

ECE 7680

Lecture BSS – Application of Information theory to Blind Source Separation

Objective: To discuss the infomax criterion and apply it to the problem of blind source separation.

Homework ideas:

1. Given $f_X(x)$, determine a transformation g as in $Y = g(X)$ so that $Y \sim \mathcal{U}(0, 1)$.
2. Given that $X \sim \mathcal{U}(0, 1)$, determine a transformation g as in $Y = g(X)$ so that $f_Y(y)$ has some desired form.
3. Scalar case: If $y = g(x) = \frac{1}{1+e^{-u}}$, where $u = wx + w_0$, show that

$$\frac{\partial y}{\partial x} = wy(1 - y)$$

and that

$$\frac{\partial}{\partial w} \frac{\partial y}{\partial x} = y(1 - y)[1 + wx(1 - 2y)]$$

and that

$$\frac{\partial}{\partial w_0} \frac{\partial y}{\partial x} = wy(1 - y)x(1 - 2y).$$

Hence, explain the learning rule

$$\Delta w \propto \frac{1}{w} + x(1 - 2y)$$

and

$$\Delta w_0 \propto 1 - 2y.$$

4. If $y = g(x) = \tanh(wx + w_0)$, show that $\Delta w \propto \frac{1}{w} - 2xy$.
5. When $\mathbf{y} = g(W\mathbf{x} + \mathbf{w}_0)$, where g is the logistic function

$$g(u) = \frac{1}{1 + e^{-u}},$$

applied element-by-element, show that

$$\Delta W \propto [W^{-T}] + (\mathbf{1} - 2\mathbf{y})\mathbf{x}\mathbf{b}f^T$$

and

$$\Delta \mathbf{w}_0 \propto \mathbf{1} - 2\mathbf{y}.$$

6. For $\mathbf{x} = (x_1, \dots, x_n)$, define

$$I(\mathbf{x}) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{\prod_{i=1}^n p(x_i)} d\mathbf{x} = D(p(\mathbf{x}) || \prod_{i=1}^n p(x_i)).$$

Show that

$$H(\mathbf{x}) = \sum_{i=1}^n H(x_i) - I(\mathbf{x}).$$

This is a generalization of the formula (2.45) of the text.

Introduction

The principles of information theory can be applied to the blind source separation problem. We will briefly state the problem, then develop steps toward its solution.

Background and some preliminary results

We consider first the case of adapting a processing function g which operates on a scalar X using a function $Y = g(X)$ in order to maximize the mutual information between X and Y . That is, we assume that $g(X) = g(X; w, w_0)$ for some parameters w and w_0 , which are to be chosen to maximize $I(X; Y)$. We assume that g is a deterministic function. We have

$$I(X; Y) = H(Y) - H(Y|X).$$

But since g is deterministic, $H(Y|X) = H(g(X)|X) = 0$, so the mutual information is maximized when $H(Y)$ is maximized. (Actually, if we are dealing with differential entropy, this may not be the case. But we will take derivatives, and in any event $H(Y|X)$ is constant.)

Now, assuming the range of g is restricted (a reasonable assumption), what form should g be ideally? (the CDF of X). Draw a picture. Recall that

$$f_Y(y) = f_X(x) \left| \frac{dx}{dy} \right|_{x=g^{-1}(y)} = f_X(x) / |dy/dx|_{x=g^{-1}(y)}.$$

If $g(x) = F_X(x)$, then $dy/dx = f_X(x)$, and we get $f_Y(y) = 1$ (fill in some details).

Under the rule for transformations,

$$H(y) = -E[\ln f_Y(y)] = E[\ln \left| \frac{\partial y}{\partial x} \right|] - E[\ln f_X(x)].$$

But $f_X(x)$ does not depend on our parameters, so we can ignore it.

Of course, we may not know the pdf of X , and may not have the flexibility to choose. However, what is frequently done is to assume a particular functional form, and just fill in the parameters. Take

$$y = g(x) = \frac{1}{1 + e^{-u}}, \quad u = wx + w_0.$$

Then an adaptive scheme is to take

$$\Delta w \propto \partial H w = \frac{\partial}{\partial w} (\ln \left| \frac{\partial y}{\partial x} \right|) = \left(\frac{\partial y}{\partial x} \right)^{-1} \frac{\partial}{\partial w} \frac{\partial y}{\partial x}.$$

As examined in the HW, we find

$$\Delta w \propto \frac{1}{w} + x(1 - 2y)$$

Similarly, we find

$$\Delta w_0 \propto 1 - 2y.$$

We define by this means a weight update rule:

$$w^{[k+1]} = w^{[k]} + \mu_w \Delta w$$

$$w_0^{[k+1]} = w_0^{[k]} + \mu_0 \Delta w_0.$$

The effect of this learning rule is to drive Y to be as uniform as possible, then the form of g .

We can generalize this to N inputs and N outputs. Suppose we take

$$\mathbf{y} = g(W\mathbf{x} + \mathbf{w}_0),$$

where the function is applied element-by-element (expand out). Then

$$I(X;Y) = H(Y) - H(Y|X) = H(Y).$$

We want to determine W and \mathbf{w}_0 to maximize the joint entropy of the output, $H(\mathbf{y})$. W is a matrix, \mathbf{w}_0 is a vector. We have the pdf transformation equation

$$f_{\mathbf{y}}(\mathbf{y}) = f_{\mathbf{x}}(\mathbf{x})|J|^{-1},$$

where J is the Jacobian of the transformation,

$$J = \det \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \dots & \frac{\partial y_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial y_n}{\partial x_1} & \dots & \frac{\partial y_n}{\partial x_n} \end{bmatrix}.$$

Then, we before, we find

$$H(\mathbf{y}) = E \ln |J| - E \ln f_{\mathbf{x}}(\mathbf{x}),$$

where the second term does not depend upon the parameters. Then

$$\Delta W = \frac{\partial H(\mathbf{y})}{\partial W} = \frac{\partial}{\partial W} \ln |J|.$$

As explored in the homework,

$$\Delta W \propto [W^{-T}] + (\mathbf{1} - 2\mathbf{y})\mathbf{x}\mathbf{b}f^T \tag{1}$$

and similarly,

$$\Delta \mathbf{w}_0 \propto \mathbf{1} - 2\mathbf{y}.$$

BSS

Let $s_i(t), i = 1, 2, \dots, n$ be a set of statistically independent signals. We will later examine some other assumptions, but for now assume simply that they are independent. The signals are processed according to

$$\mathbf{x}(t) = A\mathbf{s}(t).$$

Now, not knowing either $\mathbf{s}(t)$ or A , we desire to determine a matrix W so that

$$\mathbf{y}(t) = W\mathbf{x}(t) = WAs(t)$$

recovers $\mathbf{s}(t)$ as fully as possible. Let us take as a criterion the mutual information at the output: $H(\mathbf{y})$. (Q: how did they know to try this? A: It seemed plausible, they tried it, and it worked! Moral: think about the implications of ideas, then see if it works.) Then, as shown in the exercises,

$$H(\mathbf{y}) = \sum_{i=1}^N H(y_i) - I(y_1, \dots, y_N).$$

If we maximize $H(\mathbf{y})$, we should (1) maximize each $H(y_i)$ and (2) minimize $I(y_1, \dots, y_N)$. As mentioned before, the $H(y_i)$ are maximized when (and if) the outputs are uniformly distributed. The mutual information is minimized when they are all independent! Achieving both of these exactly requires that g have the form of the CDF of s_i . So we might contemplate modifying W , and also modifying g . Or we might (as Bell and Sejnowski do) fix g , and don't worry about this. This corresponds to the assumption that $p(s_i)$ is super-Gaussian (heavier tails than a Gaussian has).

We can write

$$H(y_i) = -E[\log p(y_i)],$$

where we have

$$p(y_i) = p(u_i) / \left| \frac{\partial y_i}{\partial u_i} \right|$$

so that

$$H(y_i) = -E[\log p(u_i) / \left| \frac{\partial y_i}{\partial u_i} \right|]$$

Thus

$$H(\mathbf{y}) = -\sum_{i=1}^N E[\log p(u_i) / \left| \frac{\partial y_i}{\partial u_i} \right|] - I(\mathbf{y})$$

Then

$$\frac{\partial H(\mathbf{y})}{\partial W} = \frac{\partial -I(\mathbf{y})}{\partial W} - \frac{\partial}{\partial W} \sum_{i=1}^N E[\log p(u_i) / \left| \frac{\partial y_i}{\partial u_i} \right|]$$

In the case that $p(u_i) = \left| \frac{\partial y_i}{\partial u_i} \right|$, then the last stuff goes away. In other words, we ideally want $y_i = g_i(u_i)$ to be the CDF of the u_i . When this is not exactly the case (there is a mismatch), then the last term exists and may interfere with the minimization of $I(\mathbf{y})$. We call the term $\frac{\partial}{\partial W} \sum_{i=1}^N E[\log p(u_i) / \left| \frac{\partial y_i}{\partial u_i} \right|]$ and “error term”. Now we note that

$$H(\mathbf{y}) = -E[\log p(\mathbf{y})] = -E[\log p(\mathbf{x}) / |J(\mathbf{x})|] = -E[\log p(\mathbf{x})] + E[\log |J(\mathbf{x})|].$$

The term $-E[\log p(\mathbf{x})]$ does not depend upon W , so we obtain

$$\frac{\partial H(\mathbf{y})}{\partial W} = \frac{\partial}{\partial W} E[\log |J(\mathbf{x})|].$$

Now we come to an important concept: We would like to compute the derivative, but can't compute the expectation. We make the **stochastic gradient approximation**: $E[\log |J(\mathbf{x})|] \approx \log |J(\cdot)|$. We just throw the expectation away! Does it work? On average!

Now it becomes a matter of grinding through the calculus to take the appropriate partial derivative. Since

$$J(\mathbf{x}) = \det \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \dots & \frac{\partial y_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial y_n}{\partial x_1} & \dots & \frac{\partial y_n}{\partial x_n} \end{bmatrix}$$

we will consider the elements:

$$\frac{\partial y_i}{\partial x_j} = \frac{\partial y_i}{\partial u_i} \frac{\partial u_i}{\partial x_j} = w_{ij} \frac{\partial y_i}{\partial u_j}$$

since $\mathbf{u} = W\mathbf{x}$, and $y_i = g(u_i)$. Because this connection, the partial $\frac{\partial y_i}{\partial u_j}$ is nonzero only when $i = j$. Combining these facts, we find

$$J(\mathbf{x}) = \det(W) \prod_{i=1}^N \left| \frac{\partial y_i}{\partial u_i} \right|$$

Thus

$$\begin{aligned} \frac{\partial H(\mathbf{y})}{\partial W} &= \frac{\partial}{\partial W} \log \left(|\det(W)| \prod_{i=1}^N \left| \frac{\partial y_i}{\partial u_i} \right| \right) = \frac{\partial}{\partial W} \log \det |W| + \frac{\partial}{\partial W} \sum_{i=1}^N \log \left| \frac{\partial y_i}{\partial u_i} \right| \\ &= W^{-T} + \sum_{i=1}^N \frac{\partial}{\partial W} \log \left| \frac{\partial y_i}{\partial u_i} \right|. \end{aligned}$$

(See appdx E of Moon and Stirling.) Looking at the second term,

$$\frac{\partial}{\partial w_{ij}} \sum_{k=1}^N \log |\partial y_k u_k| = \sum_k 1 / \left(\frac{\partial y_k}{\partial u_k} \right) \frac{\partial}{\partial w_{ij}} \frac{\partial y_k}{\partial u_k} = \sum_k 1 / \left(\frac{\partial y_k}{\partial u_k} \right) \frac{\partial}{\partial u_i} \frac{\partial u_i}{\partial w_{ij}} \frac{\partial y_k}{\partial u_k} = 1 / \left(\frac{\partial y_i}{\partial u_i} \right) \frac{\partial^2 y_i}{\partial u_i^2} x_j$$

since $\frac{\partial u_i}{\partial w_{ij}} = x_j$.

Let us write

$$p(u_i) = \frac{\partial y_i}{\partial u_i}.$$

This looks like a density, and ideally would be so, as discussed above. But we can think of this as simply a function. We thus find, stacking all the results,

$$\frac{\partial}{\partial W} \sum_{i=1}^N \log \left| \frac{\partial y_i}{\partial u_i} \right| = \frac{\partial p(\mathbf{u})}{p(\mathbf{u})} \mathbf{x}^T.$$

This gives us the learning rule:

$$\frac{\partial H(\mathbf{y})}{\partial W} = W^{-T} + \left(\frac{\partial p(\mathbf{u})}{p(\mathbf{u})} \right) \mathbf{x}^T.$$

We will let

$$\psi(\mathbf{u}) = - \frac{\partial p(\mathbf{u})}{p(\mathbf{u})}$$

be the learning nonlinearity, also called in the literature the score function. Then

$$\frac{\partial H(\mathbf{y})}{\partial W} = W^{-T} - \psi(\mathbf{u}) \mathbf{x}^T.$$

Example 1 Let

$$y = g(u) = \frac{1}{1 + e^{-u}}$$

Then

$$p(u) = \frac{\partial y}{\partial u} = y(1-y) \quad \partial p u = y(1-y)^2 - y^2(1-y)$$

and

$$\psi(u) = 1 - 2y.$$

We thus obtain the weight update rule

$$\frac{\partial H(\mathbf{y})}{\partial W} = W^{-T} + (\mathbf{1} - 2\mathbf{y})\mathbf{x}^T.$$

□

Example 2 If $g(u) = \tanh(u)$, then $\phi(u) = 2 \tanh(u)$. □

This approach can only separate super-Gaussian distributions (heavy tails).

Mackay's approach

We consider the BSS problem as a ML estimation problem. To begin with, we state a lemma that we will need.

Lemma 1 Let $\mathbf{x} = A\mathbf{s}$, so that $x_j = \sum_i v_{ji}s_i$. Assume that A is invertible. Then

$$\int \prod_j \delta(x_j - \sum_i v_{ji}s_i) \prod_j p_j(s_j) ds = \frac{1}{\det(A)} \prod_j p_j(\sum_i (A^{-1})_{ji}x_i)$$

Proof We shall give an explicit proof for the 2×2 case. Explicitly, we have

$$I = \int \left[\int \delta(x_1 - a_{11}s_1 - a_{12}s_2) \delta(x_2 - a_{21}s_1 - a_{22}s_2) p_1(s_1) p_2(s_2) ds_1 \right] ds_2.$$

In the inner integral, we must have $s_1 = \frac{1}{a_{11}}(x_1 - a_{12}s_2)$, and we get a factor of $\frac{1}{a_{11}}$. (Recall that by the definition of the δ function,

$$\int \delta(x - vs) f(s) ds = \frac{1}{v} f(x/v).$$

So

$$I = \frac{1}{a_{11}} \int \delta(x_2 - \frac{a_{21}}{a_{11}}x_1 - s_2(\frac{a_{22}a_{11} - a_{12}a_{21}}{a_{11}})) p_1(\frac{1}{a_{11}}(x_1 - a_{12}s_2)) p_2(s_2) ds_2$$

Solving for s_2 in the argument of the δ function, we must have

$$s_2 = \frac{a_{11}x_2 - a_{21}x_1}{a_{11}a_{22} - a_{12}a_{21}}.$$

Now evaluating the integral and substituting for this value of s_2 we find

$$I = \frac{1}{a_{11}} \frac{a_{11}}{a_{22}a_{11} - a_{12}a_{21}} p_1(\frac{a_{22}x_1 - a_{12}x_2}{a_{22}a_{11} - a_{12}a_{21}}) p_2(\frac{a_{11}x_2 - a_{21}x_1}{a_{22}a_{11} - a_{12}a_{21}})$$

from which the result is apparent for this case. □

Given a sequence of observed data $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ obtained from $\mathbf{x} = A\mathbf{s}$ we write down the joint probability as

$$P(\mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{s}_1, \dots, \mathbf{s}_N | A) = \prod_{n=1}^N P(\mathbf{x}_n | \mathbf{s}_n, A) P(\mathbf{s}_n) = \prod_{n=1}^N \left[\prod_j \delta(\mathbf{x}_{n,j} - \sum_i a_{ji} \mathbf{s}_{n,i}) \prod_j p_j(\mathbf{s}_{n,j}) \right].$$

To do ML estimation of A we examine

$$P(\mathbf{x}_1, \dots, \mathbf{x}_N | A) = \prod_n P(\mathbf{x}_n | A),$$

The ML solution is to find the A that maximizes this likelihood function. We have

$$\begin{aligned} P(\mathbf{x}_n | A) &= \int P(\mathbf{x}_n | \mathbf{s}_n, A) P(\mathbf{s}_n) d\mathbf{s}_n \\ &= \int \prod_j \delta(\mathbf{x}_{n,j} - \sum_i a_{ji} \mathbf{s}_{n,i}) \prod_j p_j(\mathbf{s}_{n,j}) d\mathbf{s}_n \\ &= \frac{1}{\det(A)} \prod_j p_j(\sum_i (A^{-1})_{ji} \mathbf{x}_{n,i}), \end{aligned}$$

where the lemma we derived before has been used. The log likelihood function is

$$\log P(\mathbf{x}_n | A) = -\log \det A + \sum_j \log p_j(\sum_i (A^{-1})_{ji} \mathbf{x}_{n,i}).$$

Let $W = A^{-1}$. Then

$$\log P(\mathbf{x}_n | A) = \log \det W + \sum_j \log p_j(\sum_i W_{ji} \mathbf{x}_{n,i}).$$

Now we proceed as before, computing the derivative of the log likelihood with respect to W . The first part is easy:

$$\frac{\partial}{\partial W} \log \det W = W^{-T}.$$

For the first part:

$$\begin{aligned} \frac{\partial}{\partial W_{mn}} \sum_j \log p_j(\sum_i W_{ji} x_i) &= \sum_j \frac{1}{p_j(\sum_i W_{ji} x_i)} \frac{\partial}{\partial W_{mn}} p_j(\sum_i W_{ji} x_i) \\ &= \frac{1}{p_m(\sum_i W_{mi} x_i)} \frac{\partial}{\partial W_{mn}} p_m(\sum_i W_{mi} x_i) \\ &= \frac{1}{p_m(a_m)} \frac{\partial p_m(a_m)}{\partial a_m} \frac{\partial a_m}{\partial W_{mn}} \quad (a_m = \sum_i W_{mi} x_i) \\ &= \frac{1}{p_m(a_m)} x_n \frac{\partial p_m(a_m)}{\partial a_m} \end{aligned}$$

Let

$$\phi_m(a_m) = \frac{d}{da_m} \log p_m(a_m) = \frac{1}{p_m(a_m)} \frac{dp_m(a_m)}{da_m}$$

and let $z_m = \phi_m(a_m)$. Then we have

$$\frac{\partial}{\partial W_{mn}} \sum_j \log p_j(\sum_i W_{ji} x_i) = x_n z_m.$$

Thus

$$\frac{\partial}{\partial W} \log P(\mathbf{x}_n | A) = W^{-T} + \mathbf{z} \mathbf{x}^T.$$

Compare with what we had before!

Natural Gradient

The training law we have developed up to this point requires computation of W^{-T} . We can modify this by

$$\Delta W \propto \frac{\partial H(\mathbf{y})}{\partial W} W^T W.$$

This becomes (since $\mathbf{u} = W\mathbf{x}$)

$$\Delta W \propto (I - \phi(\mathbf{u})\mathbf{u}^T)W.$$

Example 3 With the natural gradient, the weight update for the logistic function becomes

$$(I - 2\mathbf{y}\mathbf{u}^T)W$$

□

This modification to the gradient, multiplying by $W^T W$ is called the *natural gradient* (Amari, 1998). In this section, we examine this, with any eye to the question: what is natural about it?

Comment on scaling of update formula.

We follow Amari 1998 in the following discussion. Suppose $S = \{\mathbf{w} \in \mathbb{R}^n\}$ is some parameter space (e.g., the space of parameters in the weighting matrix). Suppose there is some function $L(\mathbf{w})$ defined. Consider a parameter value \mathbf{w} , and some incremental change to $\mathbf{w} + d\mathbf{w}$. If the parameter space is *Euclidean*, then the length of the increment is

$$\|d\mathbf{w}\|^2 = \sum_{i=1}^n (dw_i)^2.$$

However, not all parameter spaces are Euclidean. Consider, for example, a case where the parameters all lie on a sphere. Then the appropriate distance measure is not simply the sum of the squares of the coordinates, especially if \mathbf{w} is measured in spherical coordinates! So we measure the change differently:

$$\|d\mathbf{w}\|^2 = \sum_{i,j} g_{ij}(\mathbf{w}) dw_i dw_j.$$

Here, g is called the **Riemannian metric tensor**; it describes the local curvature of the parameter space at the point \mathbf{w} . In terms of vectors, we can write

$$\|d\mathbf{w}\|^2 = \mathbf{w}^T G \mathbf{w},$$

where $G = G(\mathbf{w})$ (a function of \mathbf{w}). G is symmetric. We see that we are simply dealing with a weighted distance, induced from a weighted inner product, defined by

$$\langle \mathbf{x}, \mathbf{y} \rangle_G = \mathbf{y}^T G \mathbf{x}.$$

When $G(\mathbf{w}) = I$, we simply get the Euclidean distance.

Now consider the problem of learning by “steepest descent.” The question is, do we really go in the right direction, if we take into account the curvature of the parameter space. We want to decrease $L(\mathbf{w})$ by moving in a direction $d\mathbf{w}$ to obtain $L(\mathbf{w} + d\mathbf{w})$, and do the best possible job with the motion. Let us assume that we have a fixed step length,

$$\|d\mathbf{w}\|^2 = \epsilon^2$$

for some small positive ϵ .

Theorem 1 The steepest descent direction of $L(\mathbf{w})$ in a Riemannian space with metric tensor G is

$$-\tilde{\nabla}L(\mathbf{w}) = -G^{-1}(\mathbf{w})\nabla L(\mathbf{w}),$$

where G^{-1} is the inverse of G , and ∇L is the conventional gradient,

$$\nabla L(\mathbf{w}) = \begin{bmatrix} \frac{\partial L(\mathbf{w})}{\partial w_1} \\ \frac{\partial L(\mathbf{w})}{\partial w_2} \\ \vdots \\ \frac{\partial L(\mathbf{w})}{\partial w_n} \end{bmatrix}$$

Observe that the usual “steepest descent” that we deal with always assumes that $G = I$.

Proof Let \mathbf{a} be a unit vector (under the Riemannian metric), so

$$\mathbf{a}G\mathbf{a} = 1,$$

and let $d\mathbf{w} = \epsilon\mathbf{a}$. We want to find \mathbf{a} to minimize

$$L(\mathbf{w} + d\mathbf{w}) = L(\mathbf{w} + \epsilon\mathbf{a}).$$

Expanding L in the first two terms of a Taylor series (generalizing on a scalar Taylor series) we find

$$L(\mathbf{w} + \epsilon\mathbf{a}) = L(\mathbf{w}) + \epsilon\Delta L(\mathbf{w})^T\mathbf{a},$$

where $\Delta L(\mathbf{w})$ is the conventional gradient. To minimize L under the constraint, we set up the problem

$$J(\mathbf{a}) = L(\mathbf{w}) + \epsilon\Delta L(\mathbf{w})^T\mathbf{a} - \frac{\lambda}{2}\mathbf{a}^T G\mathbf{a}.$$

Then

$$\frac{\partial J(\mathbf{a})}{\partial \mathbf{a}} = \epsilon\nabla L(\mathbf{w}) - \lambda G\mathbf{a} = 0,$$

leading to

$$\mathbf{a} = \frac{\epsilon}{\lambda}G^{-1}\nabla L(\mathbf{w}).$$

Then λ is chosen to normalize \mathbf{a} (without changing its direction). \square

We call

$$\tilde{\nabla}L(\mathbf{w}) = G^{-1}\Delta L(\mathbf{w})$$

the *natural gradient* of L in the Riemannian space. In Euclidean space, it is the same as the usual gradient.

Now consider the BSS problem in the context of natural gradient. We first formulate the problem. We have, as before, signal vectors $\mathbf{s}(t)$ with independent components, so that

$$p(\mathbf{s}) = \prod_{i=1}^n p_i(s_i)$$

and $\mathbf{x}(t) = A\mathbf{s}(t)$. The output is

$$\mathbf{y}(t) = W_t \mathbf{x}(t),$$

and we update the matrix by some learning rule

$$W_{t+1} = W_t - \eta_t F(\mathbf{x}, W_t).$$

Previously, we took the learning update to be $F(\mathbf{x}, W_t) = \frac{\partial}{\partial W} H(\mathbf{y})$, but this will now change.

We observe that in order to obtain equilibrium, the function F must satisfy

$$E[F(\mathbf{x}, W)] = 0 \quad (2)$$

when $W = A^{-1}$ (we stop changing at the correct answer). Now let $K(W): \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ be an operator that maps a matrix to a matrix, and let

$$\tilde{F}(\mathbf{x}, W) = K(W)F(\mathbf{x}, W).$$

Then \tilde{F} satisfies (2) when F does (same equilibrium). We want to determine what form the transformation should take.

Let dW be a small deviation from a matrix W to $W + dW$. dW constitutes a “vector” starting from the point W . Let us define an inner product at W as

$$ds^2 = \text{squared length of the vector at } W = \langle dW, dW \rangle_W = \|dW\|^2.$$

(Draw a picture of a curved W surface, and the vector on it.) We can pull back the point, mapping to another surface, by right-multiplying by W^{-1} . Then W maps to I , and $W + dW$ maps to

$$I + dX$$

where

$$dX = dW W^{-1}.$$

A deviation dW at W is equivalent to the deviation dX at I by this mapping. The key idea is that we want the metric to be invariant under this mapping: the inner product of dW at W is to be the same as the inner product of dWY at WY for any Y . Thus we impose the invariant

$$\langle dW, dW \rangle_W = \langle dWY, dWY \rangle_{WY}$$

In particular, when $Y = W^{-1}$, we have $WY = I$. We define the inner product at I by

$$\langle dX, dX \rangle_I = \sum_{i,j} (dX_{ij})^2 = \text{tr}(dX^T dX),$$

the (unweighted, Euclidean) Frobenius norm. Under our principle of equivalence (using $dX = dW W^{-1}$), we should therefore have

$$\langle dW, dW \rangle_W = \langle dX, dX \rangle_I = \text{tr}(dX^T dX) = \text{tr}(W^{-T} dW^T dW W^{-1}) = \sum_{i,j,k,l} G_{ij,kl}(W) dW_{ij} dW_{kl}.$$

It follows that the Riemannian tensor has the form

$$G_{ij,kl} = \sum_m \delta_{ik} (W^{-1})_{jm} (W^{-1})_{lm}.$$

We can determine an explicit form for the natural gradient using the principle of invariance. We interpret $\tilde{\nabla}f(W)$ as a vector applied at W , and $\nabla f(W)$ as a vector applied at I . Then we must have

$$\langle \tilde{\nabla}f(W), dW \rangle_W = \langle \tilde{\nabla}f(W)W^{-1}, dWW^{-1} \rangle_{WW^{-1}} \triangleq \langle \nabla f(W), \Delta W \rangle_I$$

We thus have (using the definition of the inner product)

$$\text{tr}(W^{-T} \tilde{\nabla}f(W)^T dWW^{-1}) \triangleq \text{tr}(\nabla f(W)^T dW).$$

Using the commuting properties of trace we find

$$\text{tr}(W^{-1}W^{-T} \tilde{\nabla}f(W)^T dW) = \text{tr}(\nabla f(W)^T dW)$$

$$\text{tr}[(W^{-1}W^{-T} \tilde{\nabla}f(W)^T - \nabla f(W)^T) dW] = 0.$$

Since this must be true for arbitrary dW , we must have

$$(W^{-1}W^{-T} \tilde{\nabla}f(W)^T = \nabla f(W)^T$$

or

$$\tilde{\nabla}f(W) = \nabla f(W)W^T W^{-1}$$

So what about $p(u)$

We have observed that if $p(u) = \left| \frac{\partial g}{\partial u} \right|$, then the last terms of the density go away. We also commented above that the given form works only for super-Gaussian sources. So what do we do in the case where this is not a good assumption. One approach is to deal with two densities.

On the one hand, we take

$$p(u) = \frac{1}{2}(N(\mu, \sigma^2) + N(-\mu, \sigma^2))$$

giving (where $a = \mu/\sigma^2$)

$$\phi(u) = \frac{u}{\sigma^2} - a \left(\frac{\exp(au) - \exp(-au)}{\exp(au) + \exp(-au)} \right) = \mu/\sigma^2 - \mu/\sigma^2 \tanh(\mu u/\sigma^2).$$

This p is subGaussian. When $\mu = 1$ and $\sigma^2 = 1$ we obtain

$$\phi(u) = -\tanh(u),$$

and the learning rule (employing natural gradient) is

$$\Delta W \propto [I + \tanh(\mathbf{u})\mathbf{u}^T - \mathbf{u}\mathbf{u}^T]W.$$

The superGaussian density is modeled as

$$p(u) \propto [\mathcal{N}(0, 1)] \text{sech}^2(u)$$

giving rise to

$$\phi(u) = u + \tanh(u)$$

and the learning rule

$$\Delta W \propto [I - \tanh(\mathbf{u})\mathbf{u}^T - \mathbf{u}\mathbf{u}^T]W.$$

Combining these, we obtain

$$\Delta W \propto \begin{cases} (I - \tanh(\mathbf{u})\mathbf{u}^T - \mathbf{u}\mathbf{u}^T)W & \text{super-Gaussian} \\ (I + \tanh(\mathbf{u})\mathbf{u}^T - \mathbf{u}\mathbf{u}^T)W & \text{sub-Gaussian} \end{cases}$$

The decision between rules can be made on an element-by-element basis. A decision must be based on the available data.