2007

TR-2007007: Independent Component Analysis: An Introduction

Rave Harpaz

Independent Component Analysis: An Introduction

Rave Harpaz
Pattern Recognition Laboratory
The Graduate Center, City University of New York,
365 Fifth Avenue New York, NY 10016, USA

Nov. 15 2005

Abstract

Independent Component Analysis (ICA) can be described in several ways, one of which is as a technique that seeks to find a set directions (components) underlying multivariate data that are most independent of one another. While there are several ICA models and many ICA methods, in this report we focus on the most basic model and one of the most popular and simple algorithms; the One-Unit FastICA algorithm. ICA is based on several very interesting results in probability, statistics, information theory, and non-linear optimization theory. Most of the introductory publications on this topic leave these results unattended. The aim of this report is to fill this gap. Throughout this report each of these results, including its proof, is introduced in accordance with the ICA subproblem it attempts solve or underlying principle it attempts to explain.

1 Motivation

While principle component analysis seek directions in the feature space that best represent the data in a sum of squared error sense, independent component analysis (ICA) instead seeks directions that that are most independent from each other. ICA can best be understood by looking at two of its main applications.

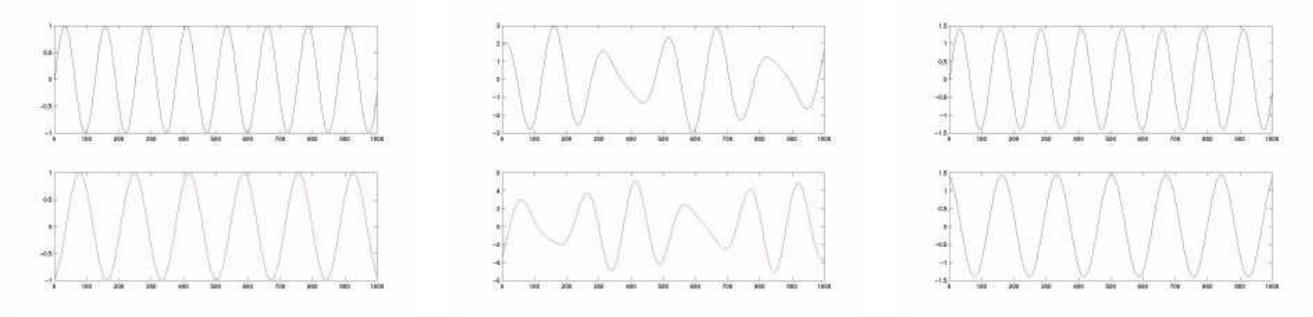
1.1 The blind Source Separation Problem (BSS)

Suppose $d \geq 2$ signals $s_1(t), s_2(t), \ldots, s_d(t)$ which are assumed to be independent, and were $1 \leq t \leq n$ is a time index (t can also be thought of the number of observations), are linearly mixed to yield at the receiver's end $x_1(t), x_2(t), \ldots, x_d(t)$, i.e.

$$\begin{aligned} x_1 &= a_{11}s_1 + a_{12}s_2 + \ldots + a_{1d} \\ x_2 &= a_{21}s_1 + a_{22}s_2 + \ldots + a_{2d} \\ &\vdots \\ x_d &= a_{d1}s_1 + a_{d2}s_2 + \ldots + a_{dd} \end{aligned}$$

were the a_{ij}'s are the mixing weights. Note that for the sake of simplicity the number of recorded signals $x_i(t)$ is equal to the number of assumed signals $s_i(t)$ which is d, however this does not need to be case. Given merely the sensed or recorded signals $x_i(t)$ and an assumed number of signals d the goal is to recover the $s_i(t)'s$, which implicitly implies that the weights a_{ij} need to be estimated. An illustration of this problem is

depicted in fig. 1. Two signals (fig. 1a) $s_1(t), s_2(t)$ are linearly mixed to yield (fig. 1b) $x_1(t) = s_1(t) - 2s_2(t)$ and $x_2(t) = 1.73s_1(t) + 3.41s_2(t)$ at the receiver's end. We then input the mixed signal into an ICA algorithm (fig. 1c), which is able to recover the original signals (but not their amplitude).



a b c

Figure 1: An illustration of blind source separation

1.2 Feature Selection for Classification

It is well accepted that PCA is not always best for pattern recognition from the class separation point of view. In many cases projecting the data onto the leading principle components (eigenvectors with largest eigenvalues) will cause two classes to coincide. In contrast ICA can achieve much more than simple decorrelation of the data required by PCA. Searching for independence is a stronger condition then uncorrelatedness, which are equivalent only for Gaussian variables. Searching for independent features gives us the means of exploiting a lot more information hidden in higher order statistics. Constraining the search by mining information in second-order statistics only results in the least interesting projection directions from the class separation point of view, as illustrated in fig. 2. The vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ obtained by PCA point in the principle and minor axis directions respectively. Projecting the data onto the direction of principle component, will result in a variable with a pdf close to a Gaussian, and thus cause the two class to coincide. However the most interesting or appropriate direction from the class separation point of view is the direction of the minor axis. Projecting the data onto this axis will result in a variable whose pdf deviates substantially from the Gaussian. ICA can unveil from higher order statistics of the data the piece of information that points $\mathbf{a}_2$ as the most interesting direction.

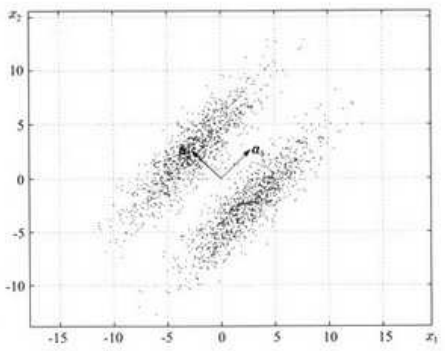


Figure 2: ICA from the class separation point of view

2 The ICA Model

Assume we observe d linear mixtures $x_1, x_2, \ldots, x_d$ of d independent components $s_1, s_2, \ldots, s_d$, such that

$$x_i = a_{i1}s_1 + a_{i2}s_2 + \cdots + a_{id}s_d, \qquad \text{for all } i$$

We drop the time index t, and assume that each mixture x_i as well as each independent component s_j are random variables instead of time signals. Without loss of generality, we assume that both the mixture and independent components have zero mean. If not, then the observed variables x_i can always be centered by subtracting the sample mean which will also make the independent components centered. Further more, for the sake of simplicity we use vector-matrix notation, were $\mathbf{x}$ is a random vector whose elements are $x_1, \ldots, x_d$, likewise $\mathbf{s}$ a random vector whose elements are $s_1, \ldots, s_d$, and A the matrix with elements a_{ij}. Using this notation the ICA model is written as

$$\mathbf{x} = A\mathbf{s} \tag{1}$$

The ICA model describes how the observed data are generated by a process of mixing the independent components s_j. The independent components are latent variables, which means they cannot be directly observed. Also the matrix A called the mixing matrix is assumed to be unknown. All we observe is $\mathbf{x}$ from which $\mathbf{s}$ and A must be estimated. The most important assumption about this model is that the components s_j are statistically independent. We must also assume that the independent components must have nongaussian distributions, however their distributions are unknown. For simplicity, we also assume that A is square. After, estimating A, we can compute its inverse $W = A^{-1}$ called the unmixing or demixing matrix, and use it to obtain the independent components by

$$\mathbf{s} = W\mathbf{x} \tag{2}$$

The model can be extended to include noise in the observations, which will mean adding a noise term to the model, and to relax the requirement that A be square. However the discussion of these extensions are beyond the scope of this report.

2.1 Identifiability Conditions for the ICA Model

1. All independent components, with the possible exception of one, must be nongaussian. Thus, ICA is meaningful only if the involved random variables are nongaussian. For Gaussian random variables independence is equivalent to uncorrelatedness (see appendix 7.2 for a proof), and therefore any decorrelating representation, such as the one obtained by PCA will yield independent components. Mathematically, one can show that any orthogonal transformation of Gaussian random variables will have the exact same distribution as of the original variables, and furthermore, if the original variables were independent so will the transformed variables (see appendix 7.2 for a proof). Thus, in the case of gaussian random variables the ICA model can be estimated only up to an orthogonal tansformation, or in other words the mixing matrix A is unidentifiable.

2. The number of observed linear mixtures must be at least as large as the number of independent components, and the mixing matrix A must be of full column rank. In the following we will assume that the number of observed mixtures and independent components is the same. This can be justified by the fact that if the number of observed mixtures is larger than the independent components, then the dimension of the observed vectors can be reduced by methods such as PCA.

2.2 Ambiguities in the ICA Model

1. The variances of the independent components cannot be estimated. The reason is that since $\mathbf{s}$ and A are unknown, any scalar multiplier of one of the independent components s_i in eq. 1 can always be canceled

by dividing the corresponding column a_i of A by the same scaler. As a consequence, for mathematical convenience the independent components are assumed to have unit variance. This restriction is then taken into account by the ICA solution of the mixing matrix A. However the cancelation of the scaler still leaves ambiguity about the sign of the independent component, as it can always be multiplied by -1. This in turn makes the independent components unique up to a multiplicative sign, which is an insignificant indeterminacy.

2. The order of the independent components cannot be determined. This is again, since both $\mathbf{s}$ and A are unknown we can freely change the order of the independent components and rename s_i with s_j without effecting the model. This is in contrast to PCA, where specific ordering is associated with with the values of the corresponding eigenvalues. However, in practice it is possible to introduce some form of ordering on the independent components. The most common ordering, which comes in handy for classification purposes (from the class separation point of view), is to order the components according to their degree of non-gaussianity, measured by an appropriate index which will be covered later. The rational is that a Gaussian pdf is the most "random" (from the entropy perspective) among a all pdf's of a given mean and variance (this result is proven in appendix 7.3). From this point of view the Gaussian is the least "interesting" or least informative with respect to the underlying structure of the data. In contrast, distributions that have the least "resemblance" to the Gaussian are more interesting since they display some structure associated with the data.

3 The Underlying Idea of ICA Estimation

The key to the estimation of the ICA model is the exploitation of nongaussianity through the Central Limit Theorem, which states that the distribution of a sum of independent random variables tends towards a Gaussian distribution. Thus, the sum of two or more independent random variables usually has a distribution that is closer to the gaussian than any of its individual random variables.

Assume that a data vector $\mathbf{x}$ follows the ICA model, i.e. $\mathbf{x} = A\mathbf{s}$. To estimate one of the independent components, call it y, were from eq. 2 $y = \mathbf{w}'\mathbf{x}$, we need to estimate the vector $\mathbf{w}$. If $\mathbf{w}$ were indeed one of the rows of W from eq. 2 or equivalently one of the rows of A^{-1} then $\mathbf{w}'\mathbf{x}$ would actually equal one of the independent components. Using the central limit theorem we can estimate $\mathbf{w}$.

Suppose we let

$$\mathbf{z} = A'\mathbf{w}$$

then

$$y = \mathbf{w}'\mathbf{x} = \mathbf{w}'A\mathbf{s} = \mathbf{z}'\mathbf{s}$$

thus, y is a linear combination of the independent components s_i with weights z_i. Since the sum of a linear combination of the independent random variables s_i is more gaussian than any individual s_i, $\mathbf{z}'\mathbf{s}$ is more gaussian than any s_i and becomes least gaussian when it equals one of the s_i's, in which case only one of the elements of $\mathbf{z}$ is nonzero and equals one. Note that if $\mathbf{w}$ indeed equals one of the rows of A^{-1} then $\mathbf{z} = A'\mathbf{w} = e_i$, were e_i is one of the columns (or rows) of I. This is because $A'(A^{-1})' = I$. Therefore we can estimate w by maximizing the nongaussianity of $\mathbf{w}'\mathbf{x} = \mathbf{z}'\mathbf{s}$. Such a vector will necessarily correspond (after the transformation $\mathbf{z} = A'\mathbf{w}$) to a $\mathbf{z}$ which has only one nonzero component.

4 Preprocessing for ICA

In order to simplify an ICA algorithm it is useful to preprocess the data, i.e. the observations $\mathbf{x}$.

4.1 Centering the Data

As a first step it is necessary to center $\mathbf{x}$, i.e. to create new observations $\tilde{\mathbf{x}}$ such that $E[\tilde{\mathbf{x}}] = 0$. This is done by subtracting $E[\mathbf{x}]$ from each observation. As a result $E[\tilde{\mathbf{s}}] = 0$, this is because $E[\tilde{\mathbf{s}}] = E[W\tilde{\mathbf{x}}] = WE[\tilde{\mathbf{x}}] = 0$. After estimating A with the centered data, we can complete the estimation of $\mathbf{s}$ by adding $E[\mathbf{s}] = WE[\mathbf{x}]$ to the centered estimates $\tilde{\mathbf{s}}$.

4.2 Whitening/Sphering the Data

Assuming that $\mathbf{x}$ is already centered, another useful preprocessing step is to transform $\mathbf{x}$ to a new vector $\tilde{\mathbf{x}}$ which is white, i.e. its making its components uncorrelated with unit variance, or equivalently the covariance matrix of $\tilde{\mathbf{x}}$, call it C is the identity

$$C = E[\tilde{\mathbf{x}}\tilde{\mathbf{x}}'] = I$$

One way of whitening the data is to use the eigen-decomposition of the covariance matrix C. Let $C = VDV'$ where V is the orthonormal matrix of eigenvectors and D is a diagonal matrix of eigenvalues. Then whitening can be obtained by the following transformation

$$\tilde{\mathbf{x}} = VD^{-1/2}V'\mathbf{x}$$

It easy to see why the transformed vectors $\tilde{\mathbf{x}}$ are now whitened.

$$E[\tilde{\mathbf{x}}\tilde{\mathbf{x}}'] = E[VD^{-1/2}V'\mathbf{x}\mathbf{x}'VD^{-1/2}V']$$

$$= E[VD^{-1/2}V'VDV'VD^{-1/2}V'] = VD^{-1/2}DD^{-1/2}V' = VIV' = VV' = I$$

Another way to whiten the data is by using the following transformation

$$\tilde{\mathbf{x}} = C^{-1/2}\mathbf{x}$$

so that

$$E[\tilde{\mathbf{x}}\tilde{\mathbf{x}}'] = E[C^{-1/2}\mathbf{x}\mathbf{x}'(C^{-1/2})']$$

$$= C^{-1/2}E[\mathbf{x}\mathbf{x}']C^{-1/2} = C^{-1/2}CC^{-1/2} = I$$

Whitening transforms the original mixing matrix A into a new one $\tilde{A}$. If let $\tilde{A} = VD^{-1/2}V'A$, since $\tilde{\mathbf{x}} = VD^{-1/2}V'\mathbf{x}$ and $\mathbf{x} = A\mathbf{s}$ we have

$$\tilde{\mathbf{x}} = VD^{-1/2}V'\mathbf{x} = VD^{-1/2}V'A\mathbf{s} = \tilde{A}\mathbf{s}$$

The main advantage of whitening is that it reduces the number of parameters (elements of the mixing matrix) that need to be estimated. To see that note that the new mixing matrix $\tilde{A}$ is orthonormal. Since we have assumed that $E[\mathbf{s}\mathbf{s}'] = I$

$$I = E[\tilde{\mathbf{x}}\tilde{\mathbf{x}}'] = E[\tilde{A}\mathbf{s}\mathbf{s}'\tilde{A}'] = \tilde{A}E[\mathbf{s}\mathbf{s}']\tilde{A}' = \tilde{A}\tilde{A}'$$

Now, instead of estimating the d^2 elements of the mixing matrix we need to estimate the d^2 elements, but subject to $d + \binom{d}{2}$ orthogonality constraints. Thus, the new mixing matrix $\tilde{A}$ has $d^2 - d - \binom{d}{2} = \binom{d}{2}$ degrees of freedom, reducing by nearly half the number of parameters that need to be estimated.

In the following we assume that the data has already been preprocessed by centering and whitening, and will denote by $\mathbf{x}$ and A the preprocessed data and mixing matrix.

5 Measures of Nongaussianity

Assume that we want to measure the nongaussianity of the random variable $y = \mathbf{w}'\mathbf{x}$, given that $E[y] = 0$ and $var[y] = 1$.

5.1 Kurtosis

The classical measure of of nongaussianity is the forth-order cumulant known as kurtosis (see appendix 7.1), which is defined as

$$kurt(y) = k_4(y) = E[(y - E[y])^4] - 3E[(y - E[y])^2]^2 = E[y^4] - 3\sigma^4$$

For a gaussian variable y the forth moment equals $3E[[(y - E[y])^2]^2$, thus the kurtosis equals zero which explains why gaussian independent components cannot be estimated using this method.

To illustrate how independent components can be found by maximizing or minimizing kurtosis we will assume that each independent component s_j has kurtosis $kurt(s_j)$. From eq. 2 to find one independent component we would like estimate $\mathbf{w}$ by maximizing or minimizing the kurtosis of $y = \mathbf{w}'\mathbf{x}$. This will be meaningful only if we bound the norm of $\mathbf{w}$. So lets assume that $\|\mathbf{w}\| = 1$. Using the the orthonormal mixing matrix A we define again $\mathbf{z} = A'\mathbf{w}$. Using the properties of kurtosis

$$kurt(\mathbf{w}'\mathbf{x}) = kurt(\mathbf{w}'A\mathbf{s}) = kurt(\mathbf{z}'\mathbf{s}) = \sum_{i=1}^{d} z_i^4 kurt(s_i)$$

also note that

$$\|\mathbf{z}\| = \mathbf{z}'\mathbf{z} = \mathbf{w}'AA'\mathbf{w} = \mathbf{w}'\mathbf{w} = 1$$

It can be shown that the maxima or minima of $kurt(\mathbf{w}'\mathbf{x})$ under the constraints $\|\mathbf{w}\| = \|\mathbf{z}\| = 1$ occurs when $\mathbf{z} = \pm e_j$ where e_j is one of the columns of I. Therefore $\mathbf{w} = A\mathbf{z} = Ae_j = a_j$ $(\pm a_j)$, i.e. one of the columns of the orthonormal mixing matrix A. So by maximizing or minimizing the kurtosis of $\mathbf{w}'\mathbf{x}$ under the given constraints, the columns of the mixing matrix are obtained as solutions for $\mathbf{w}$.

In practice we would use a gradient method (appendix 7.7) or one of its extensions to find the maxima or minima of $kurt(\mathbf{w}'\mathbf{x})$. However when the kurtosis needs to be estimated from a sample its value may be very sensitive to outliers. In other words, it is not a robust measure of nongaussianity, thus other more robust measures of nongaussianity need to be employed.

5.2 Negentropy

Another very useful measure of nongaussianity is is given by negentropy. The entropy (see appendix 7.3 for the relevant mathematics of information theory) H for a discrete random variable X is defined by

$$H = -\sum_x P(x) log P(x)$$

were $P(x)$ is the probability that $X = x$ and $H = 0$ when $P(x) = 0$. The generalization of entropy to a continuous random variable often called differential entropy, or to a random vector $\mathbf{x}$ is defined by

$$H = -\int_{-\infty}^{\infty} f(\mathbf{x}) log f(\mathbf{x}) d\mathbf{x}$$

were $f(\mathbf{x})$ is the density function of the random vector $\mathbf{x}$. A fundamental result in information theory is that a gaussian variable has the largest entropy among all random variables of equal variance (proof in appendix 7.3). This means that entropy can be used as a measure of nongaussianity. To obtain a measure of nongaussianity that is zero for a gaussian random variable and always nonnegative we can use negentropy J defined as follows

$$J(\mathbf{x}) = H(\mathbf{x}_{gauss}) - H(\mathbf{x})$$

were $\mathbf{x}_{gauss}$ is a Gaussian random variable of with the same covariance matrix Σ as $\mathbf{x}$. The main advantage of using negentropy is that from the statistical point of view it is the optimal estimator of nongaussianity. However the problem in using negentropy is that it would require an estimate of a pdf which makes it from the computational point of view only a theoretical measure of nongaussianity. This conflict lead to the development of approximations to negentropy.

5.2.1 Approximations of Negentropy

The classical approximation to negentropy is through the use of higher order moments and cumulants. One example is the following (see survey, the first term needs to be replaced with the 3rd cumulant)

$$J(x) = \frac{1}{12}E[x^3]^2 + \frac{1}{48}kurt(x)^2 \tag{3}$$

were x is a random variable assumed to be of zero mean and unit variance. However this approximation suffers from the same nonrobustness issues involved in using kurtosis. To avoid this problem a family of approximations based on the maximum-entropy principle were developed. The general approximation is as follows:

$$J(x) = [E[G(x)] - E[G(v)]]^2 \tag{4}$$

were x is assumed to be of zero mean and unit variance, v is a Gaussian random variable of zero mean and unit variance, and G is some nonquadratic function. One can see that this approximation is always non-negative and equal to zero when x has a Gaussian distribution, making it consistent with measures of nongaussianity. One can check that this is a generalization of the approximation given in eq. 3 (might be a mistake and should be a generalization of kurtosis, see survey and the paper one unit contrast function). For example by taking $G(x) = x^4$, one should obtain the approximation given in eq. 3 (should be the square of kurtosis since $E[v^4] = 3$ for 0,1 gaussian). But by choosing G wisely one can obtain better approximations than the one given in eq. 3. In particular choosing a G that does not grow too fast, one obtains more robust approximations to negentropy. The following choices of G have proven useful.

$$G_1(x) = \frac{1}{c}\log\cosh cx \qquad G_1(x) = -\exp\left(-\frac{x^2}{2}\right)$$

were c is some constant such that $1 \leq c \leq 2$. This family of approximations to negentropy present a good compromise between the properties of the two classical measures of nongaussianity given by kurtosis and negentropy.

6 The One-Unit FastICA Algorithm

FastICA is the most popular algorithm for finding independent components. The one-unit FastICA algorithm finds one independent components or one direction $\mathbf{w}$ of the independent components at a time. It does so by maximizing the nongaussianity of $\mathbf{w}'\mathbf{x}$, were nongaussianity is measured by means of the approximation to negentropy as given in eq. 4. To find the maxima of the nongaussianity of $\mathbf{w}'\mathbf{x}$ FastICA uses a Newton Iteration scheme (appendix 7.8).

One-Unit FastICA

1. Choose an initial (random) vector $\mathbf{w}_0$, let $k = 0$

2. $\mathbf{w}_{k+1} = E[\mathbf{x}g(\mathbf{w}_k'\mathbf{x})] - E[g'(\mathbf{w}_k'\mathbf{x})]\mathbf{w}_k$

3. $\mathbf{w}_{k+1} = \mathbf{w}_{k+1} / \|\mathbf{w}_{k+1}\|$

4. if $|\mathbf{w}_{k+1}'\mathbf{w}_k| \geq (1-\epsilon)$ output $\mathbf{w}_{k+1}$, else let $k = k+1$ and goto step 2

The last step (4) checks for convergence, i.e. that the new and old $\mathbf{w}$ point in the same direction, or in other words their dot-product is very close to 1. The absolute value of the dot-product is used because $\mathbf{w}$ and $-\mathbf{w}$ define the same direction, i.e. it is not necessary that $\mathbf{w}$ converges to a single point, supporting what was stated earlier that independent components can only be defined up to a multiplicative sign. The final vector $\mathbf{w}_{k+1}$ output by the algorithm equals one of the columns of the orthonormal mixing matrix $\tilde{A}$. Also note that g denotes the derivative of the nonquadratic function G used in eq. 4, and similarly g' denotes its second derivative. In addition the expectations in the FastICA algorithm must be estimated using the data, ideally all the data should be used.

6.1 Estimating Several Independent Components

To estimate several independent components we will need to run FastICA several times. However to ensure that we estimate each time a different independent component, i.e., to ensure that the vectors do not converge to the same point we need to orthogonalize the vectors $\mathbf{w}$ at each iteration. This is achieved by projecting the current solution $\mathbf{w}_k$ to the space orthogonal to the columns of the mixing matrix that were previously estimated. Assuming we have already estimated n independent components or n vectors which are the columns of the incomplete $d \times n$ mixing matrix A, then step 3 of the algorithm becomes

$$3.\ \mathbf{w}_{k+1} = (I - AA')\mathbf{w}_{k+1}, \text{ and } \mathbf{w}_{k+1} = \mathbf{w}_{k+1} / \|\mathbf{w}_{k+1}\|$$

One advantage of estimating one independent component at a time is in cases where not all of the components need to be estimated. Another advantage of FastICA over other algorithms is that its convergence is cubic as opposed to other algorithms that are based on gradient descent methods whose convergence is typically linear.

As mentioned earlier unlike PCA the order of the independent components or their corresponding directions cannot be determined. Thus, the order in which the FastICA algorithm outputs the vectors $\mathbf{w}$ has no statistical importance. However, as mentioned already the vectors that are output by the FastICA algorithm (or any ICA procedure) can be sorted in decreasing degree of nongaussianity which we can then interpreted as the degree of the "interestingness" of the directions of the independent components, as opposed to the degree of "faithfulness" in representation, for components that are obtained by a PCA procedure.

6.2 Derivation of the One-Unit FastICA Algorithm

First thing to note is that since the ICA model assumes that the independent components $y = \mathbf{w}'\mathbf{x}$ have unit variance $E[(\mathbf{w}'\mathbf{x})^2]$ must be constraint to unity. Since the data is assumed to be whitened we have

$$E[(\mathbf{w}'\mathbf{x})^2] = E[(\mathbf{w}'\mathbf{x})(\mathbf{w}'\mathbf{x})] = E[(\mathbf{w}'\mathbf{x})(\mathbf{x}'\mathbf{w})]$$

$$= E[\mathbf{w}'\mathbf{x}\mathbf{x}'\mathbf{w}] = \mathbf{w}'E[\mathbf{x}\mathbf{x}']\mathbf{w} = \mathbf{w}'I\mathbf{w} = \mathbf{w}'\mathbf{w}$$

Thus, constraining $E[(\mathbf{w}'\mathbf{x})^2] = 1$ is equivalent to requiring that $\|\mathbf{w}\| = 1$.

Maximizing the negentropy approximation $[E[G(\mathbf{w}'\mathbf{x})] - E[G(v)]]^2$ is equivalent to finding the optima (maximum or minimum) of $E[G(\mathbf{w}'\mathbf{x})]$. Using the method of Lagrange multipliers to find the optima of $E[G(\mathbf{w}'\mathbf{x})]$

subject to the constraint $\mathbf{w}'\mathbf{w} = 1$ we get

$$\frac{\partial}{\partial \mathbf{w}} \left(E\left[G(\mathbf{w}'\mathbf{x})\right] - \lambda(\mathbf{w}'\mathbf{w} - 1)\right)$$

$$= E\left[\frac{\partial}{\partial \mathbf{w}}(G(\mathbf{w}'\mathbf{x}))\right] - 2\lambda\mathbf{w}$$

$$= E\left[\mathbf{x}g(\mathbf{w}'\mathbf{x})\right] - \beta\mathbf{w} \qquad (\beta = 2\lambda)$$

Thus, the optima of $E\left[G(\mathbf{w}'\mathbf{x})\right]$ subject to the constraint $\mathbf{w}'\mathbf{w} = 1$ is obtained when

$$E\left[\mathbf{x}g(\mathbf{w}'\mathbf{x})\right] - \beta\mathbf{w} = 0$$

Recall that solutions to equations such as $\mathbf{f}(\mathbf{w}) = E\left[\mathbf{x}g(\mathbf{w}'\mathbf{x})\right] - \beta\mathbf{w} = 0$ can be found using Newton's method (appendix 7.8), with the iteration step defined as

$$\mathbf{w}_{k+1} = \mathbf{w}_k - J_{\mathbf{f}}^{-1}(\mathbf{w}_k)\mathbf{f}(\mathbf{w}_k)$$

Thus, to use Newton's method we need to compute the Jacobian of $\mathbf{f}(\mathbf{w}) = E\left[\mathbf{x}g(\mathbf{w}'\mathbf{x})\right] - \beta\mathbf{w} = 0$.

$$J_{\mathbf{f}} = \begin{pmatrix} (\nabla f_1)' \\ (\nabla f_2)' \\ \vdots \\ (\nabla f_d)' \end{pmatrix}$$

were $f_i(\mathbf{w}) = E[x_i g(\mathbf{w}'\mathbf{x})] - \beta w_i$. Now

$$\nabla f_i = E[\nabla(x_i g(\mathbf{w}'\mathbf{x}))] - \beta e_i = E[x_i g'(\mathbf{w}'\mathbf{x})\mathbf{x}] - \beta e_i$$

and therefore

$$J_{\mathbf{f}} = \begin{pmatrix} (E[x_1 g'(\mathbf{w}'\mathbf{x})\mathbf{x}] - \beta e_1)' \\ (E[x_2 g'(\mathbf{w}'\mathbf{x})\mathbf{x}] - \beta e_2)' \\ \vdots \\ (E[x_d g'(\mathbf{w}'\mathbf{x})\mathbf{x}] - \beta e_d)' \end{pmatrix} = E[\mathbf{x}\mathbf{x}' g'(\mathbf{w}'\mathbf{x})] - \beta I$$

Since the data is whitened we, to simplify the matrix inversion of $J_{\mathbf{f}}$ we can approximate $E[\mathbf{x}\mathbf{x}' g'(\mathbf{w}'\mathbf{x})]$ as follows

$$E[\mathbf{x}\mathbf{x}' g'(\mathbf{w}'\mathbf{x})] \approx E[\mathbf{x}\mathbf{x}']E[g'(\mathbf{w}'\mathbf{x})] = E[g'(\mathbf{w}'\mathbf{x})]I$$

Thus, $J_{\mathbf{f}}$ becomes diagonal and its inverse is equal to

$$J_{\mathbf{f}}^{-1} = \frac{1}{E[g'(\mathbf{w}'\mathbf{x})] - \beta} I$$

and the Newton iteration step becomes

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \frac{E\left[\mathbf{x}g(\mathbf{w}_k'\mathbf{x})\right] - \beta\mathbf{w}_k}{E[g'(\mathbf{w}_k'\mathbf{x})] - \beta}$$

This could be simplified by multiplying both sides of the equation by $\beta - E[g'(\mathbf{w}_k'\mathbf{x})]$ which will give

$$\mathbf{w}_{k+1} = \frac{1}{(\beta - E[g'(\mathbf{w}_k'\mathbf{x})])} \left(E[\mathbf{x}g(\mathbf{w}_k'\mathbf{x})] - E[g'(\mathbf{w}_k'\mathbf{x})]\mathbf{w}_k\right) \tag{5}$$

Since we are interested in the direction of $\mathbf{w}_{k+1}$ the scaler on the right side of eq. 5 is not significant and therefore can be omitted (its effect can be canceled by normalization), which will then give the FastICA iteration step.

7 Appendix

7.1 Moment and Cumulant Generating Functions

The moment generating function (mgf) of a random variable X is defined as the expected value of e^{tX}

$$M_X(t) = E[e^{tX}] \quad t \in \mathbb{R}$$

The mgf can be used to generate the moments (about the origin) of a pdf (provided that the mgf exists or is differentiable at an interval around $t = 0$) as follows

$$e^{tX} = 1 + tX + \frac{t^2X^2}{2!} + \frac{t^3X^3}{3!} + \cdots = \sum_{k=0}^{\infty} \frac{t^kX^k}{k!}$$

thus

$$\begin{aligned} M_X(t) = E[e^{tX}] &= E[1 + tX + \frac{t^2X^2}{2!} + \frac{t^3X^3}{3!} + \cdots] \\ &= E[1] + E[tX] + E[\frac{t^2X^2}{2!}] + E[\frac{t^3X^3}{3!} + \cdots] \\ &= 1 + tE[X] + \frac{t^2E[X^2]}{2!} + \frac{t^3E[X^3]}{3!} + \cdots \\ &= \sum_{k=0}^{\infty} \frac{t^kE[X^k]}{k!} \end{aligned}$$

Hence if we differentiate $M_X(t)$ n times with respect to t and then set $t = 0$ we obtain the n-th moment (about the origin) of X, i.e.

$$\begin{aligned} M_X^n(0) &= \left.\frac{\mathrm{d}^n}{\mathrm{d}^n t}\right|_{t=0} M_X(t) \\ &= \left.\frac{\mathrm{d}^n}{\mathrm{d}^n t}\right|_{t=0} E[e^{tX}] = E\left[\frac{\mathrm{d}^n}{\mathrm{d}^n t}e^{tX}\right]_{t=0} \\ &= E\left[X^n e^{tX}\right]_{t=0} = E[X^n] \end{aligned}$$

If X has a continues pdf $f(x)$ then the mgf is given by

$$\begin{aligned} M_X(t) = \int_{-\infty}^{\infty} e^{tx} f(x)dx &= \int_{-\infty}^{\infty} \left(1 + tx + \frac{t^2x^2}{2!} + \frac{t^3x^3}{3!} + \cdots\right) f(x)dx \\ &= 1 + t\mu_1' + \frac{t^2\mu_2'}{2!} + \frac{t^3\mu_3'}{3!} + \cdots \end{aligned}$$

where μ_i' is the i-th moment (about the origin). In particular if X is a Gaussian random variable then its mgf is given by

$$\begin{aligned} M_X(t) = \int_{-\infty}^{\infty} e^{tx} f(x)dx &= \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} e^{tx} e^{\frac{(x-\mu)^2}{-2\sigma^2}} dx \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} e^{\frac{-2\sigma^2 tx + (x-\mu)^2}{-2\sigma^2}} dx \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} e^{\frac{x^2 - 2x(\mu+\sigma^2 t) + \mu^2}{-2\sigma^2}} dx \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} e^{\frac{(x-(\mu+\sigma^2 t))^2 + \mu^2 - (\mu+\sigma^2 t)^2}{-2\sigma^2}} dx \end{aligned}$$

$$= \left(\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(x-(\mu+\sigma^2 t))^2}{-2\sigma^2}} dx \right) e^{\frac{\mu^2-(\mu+\sigma^2 t)^2}{-2\sigma^2}}$$

$$= 1 \cdot e^{\frac{\mu^2-\mu^2-2\sigma^2\mu t-\sigma^4 t^2}{-2\sigma^2}}$$

$$= e^{\mu t+\sigma^2 t^2/2}$$

Some properties of mgfs

1. if X has mgf $M_X(t)$ then the mgf of a linear combination of X, say $Y = a + bX$ is

   $$M_Y(t) = E[e^{tY}] = E[e^{t(a+bX)}] = E[e^{ta}e^{tbX}] = e^{ta}E[e^{tbX}] = e^{ta}M_X(tb)$$

   In particular if $Y = X - \mu$, then $e^{-\mu t}M_X(t)$ is the mgf of X about the mean μ.

2. if $X_1, X_2, \ldots, X_n$ are independent random variables with mgfs $M_{X_i}(t)$ then a new random variable $Y = X_1 + X_2 + \ldots + X_n$ has mgf

   $$M_Y(t) = E[e^{tY}] = E[e^{t(X_1+X_2+\ldots+X_n)}] = E[e^{tX_1}e^{tX_2}\cdots e^{tX_n}] = \prod_{i=1}^{n} E[e^{tX_i}] = \prod_{i=1}^{n} M_{X_i}(t)$$

The cumulant generating function (cgf) of a random varaible X is given by

$$K_X(t) = \ln(M_X(t)) = \sum_{n=1}^{\infty} k_n \frac{t^n}{n!}$$

where k_n is called the n-th cumulant of X and is given by

$$k_n = \frac{\mathrm{d}^n}{\mathrm{d}t^n} K_X(0)$$

To find each k_n we need to compute the power series expansion of $\ln(M_X(t))$ as follows

$$K_X(t) = \ln(M_X(t)) = \ln(1 + t\mu_1' + \frac{t^2\mu_2'}{2!} + \frac{t^3\mu_3'}{3!} + \cdots)$$

$$= ln(1+x) \qquad \text{where } x = t\mu_1' + \frac{t^2\mu_2'}{2!} + \frac{t^3\mu_3'}{3!} + \cdots$$

now

$$ln(1+x) = x - x^2/2 + x^3/4 - x^4/4 \ldots$$

and

$$x^2 = \mu_1'^2 t^2 + \mu_1'\mu_2' t^3 + (2\mu_3'\mu_1'/3! + \mu_2'^2/4)t^4 + \cdots$$

$$x^3 = \mu_1'^3 t^3 + 3\mu_1'^2\mu_2' t^4/2 + \cdots$$

$$x^4 = \mu_1'^4 t^4 + \cdots$$

Gathering up the terms by powers of t we get

$$K_X(t) = \mu_1' t + (\mu_2' - \mu_1'^2)t^2/2 + (\mu_3' - 3\mu_1'\mu_2' + 2\mu_1'^3)t^3/3! + (\mu_4' - 4\mu_3'\mu_1' - 3\mu_2'^2 + 12\mu_2'\mu_1'^2 - 6\mu_1'^4)t^4/4! + \cdots$$

Taking the n-th derivative we see that of $K_X(t)$ and setting $t = 0$ we get

$$k_1 = \mu_1' = E[X] = \mu$$

$$k_2 = \mu_2' - \mu^2 = E[(X-\mu)^2] = \sigma^2$$

$$k_3 = \mu'_3 - 3\mu'_1\mu'_2 + 2\mu'^3_1 = E[(X-\mu)^3]$$
$$k_4 = \mu'_4 - 4\mu'_3\mu'_1 - 3\mu'^2_2 + 12\mu'_2\mu'^2_1 - 6\mu'^4_1 = E[(X-\mu)^4] - 3E[(X-\mu)^2]^2 = E[(X-\mu)^4] - 3\sigma^4$$

We can see the the first three cumulants are equal to the first three moments (about the mean), however the fourth cumulant which is a normalized version of the fourth moment, is known as the *kurtosis*. For a Gaussian all cumulants of order higher than two are zero. In particular the fourth moment of a Gaussian is equal to $3E[(X-\mu)^2]^2$, thus the kurtosis is zero. The kurtosis is commonly used to measure the Gaussianity or non-Gaussianity of a random variable. Kurtosis can be positive or negative, random variables that have negative kurtosis are called subgaussian or *platykurtic*, and those with positive are called supergaussian or *leptokurtic*. Supergaussian rv have a spiky pdf with heavy tails like the Laplace distribution $p(x) = 1/\sqrt{2}exp(\sqrt{(}2)|x|)$ (normalized to unit variance). Subgaussian rv have a relatively flat pdf which is rather constant near zero. A typical example is the uniform distribution. There are non-gaussian rv that have zero kurtosis, but they can be considered very rare. Typically nongaussianity is measured by the absolute value of the kurtosis.

Some properties of kurtosis:

1. if X_1 and X_2 are independent then
$$kurt(X_1 + X_2) = kurt(X_1) + kurt(X_2)$$
2. if a is some constant the
$$kurt(aX) = a^4 kurt(X)$$

7.2 Some Results About Gaussian Random Variables

Several theoretical results concerning Gaussian random variables play a crucial role in independent component analysis. Two of them which are presented in this section explain why the independent components cannot be Gaussian. Another result which forms the basis for a family of contrast (criterion) functions used in ICA, and which measures the nongaussianity of random variables will be presented in the next section along with other results and definitions pertaining to information theory.

Proposition 1. *For Gaussian random variables independence is equivalent to uncorrelatedness.*

Proof: Since independence implies uncorrelatedness for any random variables, What needs to be shown is that for Gaussian random variables uncorrelatedness implies independence. Let $X_1, X_2, \ldots, X_d$ be d uncorrelated Gaussian random variables, with pdf's $g_1(x_1) \sim N(\mu 1, \sigma_1), g_2(x_2) \sim N(\mu 2, \sigma_2), \ldots, g_d(x_d) \sim N(\mu_d, \sigma_d)$ respectively, and let $\mathbf{x}$ be a random vector whose elements are realizations of the d random variables. The joint pdf of these random variables is given by

$$g(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)'\Sigma^{-1}(\mathbf{x}-\mu)\right)$$

Since the random variables are uncorrelated, their covariance matrix Σ is diagonal which in turn implies that the determinant of Σ is equal to the product of its elements along the diagonal, the inverse of Σ is also diagonal, and the quadratic form $\big(\mathbf{x}-\mu)'\Sigma^{-1}(\mathbf{x}-\mu)\big)$ can be written as a summation. Thus, $g(\mathbf{x})$ can be written as

$$g(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}(\sigma_1\sigma_2\cdots\sigma_d)} \exp\left(-\frac{1}{2}\left(\frac{(x_1-\mu_1)^2}{\sigma_1^2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2} + \ldots + \frac{(x_d-\mu_d)^2}{\sigma_d^2}\right)\right)$$

$$= \left(\frac{1}{\sqrt{2\pi}\sigma_1} \exp\frac{(x_1-\mu_1)^2}{-2\sigma_1^2}\right)\left(\frac{1}{\sqrt{2\pi}\sigma_2} \exp\frac{(x_2-\mu_2)^2}{-2\sigma_2^2}\right)\cdots\left(\frac{1}{\sqrt{2\pi}\sigma_d} \exp\frac{(x_d-\mu_d)^2}{-2\sigma_d^2}\right)$$
$$= g_1(x_1)g_2(x_2)\cdots g_d(x_d)$$

i.e., the joint equals the product of the marginals which shows that the variables are independent.

Lemma 1. *If $X_1, X_2, \ldots, X_d$ are mutually independent Gaussian random variables then a linear combination of them yielding a new variable Y where $Y = a_1X_1 + a_2X_2+, \ldots, a_dX_d$ is also Gaussian.*

Proof: From appendix 7.1 we know that the mgf of a Gaussian random variable X is $M_X(t) = e^{\mu t+\sigma^2t^2/2}$ and that the mfg of a scaler a multiplied by X is $M_{aX}(t) = M_X(at) = e^{\mu at+\sigma^2a^2t^2/2}$. We also know from appendix 7.1 that the mgf of a sum of independent random variables is equal to the product of their individual mgf's. Since the random variables a_iX_i are independent (this is easy to show since the X_i's are independent), the mgf of Y is given by

$$M_Y(t) = \prod_{i=1}^{d} M_X(ta_i) = \prod_{i=1}^{d} e^{(\mu_i a_i t+\sigma_i^2a_i^2t^2/2)}$$
$$= e^{\sum_{i=1}^{d}(\mu_i a_i t+\sigma_i^2a_i^2t^2/2)}$$
$$= e^{t(\sum_{i=1}^{d}\mu_i a_i)+t^2/2\sum_{i=1}^{d}(\sigma_i^2a_i^2)}$$

which is the mgf of a Gaussian with mean $\sum_{i=1}^{d}(\mu_i a_i)$ and variance $\sum_{i=1}^{d}(\sigma_i^2a_i^2)$.

Proposition 2. *If $X_1, X_2, \ldots, X_d$ are independent Gaussian random variables then any orthogonal transformation A will result in a new set of variables $Y_1, Y_2, \ldots, Y_d$ which are also Gaussian and independent.*

Proof: Let $\mathbf{x}$ and $\mathbf{y}$ where $\mathbf{y} = A\mathbf{x}$ be two random vectors whose elements are realizations of $X_1, X_2, \ldots, X_d$ and $Y_1, Y_2, \ldots, Y_d$. Furthermore, assume without loss of generality that each of the X's has zero mean and unit variance, i.e. $g_1(x_1) = g_2(x_2) = \ldots = g_d(x_d) \sim N(0, 1)$ (this is an assumption that is any how made by ICA algorithms due to one of the ambiguities of the ICA model). We know from lemma 1 that each Y_i that is a linear combination of the X_i's with coefficients given by the elements of row i in A, is Gaussian. Furthermore, since $AA' = A'A = I$ and the covariance of $\mathbf{x}$ is $E[\mathbf{x}\mathbf{x}'] = I$, the covariance matrix of $\mathbf{y}$ is given by

$$E[\mathbf{y}\mathbf{y}'] = E[A\mathbf{x}\mathbf{x}'A'] = AE[\mathbf{x}\mathbf{x}']A' = AIA' = AA' = I$$

Thus $\mathbf{y}$ has the distribution of a multivariate Gaussian with a diagonal covariance matrix. From proposition 1 we know that because $\mathbf{y}$ has a diagonal covariance matrix its pdf is equal to the product of its marginals. Therefore the random variables Y_i are independent. Furthermore, since $E[\mathbf{y}] = E[A\mathbf{x}] = AE[\mathbf{x}] = A\mathbf{0} = \mathbf{0}$, and $E[\mathbf{y}\mathbf{y}'] = E[\mathbf{x}\mathbf{x}'] = I$, each random variable Y_i has the same distribution as X_i, namely a Gaussian distribution with zero mean and unit variance.

7.3 Information Theory

7.3.1 The Entropy of The Gaussian Distribution

For a univariate Gaussian distribution the pdf $g(x)$ is defined as

$$g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

were μ and σ^2 are the mean and variance of the distribution.

For the multivariate case the pdf $g(\mathbf{x})$ were $\mathbf{x}$ is a random vector of d elements and Σ is the covariance matrix is defined as

$$g(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)'\Sigma^{-1}(\mathbf{x}-\mu)\right)$$

The entropy of a continuous random variable often refereed to as differential entropy is defined as (using the natural logarithm)

$$H = -E[\ln g(\mathbf{x})] = -\int_{-\infty}^{\infty} g(\mathbf{x}) \ln g(\mathbf{x}) d\mathbf{x}$$

Thus, for a univariate Gaussian the entropy is

$$H = -E[\ln g(x)] = -E\left[\ln\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)\right)\right] = \frac{1}{2}\ln 2\pi\sigma^2 + \frac{1}{2\sigma^2}E[(x-\mu)^2]$$

$$= \frac{1}{2}\ln 2\pi\sigma^2 + \frac{1}{2\sigma^2}\sigma^2 = \frac{1}{2}(\ln 2\pi\sigma^2 + 1)$$

and in the multivariate case the entropy is

$$H = -E[\ln g(\mathbf{x})] = -E\left[\ln\left(\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)'\Sigma^{-1}(\mathbf{x}-\mu)\right)\right)\right]$$

$$= \frac{d}{2}\ln 2\pi + \frac{1}{2}\ln|\Sigma| + \frac{1}{2}E[(\mathbf{x}-\mu)'\Sigma^{-1}(\mathbf{x}-\mu)]$$

now

$$E[(\mathbf{x}-\mu)'\Sigma^{-1}(\mathbf{x}-\mu)] = E[tr((\mathbf{x}-\mu)'\Sigma^{-1}(\mathbf{x}-\mu))] = E[tr(\Sigma^{-1}(\mathbf{x}-\mu)(\mathbf{x}-\mu)')]$$
$$= tr(\Sigma^{-1}E[(\mathbf{x}-\mu)(\mathbf{x}-\mu)']) = tr(\Sigma^{-1}\Sigma) = tr(I) = d$$

Therefore

$$H = \frac{d}{2}\ln 2\pi + \frac{1}{2}\ln|\Sigma| + \frac{d}{2} = \frac{1}{2}(d\ln 2\pi + \ln|\Sigma| + d)$$

7.3.2 Kullback-Leibler Distance (Relative Entropy)

The Kullback-Leibler (KL) measure of two distributions with pdf's $f(\mathbf{x})$ and $g(\mathbf{x})$ denoted $D(f\|g)$ can be thought of a distance measure between the two distributions. For the continuous case it is defined as

$$D(f\|g) = \int f(\mathbf{x}) \log\frac{f(\mathbf{x})}{g(\mathbf{x})}\, d\mathbf{x} = -\int f(\mathbf{x}) \log\frac{g(\mathbf{x})}{f(\mathbf{x})} d\mathbf{x} = E\left[\log\frac{f(\mathbf{x})}{g(\mathbf{x})}\right]$$

The KL distance can be shown to be always nonnegative and equal to zero when the two distributions are the same, however is is not symmetric. The proof that KL is nonnegative is as follows

$$\begin{aligned} -D(f\|g) &= \int f(\mathbf{x}) \log\frac{g(\mathbf{x})}{f(\mathbf{x})} d\mathbf{x} \\ &\leq \log\left(\int f(\mathbf{x})\frac{g(\mathbf{x})}{f(\mathbf{x})} d\mathbf{x}\right) \quad \text{Jensen's inequality} \\ &= \log\left(\int g(\mathbf{x}) d\mathbf{x}\right) = \log 1 = 0 \end{aligned}$$

were Jensen's inequality states that if a function f is convex and X is a random variable then

$$E[f(X)] \geq f(E[X])$$

or equivalently

$$\int p(x)f(x)dx \geq f\left(\int p(x)xdx\right)$$

were $p(x)$ is the pdf of x. When f is concave then the inequity is reversed. Since log is a concave function we get the inequity used in the proof. Another somewhat simpler proof uses the inequity $\log x \leq x - 1$.

$$\begin{aligned} -D(f\|g) &= \int f(\mathbf{x}) \log \frac{g(\mathbf{x})}{f(\mathbf{x})} d\mathbf{x} \\ &\leq \int f(\mathbf{x}) \left(\frac{g(\mathbf{x})}{f(\mathbf{x})} - 1\right) d\mathbf{x} \quad (\log x \leq x - 1) \\ &= \int (g(\mathbf{x}) - f(\mathbf{x}))d\mathbf{x} = \int g(\mathbf{x})d\mathbf{x} - \int f(\mathbf{x})d\mathbf{x} = 1 - 1 = 0 \end{aligned}$$

Note that in both proofs equality holds, i.e. $D(f\|g) = 0$ when $g(\mathbf{x}) = f(\mathbf{x})$.

7.3.3 Mutual Information

If X and Y are two continuous random variables with joint pdf $f(x, y)$ the the **joint entropy** is

$$H(X, Y) = -\int\int f(x, y) \log f(x, y)dx\ dy = -E[\log f(x, y)]$$

When we have more than two random variables and we are interested in their joint entropy we can put them in vector form to get

$$H = -\int f(\mathbf{x}) \log f(\mathbf{x})d\mathbf{x}$$

were $\mathbf{x}$ is a random vector of d random variables.

The conditional entropy of two continuous random variables is defined as

$$H(Y|X) = -E[\log f(y|x)] = -\int\int f(x, y) \log f(y|x)dx\ dy = -\int f(x) \left(\int f(y|x) \log f(y|x)dy\right) dx$$

$$= -\int f(x)H(Y|X = x)dx$$

Note that

$$\begin{aligned} H(X, Y) &= -E[\log f(x, y)] = -E[\log f(x)f(y|x)] = -E[\log f(x) + \log f(y|x)] \\ &= -E[\log f(x)] - E[\log f(y|x)] = H(X) + H(Y|X) \end{aligned}$$

and similarly

$$H(X, Y) = H(Y) + H(X|Y)$$

The interpretation is that the uncertainty (entropy) about both X and Y is equal to the uncertainty (entropy) we have about X, plus whatever we have about Y, given that we know X.

The **mutual information** denoted $I(X; Y)$ between to continuous random variables X and Y is the relative entropy between the joint distribution and the product distribution (product of marginals) , that is,

$$I(X; Y) = D(f(x, y)\|f(x)f(y)) = \int\int f(x, y) \log \frac{f(x, y)}{f(x)f(y)} dx\ dy = E\left[\log \frac{f(x, y)}{f(x)f(y)}\right]$$

Note that

$$E\left[\log\frac{f(x,y)}{f(x)f(y)}\right] = E\left[\log\frac{f(y)f(x|y)}{f(x)f(y)}\right] = E\left[\log\frac{f(x|y)}{f(x)}\right] = E\left[\log f(x|y) - \log f(x)\right]$$

$$= E\left[\log f(x|y)\right] - E\left[\log f(x)\right] = H(X) - H(X|Y)$$

From the last equation we can think of mutual information as the reduction in uncertainty about X due to the knowledge of Y. It can easily be checked that $I(X;Y) = I(Y;X)$, and using $H(X,Y) = H(X) + H(Y|X)$ we also have

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

So mutual information can also be thought of as the uncertainty in X plus the uncertainty in Y less the uncertainty in both X and Y.

From the definitions of mutual information we can see that if X and Y are independent then $I(X;Y) = I(Y;X) = 0$. Thus mutual information can be used as a measure of dependence between random variables. Extending to more than two random variables and using vector notation, we can define mutual information as

$$I(x_1; x_2; \ldots; x_d) = -H(\mathbf{x}) + \sum_{i=1}^{d} H(x_i)$$

7.3.4 The Gaussian Maximizes Entropy for a Given Covariance

The following is a proof that a Gaussian random variable has the largest entropy among all random variables of equal mean and variance. For simplicity we will assume that the distributions have zero mean and that the natural logarithm used.

Let $g(x)$ be the pdf of a gaussian random variable, and $f(x)$ the distribution of some other unknown random variable. What we would like to show is that $H(g) \geq H(f)$.

$$\begin{aligned}
0 &\leq D(f\|g) \\
&= \int f(x)\log\frac{f(x)}{g(x)}dx \\
&= \int f(x)\log f(x)dx - \int f(x)\log g(x)dx \\
&= \int f(x)\log f(x)dx - \int f(x)\left(\log\frac{1}{\sqrt{2\pi\sigma^2}} - \frac{x^2}{2\sigma^2}\right)dx \\
&= \int f(x)\log f(x)dx - \log\frac{1}{\sqrt{2\pi\sigma^2}}\int f(x)dx + \frac{1}{2\sigma^2}\int f(x)x^2dx \\
&\quad \left(\int f(x)dx = \int g(x)dx = 1, \quad \int f(x)x^2dx = \int g(x)x^2dx = \sigma^2\right) \\
&= \int f(x)\log f(x)dx - \log\frac{1}{\sqrt{2\pi\sigma^2}}\int g(x)dx + \frac{1}{2\sigma^2}\int g(x)x^2dx \\
&= \int f(x)\log f(x)dx - \int g(x)\log g(x)dx \\
&= -H(f) + H(g) \\
&\quad \therefore H(g) \geq H(f)
\end{aligned}$$

Following the same line this proof can be extended to multivariate distributions.

7.4 Differentials

Let $y = f(x)$ be some function of x then the derivative (if it exists) is defined as

$$\frac{dy}{dx} = f'(x) = \lim_{\triangle x \to 0} \frac{\triangle y}{\triangle x} = \lim_{\triangle x \to 0} \frac{f(x + \triangle x) - f(x)}{\triangle x}$$

We are used to the Leibniz notation dy/dx to denote the derivative, in this case is just a symbol and is not regarded as a ratio. However dx and dy which are called differentials also have a meaning.

Definition 1. *Let $y = f(x)$, where f is a differentiable function. Then the* ***differential*** *dx is an independent variable that can be given the value of any real number. The* ***differential*** *dy is then defined in terms of dx by*

$$dy = f'(x)dx$$

That is, dy is a dependent variable that depends of the values of x and dx. The geometric meaning of the differential is shown in fig. 3. Let $P(x, f(x))$ and $Q(x + \triangle x, f(x + \triangle x))$ be two points on the graph of f, and let $dx = \triangle x$. So the corresponding change in y is $\triangle y = f(x + \triangle x) - f(x)$, and the slope of the tangent line PR is the derivative $f'(x)$. Thus the distance from S to R is $f'(x)dx = dy$. Therefore dy represents the amount that the tangent rises or falls when x changes by $\triangle x$, whereas $\triangle y$ the amount $f(x)$ rises or falls when x changes by $\triangle x$. Since

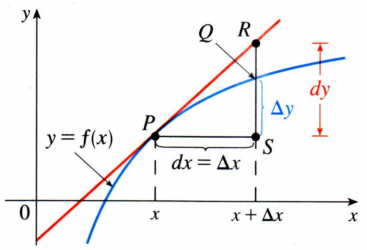


Figure 3:

$$\frac{dy}{dx} = \lim_{\triangle x \to 0} \frac{\triangle y}{\triangle x}$$

when $\triangle x$ is small we have

$$\frac{dy}{dx} \approx \frac{\triangle y}{\triangle x}$$

and if we take $\triangle x = dx$ then

$$\triangle y \approx dy$$

which can be used in computing approximate value of functions. That is, suppose $f(a)$ is known and we want to approximate $f(a + \triangle x)$ where $\triangle x$ is small. Then

$$f(a + \triangle x) = f(a) + \triangle y \approx f(a) + dy = f(a) + f'(a)dx \tag{6}$$

For example

$$\sqrt[3]{65} = \sqrt[3]{64 + 1} \approx \sqrt[3]{64} + f'(64)1 = 4 + 1/3(64)^{-2/3}1$$
$$= 4 + 1/48 = 4.021$$

7.5 Linear and Quadratic Approximations

The equation of the tangent line to the curve $y = f(x)$ at $(a, f(a))$ is

$$y = f(a) + f'(a)(x - a) = f(a) + f'(a)dx$$

So for the approximation given in (6) we are in fact using the tangent line at $(a, f(a))$ as an approximation to the curve $y = f(x)$ when x is near a. For this reason the approximation

$$f(x) = f(a) + f'(x)(x - a)$$

is called the **linear approximation** of f at a and

$$L(x) = f(a) + f'(x)(x - a)$$

is called the **linearization** of f at a. For example using linearization to approximate $\sqrt{3.98}$ for $f(x) = \sqrt{x + 3}$ at $a = 1$.

$$f'(x) = 1/2(x + 3)^{-1/2}$$

$$L(x) = f(1) + f'(1)(x - 1) = 2 + 1/4(x - 1) = (7 + x)/4$$

$$\sqrt{3.98} = \sqrt{0.98 + 3} = (7 + 0.98)/4 \approx 1.995$$

$L(x)$ is the best first-degree (linear) approximation to $f(x)$ near $x = a$. For a better approximation we can use a second-degree (quadratic) approximation $P(x)$, i.e. approximate a curve by a parabola. Given that $P(x) = A + Bx + Cx^2$ to find the coefficients A, B, C we assume $P(a) = f(a), P'(a) = f'(a), P''(a) = f''(a)$. Since $P'(x) = B + 2Cx$ and $P''(x) = 2C$, the coefficients can be found by solving the three equations. In general, if we want to approximate a function by a quadratic function P near a point a, it is best to write P in the form

$$P(x) = A + B(x - a) + C(x - a)^2$$

and then by solving the three equations we get

$$P(x) = f(a) + f'(a)(x - a) + \frac{f''(a}{2}(x - a)^2$$

which is called the quadratic approximation to $f(x)$ near a. Note that the linear and quadratic approximations are also called the 1st and 2nd degree Taylor approximations (polynomials) of f at a.

7.6 The Gradient, Jacobian and Hessian

7.6.1 The Gradient

The gradient is a vector operator denoted ∇ and sometimes also called Del or nabla. It is a generalization of the ordinary derivative, and as such conveys information about the rate of change of a function relative to small variations in the independent variables. The gradient of a multivariate function f is customarily denoted by ∇f. More formally, if $x_1, x_2, \ldots, x_n$ are the variables of the multivariate function $f(x_1, x_2, \ldots, x_n)$, or in vector form $f(\mathbf{x})$ where $\mathbf{x}$ is a $n \times 1$ column vector, and $e_1, e_2, \ldots, e_n$ are the vectors of the standard basis then

$$\nabla f = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} e_i$$

i.e. ∇f is basically a the vector of first-order partial derivatives of f. Geometrically the direction of the vector ∇f is the direction of the greatest positive change, or increase, in f. For example consider a hill whose height at a point (x, y) is $f(x, y)$. The gradient of f at a point is in the direction of the steepest slope/grade

at that point. The magnitude of the gradient tells how steep the slope actually is. Just as with regular derivatives, the gradient can also be viewed as best linear approximation to a function f at particular point $\mathbf{a} \in R^n$, that is

$$f(\mathbf{x}) = f(\mathbf{a}) + \nabla f(\mathbf{a})(\mathbf{x} - \mathbf{a})$$

The gradient can also be viewed is a particular case of the Jacobian. Which leads us to the definition of a Jacobian.

7.6.2 The Jacobian

The Jacobian is shorthand for either the Jacobian matrix denoted by J or its determinant, the Jacobian determinant denoted by $|J|$. The Jacobian matrix is the matrix of all first-order partial derivatives of a vector-valued function $F : R^n \to R^m$. Given

$$F = \begin{pmatrix} f_1(x_1, x_2, \ldots, x_n) \\ f_2(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_m(x_1, x_2, \ldots, x_n) \end{pmatrix}$$

$$J = \frac{\partial(f_1, f_2, \ldots, f_m)}{\partial(x_1, x_2, \ldots, x_n)} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

Notice that the elements of each row are the partial derivatives of the function f_i with respect to all variables which is why J can also be defined in terms of gradients.

$$J = \begin{pmatrix} (\nabla f_1)' \\ (\nabla f_2)' \\ \vdots \\ (\nabla f_m)' \end{pmatrix}$$

The Jacobian has two main applications, the first, as with the gradient the jacobian can be used to find the best linear approximation of a function $F : R^n \to R^m$ near a point $\mathbf{a} \in R^n$ as follows

$$F(\mathbf{x}) = F(\mathbf{a}) + J(\mathbf{a})(\mathbf{x} - \mathbf{a})$$

Another important use of the Jacobian is when $n = m$. Similar to the test of linear dependence of a set of linear equations through regular determinants. The Jacobian determinant permits us to test both linear and non-linear dependence of a set equations. If $|J| = 0$ the the equations a functionally dependent, otherwise they are independent. The jacobian has other very important uses, like giving important information about the behavior of F near a certain point, however these is beyond the scope of this introduction.

7.6.3 The Hessian

Given a function $f : R^n \to R$ having second order partial derivatives, the Hessian matrix of f is the matrix of partial second derivatives

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n x_1} & \frac{\partial^2 f}{\partial x_n x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

Note that H is symmetric because of the equality of mixed partials. Also note that $H = J(\nabla f)$, i.e. the Hessian is equal to the Jacobian of the gradient of f. One of the main uses of the Hessian is to test for second-order conditions of stationary points (minima, maxima). Let $|H|$ be the determinant of H, and $|H_i|$ be the determinant of the i-th *principle minor*, were the the i-th principle minor is the determinant of an $i \times i$ matrix obtained by deleting the last $n - i$ rows and columns of the Hessian. For example $|H_n| = |H|$. If all the principle minors of $|H|$ are positive $|H|$ is said to be positive definite, and the second-order conditions for a relative minima are met. If all the principle minors of $|H|$ alternate in sign, then $|H|$ is said to be negative definite, and the second-order conditions for a relative maxima are met.

7.7 Gradient Descent

Gradient descent is an optimization algorithm for finding the nearest local minimum of a function f which presupposes that the gradient of the function can be computed. The method of gradient descent starts at a point x_0 (typically chosen at random) and, as many times as needed until convergence, moves from x_n to x_{n+1} by taking steps proportional to the negative of the gradient $(-\nabla f(x_n))$ of the function at the current point. If instead one searches for a local maximum then one has to take steps proportional to the gradient, a method called *gradient ascent*. More formally gradient descent can be described as follows

$$x_{n+1} = x_n - \lambda \nabla f(x_n)$$

where $\lambda > 0$ determines the magnitude or size of the step taken at each iteration. Gradient descent is based on the observation that if the real-valued function f is defined and differentiable in a neighborhood of a point, say x_n, then f decreases fastest if one goes from x_n in the direction of the negative of the gradient of f at x_n. Note that close to the minimum $\nabla f(x_n) \approx 0$, in which case $x_{n+1} \approx x_n$, showing why the algorithm is likely to converge. An illustration of the method when applied to a one-dimensional function $f(x) = x^3 - 2x^2 + 2$ with $\lambda = 0.1$ is depicted in fig. 4. Gradient Descent has two main weaknesses, the first is that the algorithm can take many iterations to converge towards a local minimum, if the curvature in different directions is very different. The second is that finding the optimal value for λ which plays a crucial role in the convergence of the algorithm may be hard. If it is too small the steps taken at each iteration are small and consequently convergence to the minimum may be very slow. On the other hand if it is too large, the algorithm may oscillate around the the minimum and convergence may be not possible. An often better method which does not require λ is Newton's method described in the next section.

7.8 Newton's Method (iteration)

Newton's method a.k.a Newton iteration or the Newton-Raphson method is used to find the roots of an equation of the form $f(x) = 0$ where f is a differentiable function. For a quadratic, third and forth degree equations there are formulas to find the roots, however if f is a polynomial of degree five or higher there

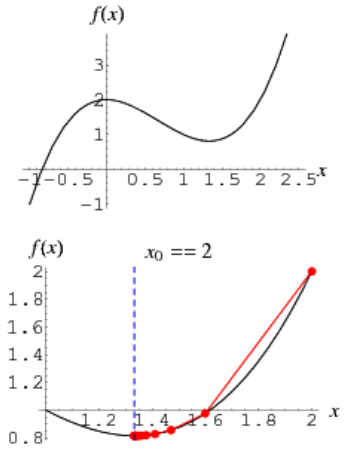


Figure 4:

are no such formulas, in which case methods such as Newton's must be used. The idea behind Newton's method is shown in fig 5. The root we are trying to find is labeled r. The first approximation of the root x_1 is obtained by guessing. The tangent line to $f(x)$ at $(x_1, f(x_1)$ is L and it intersects the x-intercept at x_2. If x_1 is close enough to r then typically x_2 will be even closer.

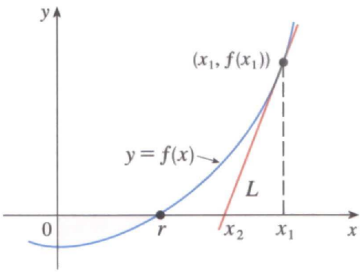


Figure 5:

To find a formula for x_2 in terms of x_1 we use the fact the the slope of L is $f'(x_1)$, so its equation is

$$f(x) - f(x_1) = f'(x_1)(x - x_1)$$

since x_2 is the x-intercept of L, we set $f(x) = 0$ and obtain

$$0 - f(x_1) = f'(x_1)(x_2 - x_1)$$

and if $f'(x_1) \neq 0$ then we can solve for x_2

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

if we repeat this procedure replacing x_1 by x_2 using the tangent line at $(x_2, f(x_2))$, and then replace x_2 by x_3 and so on, we will obtain a sequence of approximations $x_2, x_3, x_4, \ldots$ as shown in fig. 6 that will get closer an closer to r, i.e. converges to r or more formally $\lim_{x\to\infty} x_n = r$.

The general formula for Newtons method is given by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

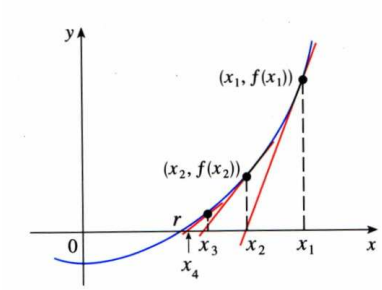


Figure 6:

The intuition behind the idea of convergence is that as x_n gets closer to r the slop at $(x_n, f(x_n)$ get closer to the slope at $(r, 0)$, now at $(r, 0)$ the slope intersects the x-intercept at r, and therefore once r is reached all the approximations thereafter will remain equal to r.

One of the more famous applications of Newton's method is for computing $\sqrt{a}$. If we let $f(x) = x^2 - a$ then

$$x_{n+1} = 1/2(x_n + a/x_n)$$

7.9 Newton's Method-Extension to Several Variables and Stationary Points

Another way of deriving Newton's formula is through a Taylor expansion. Suppose $f(x)$ is continuous and differentiable, so it may be be expanded as a Taylor series. If we replace x with x_{n+1} and a with x_n, then the Taylor expansion of x_{i+n} about x_n becomes:

$$f(x_{n+1}) = f(x_n) + f'(x_n)(x_{n+1} - x_n) + f''(x_n)(x_{n+1} - x_i)^2 \dots$$

Suppose x_{n+1} is very close to the root we are trying to find, in which case $f(x_{n+1}) \approx$, furthermore suppose that x_n and x_{n+1} are very close so that $(x_{n+1} - x_n)^2$ and higher powers can be neglected. Then the above Taylor series simplifies to

$$0 = f(x_n) + f'(x_n)(x_{n+1} - x_n)$$

Solving for x_{n+1} will then give Newton's formula.

Now lets suppose we have more than one variable and equation to solve, for example lets take two variables, were the extension to three or more will follow along the same lines.

Suppose we are solving the two simultaneous equations with two variables

$$f_1(x, y) = 0 \qquad f_2(x, y) = 0$$

where both functions are continuous and differentiable. Then the two-dimensional first-order Taylor expansions for f_1, f_2 which treats the x and y contributions separately are:

$$f_1(x_{n+1}, y_{n+1}) = f_1(x_n, y_n) + \left.\frac{\partial f_1}{\partial x}\right|_{x=x_n} (x_{n+1} - x_n) + \left.\frac{\partial f_1}{\partial y}\right|_{y=y_n} (y_{n+1} - y_n)$$

and

$$f_2(x_{n+1}, y_{n+1}) = f_2(x_n, y_n) + \left.\frac{\partial f_2}{\partial x}\right|_{x=x_n} (x_{n+1} - x_n) + \left.\frac{\partial f_2}{\partial y}\right|_{y=y_n} (y_{n+1} - y_n)$$

were the second term and third terms are the contributions due to x and y respectively, and

$$\partial f_1/\partial x,\ \partial f_2/\partial x,\ \partial f_2/\partial y,\ \partial f_2/\partial y$$

are the partial derivatives evaluated at $x = x_n, y = y_n$. Again if we suppose that x_{n+1}, y_{n+1} are very close to the root $(\hat{x}, \hat{y})$, then the above equations simplify to:

$$0 = f_1(x_n, y_n) + \left.\frac{\partial f_1}{\partial x}\right|_{x=x_n} (x_{n+1} - x_n) + \left.\frac{\partial f_1}{\partial y}\right|_{y=y_n} (y_{n+1} - y_n)$$

$$0 = f_2(x_n, y_n) + \left.\frac{\partial f_2}{\partial x}\right|_{x=x_n} (x_{n+1} - x_n) + \left.\frac{\partial f_2}{\partial y}\right|_{y=y_n} (y_{n+1} - y_n)$$

For simplicity let

$$x_{n+1} - x_n = h \qquad y_{n+1} - y_n = k$$

so that we can get the $(n+1)$th approximation by

$$x_{n+1} = x_n + h \qquad y_{n+1} = y_n + k$$

Now we need to find h and k. Substituting them back into the above equations and writing them in matrix form we get:

$$\begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} \begin{pmatrix} h \\ k \end{pmatrix} = - \begin{pmatrix} f_1(x_n, y_n) \\ f_2(x_n, y_n) \end{pmatrix}$$

where the matrix of partial derivatives is called the *Jacobian* and denoted by J or $J(x_n, y_n)$ to indicate its evaluation at (x_n, y_n). Therefore h and k can be found by

$$\begin{pmatrix} h \\ k \end{pmatrix} = -J^{-1}(x_n, y_n) \begin{pmatrix} f_1(x_n, y_n) \\ f_2(x_n, y_n) \end{pmatrix}$$

Note that h, k can also be found using Cramer's rule which requires computing the determinant of J and modifications of it.

If we denote by the column vector $\mathbf{x}$ the variables of the system, and by the column vector $\mathbf{f}$ the functions of the system then the general form of Newton's formula can be written as

$$\mathbf{x}_{n+1} = \mathbf{x}_n - J^{-1}(\mathbf{x}_n)\mathbf{f}(\mathbf{x}_n) \tag{7}$$

Observing that if $\hat{\mathbf{x}}$ is a stationary point of $\mathbf{f}(\mathbf{x})$ then $\hat{\mathbf{x}}$ is the root of the gradient/derivative of $\mathbf{f}(\mathbf{x})$, which means that we can use Newton's method to find stationary points provided that $\mathbf{f}(\mathbf{x})$ is twice differentiable. So for one variable Newton's formula can be written as

$$x_{n+1} = x_n + \frac{f'(x_n)}{f''(x_n)}$$

and generalized to several dimensions, can be written as

$$\mathbf{x}_{n+1} = \mathbf{x}_n - H^{-1}(\mathbf{x}_n)\nabla\mathbf{f}(\mathbf{x}_n) \tag{8}$$

where $H^{-1}(\mathbf{x})$ is the the *Hessian* matrix evaluated at $\mathbf{x}$, and $\nabla\mathbf{f}(\mathbf{x})$ is the gradient evaluated at $\mathbf{x}$.

References

[DHS00] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern classification, second edition*, Wiley, 2000.

[HO97] Aapo Hyvarinen and Erkki Oja, *A fast fixed-point algorithm for independent component analysis*, Neural Comput. **9** (1997), no. 7, 1483–1492.

[HO00] A. Hyvarinen and E. Oja, *Independent component analysis: algorithms and applications*, Neural Netw. **13** (2000), no. 4-5, 411–430.

[Hyv99a] A. Hyvarinen, *Fast and robust fixed point algorithms for independent component analysis*, IEEE Transactions on Neural Networks **10** (1999), no. 3, 626–634.

[Hyv99b] ———, *Survey on independent component analysis*, Neural Computing Surveys **2** (1999), 94–128.

[TK03] S. Theodoridis and K. Koutroumbas, *Pattern recognition 2nd ed*, Elsevier, 2003.