod_i p_i(x_i)\,.$$

Solution:

If all the x_i and x_j variables are decorrelated, the C_{ij} matrix must vanish outside the diagonal. The diagonal elements correspond to the variances σ_i^2, i.e.:

$$C_{ij} = \begin{cases} \sigma_i^2 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

The inverse of such a matrix is a diagonal matrix with elements $1/\sigma_i^2$. Using these two facts, we can rewrite $p(\vec{x})$

$$\begin{aligned} p(\vec{x}) &= \frac{1}{\sqrt{(2\pi)^N det(C)}} \exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu})^T C^{-1}(\vec{x}-\vec{\mu})\right) \\ &= \frac{1}{\prod_i \sqrt{2\pi\sigma_i^2}} \exp\left(-\sum_i \frac{(x_i-\mu_i)^2}{2\sigma_i^2}\right) \\ &= \frac{1}{\prod_i \sqrt{2\pi\sigma_i^2}} \prod_i \exp\left(-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}\right) \\ &= \prod_i \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}\right) = \prod_i p_i(x_i) \end{aligned}$$

where $p(x_i)$ is a 1-dim. gaussian distribution with mean μ_i and variance σ_i^2.

2.4 Now assume that the data is whitened (each component has zero mean and unit variance, and the components are pairwise decorrelated) so that $\lambda_n = \lambda_m$ for all n, m. Show that any rotation $\{\vec{y}^\mu = R\vec{x}^\mu\}$ of the data is also whitened.

Hint: R is a rotation matrix, iff $RR^T = E$, E being the identity matrix.

Solution:

By definition, a whitened data set has: 1) uncorrelated components and 2) variances of all its components equal to 1. This means that its covariance matrix is the identity matrix E:

$$C := \frac{1}{p}\sum_{\mu=1}^{p}(\vec{x}^\mu - \langle\vec{x}\rangle)(\vec{x}^\mu - \langle\vec{x}\rangle)^T = E. \tag{5}$$

We replace $\vec{x}$ by $\vec{y} = R\vec{x}$ and check its covariance matrix C^*:

$$\begin{aligned} C^* &= \frac{1}{p}\sum_\mu (\vec{y}^\mu - \langle\vec{y}\rangle)(\vec{y}^\mu - \langle\vec{y}\rangle)^T \\ &= \frac{1}{p}\sum_\mu (R\vec{x}^\mu - R\langle\vec{x}\rangle)(R\vec{x}^\mu - R\langle\vec{x}\rangle)^T \\ &= \frac{1}{p}\sum_\mu R(\vec{x}^\mu - \langle\vec{x}\rangle)(\vec{x}^\mu - \langle\vec{x}\rangle)^T R^T \\ &= R\underbrace{\left(\frac{1}{p}\sum_\mu (\vec{x}^\mu - \langle\vec{y}\rangle)(\vec{x}^\mu - \langle\vec{x}\rangle)^T\right)}_{=E} R^T \\ &= RER^T = RR^T = E. \end{aligned}$$

Exercise 3: ICA as Hebbian Learning

Consider an ICA algorithm that aims at maximizing $J(\vec{w}) = \langle F(y) \rangle$, where $y = \vec{w}^T \vec{x}$ and $F(y) = \frac{1}{a} \log \cosh(ay)$. The maximization is done by gradient ascent.

3.1 Show that: $\frac{dF}{dy} = \tanh(ay)$.

Solution:

We just derive $F(y)$, remembering that $(f(g(x)))' = f'(g(x))\, g'(x)$, and that $\cosh' = \sinh$:

$$\frac{d}{dy}\left[\frac{1}{a}\log\cosh(ay)\right] = \frac{1}{a} \cdot \frac{1}{\cosh(ay)} \cdot \sinh(ay) \cdot a = \tanh(ay).$$

3.2 Calculate $\frac{dF}{dw_j}$ for $y = \sum w_k x_k$.

Solution:

This time we use the chain rule: since F depends on w_j only through y, we can write

$$\frac{dF}{dw_j} = \frac{dF}{dy} \underbrace{\frac{dy}{dw_j}}_{x_j} = \tanh(ay)\, x_j.$$

3.3 Show that a gradient ascent on $J(\vec{w}) = \langle F(\vec{w}^T \vec{x}) \rangle$ leads to a Hebbian rule.
(Hint: Make the transition from a batch rule to an online rule).

Solution:

The gradient ascent rule suggested above is:

$$\Delta w_j = \eta \frac{d}{dw_j} J(\vec{w}) = \eta \frac{d}{dw_j} \langle F(\vec{w}^T \vec{x}) \rangle = \eta \langle \frac{d}{dw_j} F(\vec{w}^T \vec{x}) \rangle = \frac{\eta}{M} \sum_{\mu=1}^{M} \tanh(ay^\mu)\, x_j^\mu,$$

where $\eta > 0$ is a learning rate. To convert this into a neural Hebbian rule, we make two hypothesis:

a. The learning rate η is small enough, so that we can approximate the current *batch* mode (learning for all samples to be presented and update the weights only then) with an *online* mode (weights are updated at each sample presentation) with learning rate η/M.

b. We define our output neuron rate to be $x_{\text{out}} = \tanh(a\vec{w}^T \vec{x})$.

In that case the online version of the rule above becomes

$$\Delta w_j^\mu = \frac{\eta}{M} x_{\text{out}} x_j^\mu,$$

which is indeed a Hebbian learning rule.

Exercise 4: A few fun facts on Kurtosis

Students who do not want to do the calculation can make use of the statements in 4.1-4.3 as a table of results.

Variance is defined as $var(x) = E\{x^2\} - E\{x\}^2$, kurtosis as $\kappa(x) = E\{x^4\} - 3(E\{x^2\})^2$.

For each of the following distributions, calculate the variance and prove the given kurtosis:

4.1 the Gaussian distribution, with kurtosis $\kappa = 0$:

$$p(x) = \sqrt{\frac{a}{\pi}} \exp\left(-ax^2\right).$$

Hint: $x^2 \exp(-ax^2) = -\frac{d}{da}\exp(-ax^2)$.

Solution:

Let's compute the first, second and fourth moments of the Gaussian distribution. The mean is immediately solved by noticing the symmetry in the distribution:

$$\langle x \rangle = \sqrt{\frac{a}{\pi}} \int_{-\infty}^{\infty} dx\, x\, exp(-ax^2) = \sqrt{\frac{a}{\pi}} \left(\int_{-\infty}^{0} dx\, x\, exp(-ax^2) + \int_{0}^{\infty} dx\, x\, exp(-ax^2) \right)$$
$$= \sqrt{\frac{a}{\pi}} \left(\int_{0}^{\infty} dx\, x\, exp(-ax^2) - \int_{0}^{\infty} dx\, x\, exp(-ax^2) \right) = 0$$

Note that the same applies for all odd moments of any symmetric distribution.

For the second moment, we use the hint:

$$\langle x^2 \rangle = \sqrt{\frac{a}{\pi}} \int_{-\infty}^{\infty} dx\, x^2\, exp(-ax^2) = -\sqrt{\frac{a}{\pi}} \frac{d}{da} \underbrace{\int_{-\infty}^{\infty} dx\, exp(-ax^2)}_{=\sqrt{\frac{\pi}{a}}} = -\sqrt{\frac{a}{\pi}} \left(-\frac{1}{2}\sqrt{\frac{\pi}{a^3}} \right) = \frac{1}{2a}$$

For the fourth moment, we just use the same trick twice:

$$\langle x^4 \rangle = \sqrt{\frac{a}{\pi}} \int_{-\infty}^{\infty} dx\, x^4\, exp(-ax^2) = -\sqrt{\frac{a}{\pi}} \frac{d}{da} \int_{-\infty}^{\infty} dx\, x^2\, exp(-ax^2)$$
$$= \sqrt{\frac{a}{\pi}} \frac{d^2}{da^2} \underbrace{\int_{\infty}^{\infty} dx\, exp(-\frac{ax^2}{2})}_{=\sqrt{\frac{\pi}{a}}} = \sqrt{\frac{a}{\pi}} \left(\frac{3}{4}\sqrt{\frac{\pi}{a^5}} \right) = \frac{3}{4a^2}$$

Using these results one can compute the variance $var(x) = \frac{1}{2a}$, and the kurtosis $\kappa(x) = \frac{3}{4a^2} - 3\left(\frac{1}{2a}\right)^2 = 0$.

4.2 the uniform distribution, with kurtosis $\kappa = -\frac{6}{5}$:

$$p(x) = \begin{cases} \frac{1}{2\sqrt{3}} & \text{if } |x| \le \sqrt{3} \\ 0 & \text{otherwise.} \end{cases}$$

Solution:

Again, we compute the moments we need to compute the variance and the kurtosis. The first moment is again 0 by symmetry. The second moment is

$$\langle x^2 \rangle = \int_{-\sqrt{3}}^{\sqrt{3}} dx\, \frac{1}{2\sqrt{3}}\, x^2 = \frac{1}{2\sqrt{3}} \left[\frac{x^3}{3} \right]_{-\sqrt{3}}^{\sqrt{3}} = 1\,,$$

and the fourth is

$$\langle x^4 \rangle = \int_{-\sqrt{3}}^{\sqrt{3}} dx \, \frac{1}{2\sqrt{3}} \, x^4 = \frac{1}{2\sqrt{3}} \left[\frac{x^5}{5} \right]_{-\sqrt{3}}^{\sqrt{3}} = \frac{9}{5}.$$

Thus the variance is $var(x) = 1$, and the kurtosis is $\kappa(x) = \frac{9}{5} - 3 = -\frac{6}{5}$.

4.3 the exponential distribution (Laplace distribution), with kurtosis $\kappa = 3$:

$$p(x) = \frac{1}{\sqrt{2}} \exp(-\sqrt{2}|x|).$$

Solution:

Here, because the distribution is symmetric, the first moment vanishes again. To compute the second and fourth moments, it is easiest to use the same trick as for the Gaussian, i.e. to notice that

$$-\frac{d}{da} \int dx \, x^{n-1} \, \exp(-ax) = \int dx \, x^n \, \exp(-ax)$$

and that

$$\int dx \, x^0 \, \exp(-ax) = \frac{1}{a}.$$

It is also nice to get rid of the absolute value. For any symmetric functions ($f(x) = f(-x)$) one can write

$$\int_{-\infty}^{\infty} f(x)dx = \int_{-\infty}^{0} f(x)dx + \int_{0}^{\infty} f(x)dx = \int_{0}^{\infty} f(-x)dx + \int_{0}^{\infty} f(x)dx = 2 \int_{0}^{\infty} f(x)dx \, .$$

The terms we have to integrate for even moments of this distribution are symmetric. Computing the second moment then reduces to

$$\begin{aligned} \langle x^2 \rangle &= \frac{2}{\sqrt{2}} \int_0^{\infty} dx \, x^2 \, \exp(-\sqrt{2}x) = \sqrt{2} \left[\frac{d^2}{da^2} \int_0^{\infty} dx \, \exp(-ax) \right]_{a=\sqrt{2}} \\ &= \sqrt{2} \left[\frac{2}{a^3} \right]_{a=\sqrt{2}} = 1 \, , \end{aligned}$$

and the fourth

$$\begin{aligned} \langle x^4 \rangle &= \frac{2}{\sqrt{2}} \int_0^{\infty} dx \, x^4 \, \exp(-\sqrt{2}x) = \sqrt{2} \left[\frac{d^4}{da^4} \int_0^{\infty} dx \, \exp(-ax) \right]_{a=\sqrt{2}} \\ &= \sqrt{2} \left[\frac{24}{a^5} \right]_{a=\sqrt{2}} = 6 \, . \end{aligned}$$

The variance is $var(x) = 1$, and the kurtosis is $\kappa(x) = 6 - 3 = 3$.

4.4 In the above examples, do distributions with 'longer tails' than the Gaussian yield smaller or larger kurtosis? Do you think that this observation about 'tails' and kurtosis can be transformed into a general statement?

Solution:

The uniform distribution has no tail, since the probability density function (pdf) is 0 for $|x| > \sqrt{3}$, and hence shorter tails than the Gaussian distribution. Correspondingly, the (excess) kurtosis is negative $\kappa(x) < 0$. On the contrary, for the Laplace distribution, the pdf decays proportionally to $e^{-|x|}$ in contrast to e^{-x^2} for the Gaussian, and hence heavier tails. Its associated kurtosis is positive $\kappa(x) > 0$.

This in fact can be generalized, distributions with shorter tails (faster decay of the pdf) than the Gaussian distribution will have negative kurtosis, and those with longer tails will have positive kurtosis.

This can be shown for centered distribution X by remarking that,

$$sign(\mathbb{E}[X^4] - 3\mathbb{E}[X^2]^2) = sign(\frac{\mathbb{E}[X^4]}{\mathbb{E}[X^2]^2} - 3) = sign(\mathbb{E}[(\frac{X}{\sigma})^4] - 3),$$

where σ is the standard deviation of X. Hence, a positive (respectively negative) (excess) kurtosis only occurs if the fourth standardized moment of X, $E[(\frac{X}{\sigma})^4] = \int_{\mathbb{R}} x^4 p(x) dx$ is larger (resp. smaller) than the fourth standardized moment of the Gaussian which is equal to 3. The fourth moment for standardized random variables is dominated by large-valued outliers and longer tails. Indeed any value comprised within the first standard deviation will be close to 0 in the integral due to the fourth power. On the contrary, large (absolute) values of X which have more (resp. less) mass due to a slower (resp. faster) decay of the pdf will increase (resp. decrease) the fourth standardized moment in comparison to that of the Gaussian.

Exercise 5: Kurtosis maximization

Remember that the kurtosis is defined as $\kappa(x) = E[x^4] - 3E[x^2]^2$. Suppose that we have two independent variables x_1 and x_2 both of zero mean, but we measure some arbitrary mixture. In class we have argued in a hand-waving fashion that Kurtosis is maximal (or sometimes minimal) if the direction of projection yields one of the independent variables. In this exercise we have a special case, where we can explicitly show this. We mix two variables of known kurtosis, and then apply a projection in an arbitrary direction which gives a variable y. For this mixed variable y we maximize kurtosis. The calculation is a bit lengthy, but for those of you who have doubts why ICA works, it may provide useful insights. Here are the steps of the calculation:

5.1 Show that the kurtosis of $y = x_1 + x_2$ is given by $\kappa(y) = \kappa(x_1) + \kappa(x_2)$.

Solution:

Let's calculate the second and fourth moments of y:

$$\begin{aligned} E[y^2] &= E[(x_1 + x_2)^2] = E[x_1^2 + 2x_1x_2 + x_2^2] = E[x_1^2] + E[2x_1x_2] + E[x_2^2] \\ &= E[x_1^2] + 2\underbrace{E[x_1]}_{0}\underbrace{E[x_2]}_{0} + E[x_2^2] = E[x_1^2] + E[x_2^2] \end{aligned}$$

and

$$\begin{aligned} E[y^4] &= E[(x_1 + x_2)^4] = E[x_1^4 + 4x_1^3x_2 + 6x_1^2x_2^2 + 4x_1x_2^3 + x_2^4] \\ &= E[x_1^4] + 4E[x_1^3]\underbrace{E[x_2]}_{0} + 6E[x_1^2]E[x_2^2] + 4\underbrace{E[x_1]}_{0}E[x_2^3] + E[x_2^4] \\ &= E[x_1^4] + 6E[x_1^2]E[x_2^2] + E[x_2^4]. \end{aligned}$$

Thus the kurtosis of y is:

$$\begin{aligned}\kappa(y) &= E[y^4] - 3E[y^2]^2 = E[x_1^4] + 6E[x_1^2]E[x_2^2] + E[x_2^4] - 3\left(E[x_1^2] + E[x_2^2]\right)^2 \\ &= \underbrace{E[x_1^4] - 3E[x_1^2]^2}_{\kappa(x_1)} + \underbrace{6E[x_1^2]E[x_2^2] - 6E[x_1^2]E[x_2^2]}_{0} + \underbrace{E[x_2^4] - 3E[x_2^2]^2}_{\kappa(x_2)} \\ &= \kappa(x_1) + \kappa(x_2)\end{aligned}$$

5.2 Show that the kurtosis of $y = \alpha x$ is given by $\kappa(y) = \alpha^4 \kappa(x)$.

Solution:

We simply calculate:

$$\kappa(y) = E[\alpha^4 x^4] - 3E[\alpha^2 x^2]^2 = \alpha^4 (E[x^4] - 3E[x^2]^2) = \alpha^4 \kappa(x)$$

5.3 Use 1 and 2 to show that the kurtosis of $y = \sqrt{a}x_1 + \sqrt{1-a}x_2$, $a \in [0,1]$, is given by

$$\kappa(y) = a^2\kappa(x_1) + (1-a)^2\kappa(x_2)\,.$$

Solution:

Thanks to 1 and 2 above, this is simply:

$$\kappa(y) = \kappa(\sqrt{a}x_1 + \sqrt{1-a}x_2) = \kappa(\sqrt{a}x_1) + \kappa(\sqrt{1-a}x_2) = a^2\,\kappa(x_1) + (1-a)^2\,\kappa(x_2).$$

5.4 Let $\kappa(x_1) = c$ and $\kappa(x_2) = d$ be the kurtosis of x_1 and x_2. Assume that both signals are super-Gaussian and that $0 < c < d$. Show that the kurtosis of the mixture $y = \sqrt{a}x_1 + \sqrt{1-a}x_2$ has maxima for $a = 0$ and $a = 1$, and that $a = 0$ is the global maximum.

Solution:

Since we want to find maxima and minima, let's compute the derivative $d\kappa(y)/da$:

$$\frac{d\kappa(y)}{da} = \frac{d}{da}\left[a^2\,\kappa(x_1) + (1-a)^2\,\kappa(x_2)\right] = 2a\kappa(x_1) + 2(a-1)\kappa(x_2) = 2ac + 2(a-1)d$$

This is 0 only if $a = \frac{d}{c+d}$. To find whether this corresponds to a minimum or a maximum, we compute the second derivative

$$\frac{d^2\kappa(y)}{da^2} = \frac{d}{da}\left[2a\kappa(x_1) + 2(a-1)\kappa(x_2)\right] = 2\kappa(x_1) + 2\kappa(x_2) = 2c + 2d.$$

This positive for all allowed values of a, so the point $a^* = \frac{d}{c+d}$ is a minimum. Thus the maximum happen at the boundaries of the domain, $a = 0$ and $a = 1$. Since $\kappa(x_2) > \kappa(x_1)$, the global maximum happens for $a = 0$, $y = \kappa(x_2)$, see Figure 2.

This means that for linear, whitened mixtures of super-gaussian signals, maximizing the kurtosis leads to the recovery of one of the two signal. Using an analytical method, as we did, we are guaranteed to find the global minimum, but usually one would use a numerical gradient ascent method on the data. Even then, the graph shows clearly that one would recover one of the two statistically independent signals, although not necessarily the one with highest kurtosis.

5.5 Which value(s) of a maximize the kurtosis if the signals x_1 and x_2 are sub-Gaussian: $c < d < 0$?

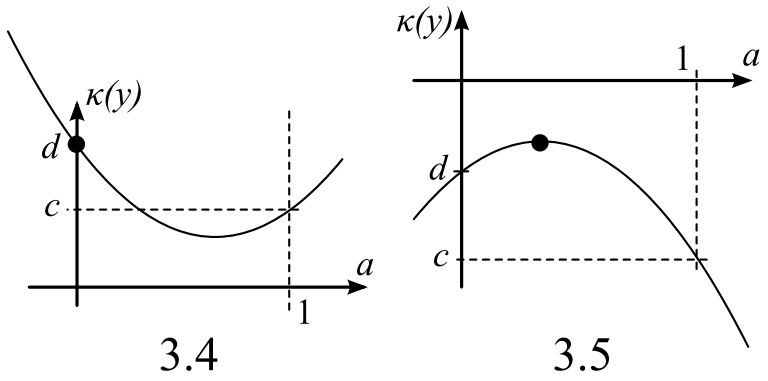

Figure 2: The kurtosis of the mixture y as a function of a, in questions 4 and 5.

Solution:

Similarly as above, there is a fixed point in $a^* = \frac{d}{c+d}$, and the second derivative is $\frac{d^2\kappa(y)}{da^2} = 2\kappa(x_1) + 2\kappa(x_2)$. Since this time the second derivative is negative, this point is a local (and global) maximum, see Figure 1.

In contrast to the situation of question 4, maximizing the kurtosis for a mixture of sub-gaussian signals is a bad idea, since it finds the linear combination that is most gaussian! In that case, the correct approach would be to *minimize* the kurtosis.