An Information-Maximization Approach to Blind Separation and Blind Deconvolution

paper by Anthony J. Bell &
Terrence J. Sejnowski

Presenter : Jarkko Salojärvi

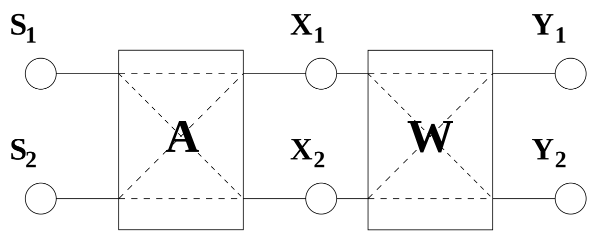
The 'Beef' of the Paper : Infomax

- Ch 1: Brief introduction
- Ch 2: Information Maximization (*important*)
  - Ch 2.1 - 2.2 One Input and One Output ; N $\rightarrow$ N Network
  - Ch 2.3 - 2.4 Extensions
  - Ch 2.5 Generalized Sigmoid Function
- Ch 3: Background (*you should already know this*)
- Ch 4: How Does the Information Maximization Reduce Statistical Dependence ? (*only heuristic but important*)
- Ch 5: Test Results (*if you want to replicate the experiments*)
- Ch 6: Discussion; Comparison with Linsker, Conclusion (*the limitations of the algorithm*)

Structure of the Presentation

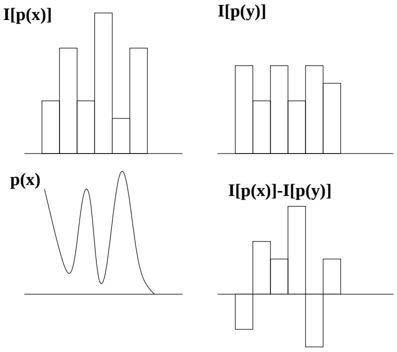
- Information Maximixation (a.k.a. *Infomax*)
- Application of Infomax to an N$\rightarrow$N Network
- Experiments
- Conclusions

Problem



$$\mathbf{y=Wx}$$

Difference between two distributions



The difference of information between two distributions can be computed by

$$\Delta I = -\sum_i \log p_X(x_i) - \log p_Y(y_i) = -\sum_i \log \frac{p_X(x_i)}{p_Y(y_i)}.$$

Now the *mean* difference between the information distributions with respect to $p_X(x)$ is

$$-\sum_i p_X(x_i) \log \frac{p_X(x_i)}{p_Y(y_i)} = D(p_X(X) \| p_Y(Y)),$$

which is the **Kullback-Leibler distance**.

What shall we do with the KL distance ?

If two random variables are independent, their mutual distribution is the same as the product of their marginal distributions,

$$p(X,Y) = p(X)p(Y).$$

Therefore the Kullback-Leibler distance between them must be zero, that is

$$D(p(X,Y) \| p(X)p(Y)) = \sum_{x,y} p(X,Y) \log \frac{p(X,Y)}{p(X)p(Y)} = 0.$$

Vice versa, we can find the maximal mutual distribution by maximizing this value.

We can write the KL distance also in another way

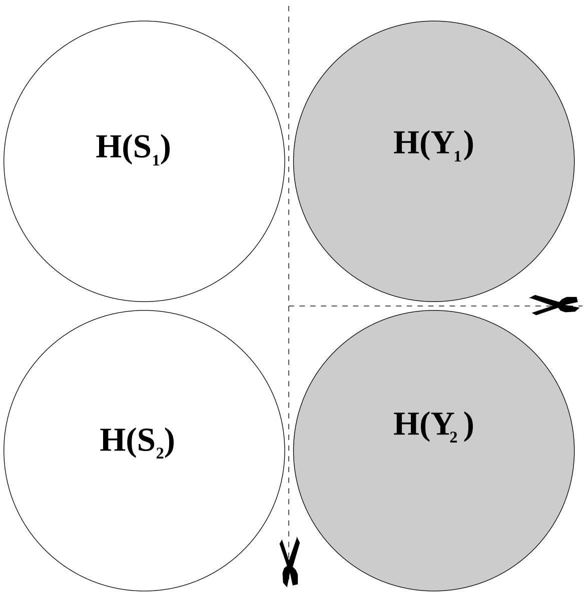
$$\sum_{x,y} p(X,Y) \log \frac{p(Y|X)}{p(Y)} =$$

$$\sum_{x,y} p(X,Y) \log p(Y|X) - \sum_y p(Y) \log p(Y) =$$

$$H(Y) - H(Y|X) = I(X,Y),$$

which is the **mutual information** between the two distributions.

'The Infomax Machine' - intuitive proof



Infomax

The idea of infomax is to maximize the mutual information between the input X and the output Y of the network, that is,

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

If we want to maximize this, we must maximize $H(Y)$ and minimize $H(Y|X)$.

If we assume that $Y = G(X) + N$, the $H(Y|X)$ term is noise $H(N)$, because it is the amount of uncertainty of $H(Y)$ that doesn't depend on the sources.

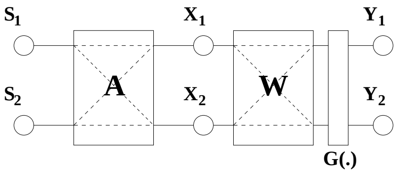
We maximize mutual information by gradient ascent. The gradient with respect to the parameters we use in our model is

$$\frac{\partial}{\partial \omega} I(X,Y) = \frac{\partial}{\partial \omega} H(Y) = \frac{\partial}{\partial \omega} H(G(\mathbf{W}(\omega)\mathbf{x})),$$

because $H(Y|X)$ doesn't depend on the parameters ω.

Neural Network Implementation of Information Maximization : revised problem setting

The complete problem with neural networks may be illustrated as

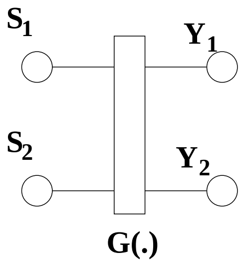


y=G(Wx)+N

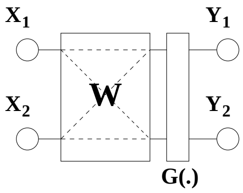
where we have added $G(.)$, which is a nonlinear **squashing function**. Usually (but not always) this is a sigmoid function.

Neural Network Implementation of Information Maximization : what it does ?

The idea for maximizing the information between input and output is that



should produce the same distribution as



Neural Network Implementation of Information Maximization : what it does ?

If the squashing function $G(.)$ is a cumulative distribution function of $p(s_n)$, the actual source signal s_n is transformed to uniform distribution. In reality, we don't know the optimal $G(.)$. Instead, we have another $\tilde{s}$ that produces the uniform distribution.

The maximal mutual information can thus be interpreted as minimization of the Kullback-Leibler distance between $G(Wx)$ and uniform distribution $U(\tilde{S})$.

$$\min D(G(Wx)\|U(\tilde{S})) = D(Wx\|\tilde{s}),$$

where the last equality hold because KL is invariant under an invertible transformation.

Realization : N → N network

The update rules with sigmoid squashing function were derived in the paper in chapter 2.2. and appendix. Therefore there's no need to repeat it here.

The gradient ascent with stochastic approximation gives

$$\begin{aligned}\mathbf{W}^{(i+1)} &= \mathbf{W}^{(i)} + \lambda(i)\left([\mathbf{W}_{(i)}^T]^{-1} + (\mathbf{1} - 2\mathbf{y}^{(i)})\mathbf{x}_{(i)}^T\right)\\ \mathbf{w}_0^{(i+1)} &= \mathbf{w}_0^{(i)} + \lambda(i)\left(\mathbf{1} - 2\mathbf{y}^{(i)}\mathbf{x}_{(i)}^T\right)\end{aligned}$$

as update rules, where

- $\mathbf{W}$ Weights of the network
- $\mathbf{w}_0$ Bias of the neurons
- λ Learning rate
- $\mathbf{y}$ Output of the network
- $\mathbf{x}$ Input of the network

Test Case : N → N network

For the presentation, I performed two experiments with

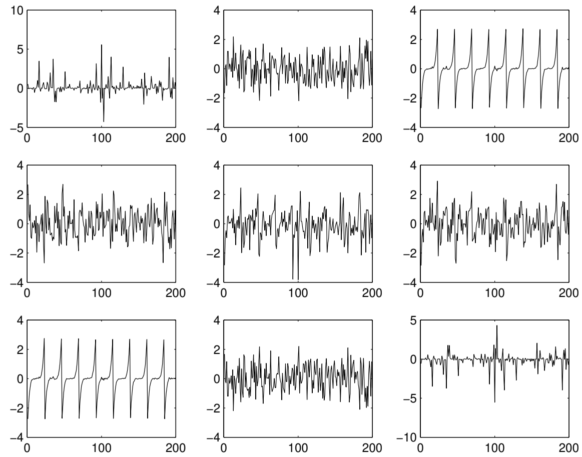
a) Three independent components, two supergaussians and one gaussian.

b) Three independent components, two supergaussians and two subgaussians

Both had a sigmoid as squashing function.

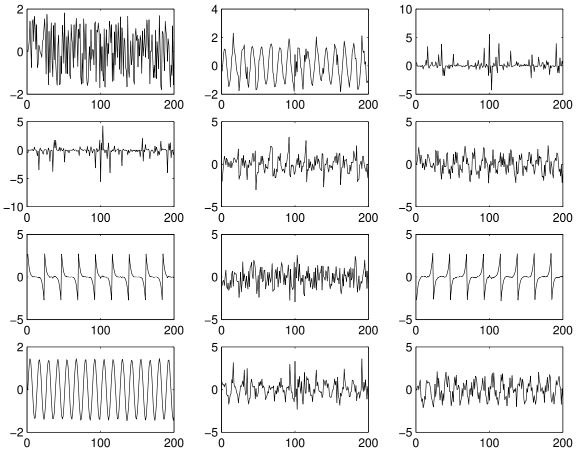
In the pre-processing stage the data vectors were whitened.

Results : 3 Independent components

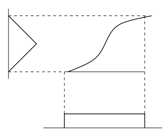


Success !!

Results : 4 Independent components



The subgaussian components do not separate, because the sigmoid function can't approximate the cumulative distribution function of the subgaussians ⇒ Generalised Sigmoid.



Discussion and further reading

Differences from Linsker's model

- No noise model.
- No assumptions of gaussianity in inputs or outputs
- Nonlinear transfer function.

Weaknesses

- ICA without adaptive time delays doesn't work in natural environment.
- Nonlocal.

Future : Addition of layers, so that the optimal mappings can be also nonlinear.

Reading

- If there is noise ? (Nadal and Parga, 1994)
- Why does the infomax approach work ? (Cardoso, 1997 (ref in the ICA tutorial))