# Multi-Channel Subspace Mapping Using An Information Maximization Criterion

Ahmed Al-Ani[1] and Mohamed Deriche[2]

[1] School of Computer Science & Software Engineering
The University of Western Australia
35 Stirling Highway, Crawley, Western Australia, 6009, Australia

[2] Department of Electrical Engineering
King Fahd University of Petroleum and Minerals
Dhahran-31261, Saudi Arabia

**Abstract**

A new hybrid information maximization (HIM) algorithm is derived. This algorithm is able to perform subspace mapping of multi-channel signals, where the input (feature) vector for each of the channels is linearly transformed to an output vector. The algorithm is based on maximizing the mutual information (MI) between input and output sets for each of the channels, and between output sets across channels. Such formulation leads to a substantial redundancy reduction in the output sets, and the extraction of higher order features that exhibit coherence across time and/or space. In this paper, we develop the proposed algorithm and show that it combines efficiently the strengths of two well-known subspace mapping techniques, namely the principal component analysis (PCA) and the canonical correlation analysis (CCA). Unlike CCA, which is limited to two channels, the HIM algorithm can easily be extended to multiple channels. A number of simulations and real experiments are conducted to compare the performance of HIM to that of PCA and CCA.

## 1 Introduction

There is an increased interest in the processing of multi-channel signals, as such signals are widely used in various fields, such as aerospace, automation, biomedical engineering, robotics, and human

computer interfaces. Such signals are generally collected using a number of sensors that can be distributed spatially, logically, or even geographically. Examples of multi-channel signals include: EEG signals, which normally have more than 5 channels, and microphone array speech signals that are collected through an array of four or more microphones, to mention a few. One area that has attracted a lot of attention among researchers is that of reducing dimensionality of collected data. Such problem is challenging by itself, but becomes, obviously, more complex when data is collected through multiple channels. A famous approach used for reducing dimensionality in signals is subspace mapping, which is defined as the process that transforms an input set of dimension $N$ to an output set of dimension $M$, with $M < N$. The input could be either a raw data vector or a high dimensional feature vector. The goal of projecting the original $N$-dimensional vector onto a new $M$-dimensional subspace is to get a more efficient and compact representation of the original elements. It has been shown that using subspace mapping to process single-channel signals could provide better results for further analysis or classification stages [1]. This is because subspace mapping can help in removing redundant information, and hence leading to a better generalization on "unseen" data. Processing multi-channel signals is by far more complex, and in fact, optimal subspace mapping for multi-channel signals is still a challenging problem. This difficulty arises because of the need to extract useful information from the data set of each of the channels taking into account coherence across different channels.

A simple form of single-channel subspace mapping can be formulated as follows:

$$\mathbf{p} = \mathbf{W}^T \mathbf{x} \tag{1}$$

where $\mathbf{x}$ and $\mathbf{p}$ are random vectors with zero mean and finite covariance, representing respectively the input and output sets for a given channel. The size of the input vector $\mathbf{x}$ is $N \times 1$, while that of the output vector $\mathbf{p}$ is $M \times 1$, with $M < N$. $\mathbf{W}$ is a transformation matrix that maps the $N$-dimensional input space into $M$-dimensional output space, *i.e.*, its size is $N \times M$. Extending this form to $K$ channels leads to:

$$\begin{aligned} \mathbf{p}_1 &= \mathbf{W}_1^T \mathbf{x}_1 \\ \vdots &\quad \vdots \\ \mathbf{p}_K &= \mathbf{W}_K^T \mathbf{x}_K \end{aligned} \tag{2}$$

One of the most established single-channel subspace mapping techniques is principal component analysis (PCA), which maps a correlated input set into an uncorrelated output set. On the other hand, given two input sets, the canonical correlation analysis (CCA) aims at transforming these (simultaneously) such that the correlation between their corresponding output sets is maximized.

Several methods have been proposed to implement PCA and CCA iteratively [2, 3, 4, 5, 6]. The purpose of developing such methods is to investigate issues that can not be explored with analytical PCA and CCA. For example, the elements of the output set could be non-orthogonal rotated versions of those found by the analytical methods, or the output subspace could be non-linear [7].

In this work, we aim at developing an algorithm that combines the strengths of both PCA and CCA. Such combination will be beneficial since CCA does not guarantee uncorrelation within the output set of each of the channels, while PCA processes each channel independently. The advantage of such combination becomes especially useful if it can be extended to more than two channels. Motivated by the *infomax* principle [8], which maximizes MI between the input and output sets of a given channel, and the *Imax* algorithm [9] that maximizes MI between the output sets of two channels, this paper proposes a hybrid information maximization (HIM) algorithm that aims at reducing redundancy within the output set of each channel, and maximizing coherence across the output sets of two or more channels using the concept of information content. Derivations for the cases of two and three channels will be given, then we discuss different options for extending the HIM to four or more channels. The performance of the HIM algorithm will be compared to that of the PCA and CCA techniques.

Basic concepts of the PCA, CCA, infomax and Imax algorithms are presented in section 2. Derivations of the HIM algorithm are given in section 3. Section 4 presents the experimental results including performance using synthetic and real data, optimal choice of learning rate, and comparison with PCA and CCA. A conclusion is given in section 5.

# 2 Background

## 2.1 Principal Component Analysis

One of the simplest and most popular techniques in feature extraction and data compression is the statistical method of principal component analysis (PCA) [1].

Let us assume that the input set $\mathbf{x}$ has $N$ elements. To capture the main features of $\mathbf{x}$, PCA looks for directions along which the variance is maximal. If $\lambda_1 > \cdots > \lambda_N > 0$ are the eigenvalues of the covariance matrix, $\mathbf{R_{xx}}$, with eigenvectors $\mathbf{a}_1, \cdots, \mathbf{a}_N$, then the first feature PCA extracts, is the principal component $\mathbf{a}_1^T\mathbf{x}$. This component is the projection onto the line spanned by $\mathbf{a}_1$. PCA then looks for lines perpendicular to the line spanned by $\mathbf{a}_1$, such that the projection of $\mathbf{x}$ onto the original component of $\mathbf{a}_1$ has maximum variance, and so forth. Hence, to extract $M$ components, PCA extracts the first $M$ principal components $\mathbf{a}_1^T\mathbf{x}, \cdots, \mathbf{a}_M^T\mathbf{x}$, and reduces $\mathbf{x}$ to

its projection onto the hyperplane spanned by the first $M$ eigenvectors. In contrast with PCA, which only ensures that output variables are uncorrelated, independent component analysis (ICA) imposes the stronger criterion that the multivariate probability density function (p.d.f.) of output elements factorizes [10, 11]. Finding such a factorization requires that the mutual information between all variable pairs goes to zero. Mutual information depends on all higher-order statistics of the output variables while decorrelation caters for 2nd-order statistics only.

## 2.2 The Infomax Principle

Based on the more comprehensive representation of data using "information content", Linsker [8] developed the infomax principle, which was inspired by Hebb's rule. The rule says that if $x_i$ is one of the elements of the input vector $\mathbf{x}$ contributing to $p_j$, which is an element of the output vector $\mathbf{p}$, and if $x_i$ "tends to agree" with $p_j$, then the future contribution that $x_i$ makes to $p_j$ should increase. In other words, the idea is to modify connection strengths according to the degree of dependency between the input and output sets. As such, an output set is generated which preserves maximum information about the input set activity, subject to constraints.

For simplicity, let us consider a single element , $p$, from the output set. The mapping of the input set $\mathbf{x}$ to $p$ in the presence of processing noise can be formulated as:

$$p = \sum_i x_i w_i + n \tag{3}$$

where $w_i$ is the weight corresponding to input unit $x_i$ and $n$ is processing noise. Both $p$ and $n$ are considered to be Gaussian with zero means and variances denoted by $v_p$ and $v_n$ respectively. The processing noise is assumed to be uncorrelated with each of the input components. It can be shown that the MI between $p$ and the input vector $\mathbf{x}$ is:

$$I(p; \mathbf{x}) = 0.5 \log \left[ \frac{v_p}{v_n} \right] \tag{4}$$

For a given noise variance, $v_n$, $I(p; \mathbf{x})$ is maximized by maximizing the output variance $v_p$. The ratio $v_p/v_n$ is essentially a signal to noise ratio. Linsker extended his work and studied the effect of noise on maximizing $I(\mathbf{p}; \mathbf{x})$ when the output consists of two elements, hence considering a trade-off between keeping the variances of $p_1$ and $p_2$ large, and reducing the correlation between the two output elements. It has been found that a high noise level favors redundancy. In this case, both $p_1$ and $p_2$ would compute the same linear combination of inputs if there is only one such combination that yields maximum output activity variances. On the other hand, a lower noise level favors diversity in responses. In this case, the two output elements compute different linear combination

of the input vector activities even though the variance of each of the two output elements may be reduced as a result of this choice. Principe and Fisher [12, 13], later, presented an interesting approach that generalizes the infomax principle to arbitrary input distributions and non-linear networks. In this paper, we extend the approach presented by Linsker to the multivariate case, then use the derived learning rules in the proposed HIM algorithm.

### 2.3 Canonical Correlation Analysis

The PCA and Infomax algorithms are very useful in reducing dimensionality within a single channel. However, both may produce output elements that are irrelevant when it comes to generating responses of two separate channels. In other words, PCA and infomax are not efficient when analyzing the relations between two sets of variables.

Contrary to PCA and Infomax, the classical canonical correlation analysis (CCA) [14] has been developed for the sole purpose of finding transformations of two input sets such that correlation between the two transformed output sets is maximized. Let $\mathbf{x}$ and $\mathbf{y}$ be two random vectors, each has $N$ elements with zero mean, then for the two linearly transformed variable $p = \mathbf{w}^T\mathbf{x}$ and $q = \mathbf{v}^T\mathbf{y}$, CCA attempts to find the transformations $\mathbf{w}$ and $\mathbf{v}$ that maximize the correlation between $p$ and $q$. This means that the function to be maximized becomes:

$$\begin{aligned} \rho &= \frac{E[pq]}{\sqrt{E[p^2]E[q^2]}} = \frac{E[\mathbf{w}^T\mathbf{x}\mathbf{y}^T\mathbf{v}]}{\sqrt{E[\mathbf{w}^T\mathbf{x}\mathbf{x}^T\mathbf{w}]E[\mathbf{v}^T\mathbf{y}\mathbf{y}^T\mathbf{v}]}} \\ &= \frac{\mathbf{w}^T\mathbf{R}_{\mathbf{xy}}\mathbf{v}}{\sqrt{\mathbf{w}^T\mathbf{R}_{\mathbf{xx}}\mathbf{w}\mathbf{v}^T\mathbf{R}_{\mathbf{yy}}\mathbf{v}}} \end{aligned} \tag{5}$$

To maximize $\rho$, we can reformulate the problem as:

$$\max_{\mathbf{w},\mathbf{v}} \mathbf{w}^T\mathbf{R}_{\mathbf{xy}}\mathbf{v}, \quad \text{subject to} \quad \mathbf{w}^T\mathbf{R}_{\mathbf{xx}}\mathbf{w}\mathbf{v}^T\mathbf{R}_{\mathbf{yy}}\mathbf{v} = 1 \tag{6}$$

Assuming that $\mathbf{R}_{\mathbf{xx}}$ and $\mathbf{R}_{\mathbf{yy}}$ are non-singular and letting $\mathbf{K} = {\mathbf{R}_{\mathbf{xx}}}^{-1/2}\mathbf{R}_{\mathbf{xy}}{\mathbf{R}_{\mathbf{yy}}}^{-1/2}$, then a singular value decomposition of $\mathbf{K}$ leads to: $\mathbf{K} = (\mathbf{a}_1, \cdots, \mathbf{a}_N)\mathbf{D}(\mathbf{b}_1, \cdots, \mathbf{b}_N)^T$, where $\mathbf{a}_i$ and $\mathbf{b}_i$ are the normalized eigenvectors of $\mathbf{K}\mathbf{K}^T$ and $\mathbf{K}^T\mathbf{K}$ respectively for the eigenvalue $\lambda_i$, and $\mathbf{D} = diag(\sqrt{\lambda_1}, \cdots, \sqrt{\lambda_N})$ is the diagonal matrix of eigenvalues of $\mathbf{K}$.

The first CC (canonical correlation) vectors (those which give the larger correlation) are: $\mathbf{w} = {\mathbf{R}_{\mathbf{xx}}}^{-1/2}\mathbf{a}_1$ and $\mathbf{v} = {\mathbf{R}_{\mathbf{yy}}}^{-1/2}\mathbf{b}_1$, with subsequent CC vectors defined in terms of the subsequent eigenvectors, $\mathbf{a}_i$ and $\mathbf{b}_i$. The reader may refer to [14] for more details.

Larimore [15] extended CCA to non-linear processes by maximizing the MI between the true density and an approximating normal density. Lai and Fyfe [6] proposed a neural implementation

of the CCA. They claimed that the advantage of such implementation is that it can be extended to deal with three data sets and non-linear correlations. Kay [16, 17] presented a class of neural networks based on the concept of MI and allows the contextual guidance of learning. Kay considered two sets of inputs: receptive and contextual. The goal of the network was to discover linear functions that maximize the MI between the two input sets. When these two sets follow a multivariate, elliptically-symmetric probability model, the network basically performs CCA.

## 2.4 The Imax Algorithm

In 1992, Becker [18, 9], generalized the CCA algorithm by using concepts from information theory. She developed the Imax algorithm which was inspired from the human sensory system. A major feature of sensory data is "coherence" across time and across different sensory channels, where coherence, here, means that one part of the signal can somehow be predicted from another part. It has been argued that spatio-temporal and multi-sensory coherence provides important clues for segmenting signals in space and time, and can be used in object localization and identification. The central idea behind the algorithm is that the transformation of two input sets can learn to extract features that are coherent across these inputs.

Assuming that the transformation of each of the two input sets produces one output element, $p$ and $q$ respectively, and both are noisy versions of the same underlying Gaussian signal, each with independent additive Gaussian noise: $p = s + n$, $q = s + m$. Then the mutual information between $p$ and $q$ is:

$$I(p;q) = 0.5 \log \frac{v_p v_q}{v_p v_q - v_{\frac{p+q}{2}} + v_{\frac{p-q}{2}}} \tag{7}$$

where $v_{\frac{p+q}{2}}$ is the variance of $(p+q)/2$. If the input of the first data set has $N$ elements, $x_1, \cdots, x_N$, and if $w_j$ is the weight between $x_j$ and $p$ (see Fig. 1), then according to Becker, maximizing $I(p;q)$ can be achieved by updating $w_j$ according to the following formula:

$$\frac{\partial I(p;q)}{\partial w_j} = \sum_{k=1}^{Q} \frac{1}{Q} \left[ \frac{p^k + q^k - \langle p+q \rangle}{v_{p+q}} - \frac{(p^k - q^k) - \langle p-q \rangle}{v_{p-q}} \right] p^k (1 - q^k) x_j \tag{8}$$

where $Q$ is the number of samples, and $\langle\rangle$ denotes statistical expectation.

Insert Figure 1 here

Becker argued that the difference between CCA and Imax is that CCA is a linear procedure that can be solved analytically. On the other hand, Imax maximizes an equivalent cost function in a non-linear network which may have multiple layers. Thus, Imax is capable of producing

non-linear outputs that CCA cannot. The drawbacks of both CCA and Imax is that they do not reduce redundancy within the output set elements of each channel, and they are incapable of analyzing more than two data sets at a time.

In summary, it is clear from the above that PCA and infomax are only suitable for processing data sets from a single channel, while CCA and Imax are limited to two channels and are unable to reduce redundancy within datasets of each of the channels. This was our rationale behind proposing the hybrid information maximization (HIM) algorithm, described below.

# 3 The Proposed HIM Algorithm

In the previous section we described the well-known subspace mapping techniques of PCA and CCA, and showed that the formal is suitable for reducing redundancy within a data set while the latter maximizes the correlation between two data sets. In addition, descriptions of the infomax and Imax algorithms, which are iterative algorithms based on the concept of MI, were presented. The formal has similar properties to the PCA, while the latter is closely related to the CCA. Both PCA and infomax can be quite useful in processing a single data set, while CCA and Imax have the ability to extract features that exhibit coherence across the data sets of two channels. The HIM algorithm aims at combining these two aspects by attempting to preserve most of the MI between the input and output sets as well as between the different output sets (two or more). Hence, the objective of the HIM is to find a good compromise between PCA and CCA for the case of two channels, then extend the concept to multi-channel data (3 or more channels).

## 3.1 Maximizing MI Between The Input and Output Sets

Maximizing MI between the input and output sets of a single channel is equivalent to maximizing the total information conveyed by the output set, and minimizing the information that the output set conveys to someone who has prior knowledge about the input set. In this paper, we adopt the infomax transformation model by considering the case where we want to map an $(N \times 1)$ input feature set, $\mathbf{x}$, into an $(M \times 1)$ output feature set, $\mathbf{p}$, with $M < N$

$$\mathbf{p} = \mathbf{W}\mathbf{x} + \mathbf{n} \tag{9}$$

where $\mathbf{n}$ is a virtual processing noise, which is considered to be uncorrelated with $\mathbf{x}$, and $\mathbf{W}$ is the transformation matrix, see Fig. 2. It is important to emphasize that this model is different from that of the probabilistic PCA [19]. Unlike the HIM model which attempts to map the observation set $\mathbf{x}$ onto a more efficient and compact set $\mathbf{p}$, the probabilistic PCA considers $\mathbf{x}$ to

be a transformation of some "unobserved" set, $\mathbf{s}$, according to: $\mathbf{x} = \mathbf{W}\mathbf{s} + \mathbf{e}$, where $\mathbf{e}$ represents the observation noise. Using basic concepts from information theory, and assuming that both the input and the noise obey Gaussian distributions, the MI becomes:

$$\begin{aligned} I(\mathbf{p};\mathbf{x}) &= H(\mathbf{p}) + H(\mathbf{x}) - H(\mathbf{p},\mathbf{x}) = H(\mathbf{p}) - H(\mathbf{n}) \\ &= \frac{1}{2}\log[(2\pi e)^M |\mathbf{R_{pp}}|] - \frac{1}{2}\log[(2\pi e)^M |\mathbf{R_{nn}}|] \\ &= \frac{1}{2}\log[|\mathbf{R_{pp}}|] - \frac{1}{2}\log[|\mathbf{R_{nn}}|] \end{aligned} \tag{10}$$

where, $H(.)$ is the entropy function, $\mathbf{R_{pp}}$ and $\mathbf{R_{nn}}$ are the covariance matrices of $\mathbf{p}$ and $\mathbf{n}$ respectively. Using Eq. 9, $\mathbf{R_{pp}}$ can be expressed as $\mathbf{R_{pp}} = \mathbf{W}\mathbf{R_{xx}}\mathbf{W}^T + \mathbf{R_{nn}}$. $\mathbf{W}^T$ is the transpose of $\mathbf{W}$.

Insert Figure 2 here

We maximizing MI between $\mathbf{x}$ and $\mathbf{p}$ by updating $\mathbf{W}$ according to the following learning rule[1]:

$$\frac{\partial I(\mathbf{p};\mathbf{x})}{\partial \mathbf{W}} = \frac{1}{2}\frac{\partial \log[|\mathbf{R_{pp}}|]}{\partial \mathbf{W}} = (\mathbf{W}\mathbf{R_{xx}}\mathbf{W}^T + \mathbf{R_{nn}})^{-1}\mathbf{W}\mathbf{R_{xx}} \tag{11}$$

The optimal value of $M$ can be obtained based on the amount of information lost in the output set, which can be measured by comparing the amount of information (entropy) available at the output set, $H(\mathbf{p})$, to that of the input set, $H(\mathbf{x})$. For example, the 13 EEG input features, described in section 4.2, have an entropy $H(\mathbf{x}) \approx 15$. If we want to preserve 90% of the information or more, *i.e.*, $H(\mathbf{p}) \geq 13.5$, then $M$ should be chosen greater or equal to 10. The amount of information preserved will obviously be application dependent. This concept is similar to the percentage of eigenvalues retained using PCA.

### 3.2 Maximizing MI Between Two Output Sets

This section deals with the problem of linearly mapping two observed feature sets, $\mathbf{x}$ and $\mathbf{y}$, such that the MI between their corresponding output sets is maximized. Let $\mathbf{p} = \mathbf{W}\mathbf{x} + \mathbf{n}$ and $\mathbf{q} = \mathbf{V}\mathbf{y} + \mathbf{m}$, where $\mathbf{W}$ and $\mathbf{V}$ are the transformation matrices of interest, $\mathbf{n}$ and $\mathbf{m}$ represent the virtual processing noise for the two sets (see Fig. 3). Let $\mathbf{z} = [\mathbf{p}^T \quad \mathbf{q}^T]^T$, then the MI between $\mathbf{p}$ and $\mathbf{q}$ becomes:

$$I(\mathbf{p};\mathbf{q}) = H(\mathbf{p}) + H(\mathbf{q}) - H(\mathbf{p},\mathbf{q})$$

[1] Refer to appendix A.1 for the full derivation.

$$= \frac{1}{2}\log[(2\pi e)^M |\mathbf{R_{pp}}|] + \frac{1}{2}\log[(2\pi e)^M |\mathbf{R_{qq}}|] - \frac{1}{2}\log[(2\pi e)^{2M} |\mathbf{R_z}|]$$
$$= \frac{1}{2}\log[|\mathbf{R_{qq}}|/|\mathbf{R_{qq}} - \mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}\mathbf{R_{pq}}|] \quad (12)$$

where,

$$\mathbf{R_z} = \begin{bmatrix} \mathbf{R_{pp}} & \mathbf{R_{pq}} \\ \mathbf{R_{qp}} & \mathbf{R_{qq}} \end{bmatrix} = \begin{bmatrix} \mathbf{W}\mathbf{R_{xx}}\mathbf{W}^T + \mathbf{R_n} & \mathbf{W}\mathbf{R_{xy}}\mathbf{V}^T \\ \mathbf{V}\mathbf{R_{yx}}\mathbf{W}^T & \mathbf{V}\mathbf{R_{yy}}\mathbf{V}^T + \mathbf{R_m} \end{bmatrix} \quad (13)$$

The dimension of $\mathbf{R_{pp}}, \mathbf{R_{pq}}, \mathbf{R_{qp}}$ and $\mathbf{R_{qq}}$ is $M \times M$, and that of $\mathbf{R_z}$ is $2M \times 2M$.

Insert Figure 3 here

Maximizing MI between $\mathbf{p}$ and $\mathbf{q}$ can be achieved by updating $\mathbf{W}$ and $\mathbf{V}$ according to the learning rules[2]:

$$\frac{\partial I(\mathbf{p};\mathbf{q})}{\partial \mathbf{W}} = \mathbf{R_{pp}}^{-1}\mathbf{R_{pq}}(\mathbf{R_{qq}} - \mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}\mathbf{R_{pq}})^{-1}[\mathbf{V}\mathbf{R_{yx}} - \mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}\mathbf{W}\mathbf{R_{xx}}] \quad (14)$$
$$\frac{\partial I(\mathbf{p};\mathbf{q})}{\partial \mathbf{V}} = \mathbf{R_{qq}}^{-1}\mathbf{R_{qp}}(\mathbf{R_{pp}} - \mathbf{R_{pq}}\mathbf{R_{qq}}^{-1}\mathbf{R_{qp}})^{-1}[\mathbf{W}\mathbf{R_{xy}} - \mathbf{R_{pq}}\mathbf{R_{qq}}^{-1}\mathbf{V}\mathbf{R_{yy}}] \quad (15)$$

To maximize MI between the input and output sets as well as between the two output sets, we propose to use the following two expressions:

$$\mathbf{W} = \mathbf{W} + \alpha\frac{\partial I(\mathbf{p};\mathbf{q})}{\partial \mathbf{W}} + \beta\frac{\partial I(\mathbf{p};\mathbf{x})}{\partial \mathbf{W}} \quad (16)$$
$$\mathbf{V} = \mathbf{V} + \alpha\frac{\partial I(\mathbf{p};\mathbf{q})}{\partial \mathbf{V}} + \beta\frac{\partial I(\mathbf{q};\mathbf{y})}{\partial \mathbf{V}} \quad (17)$$

where $\alpha$ and $\beta$ are the learning rates ranging between 0 and 1. For a certain level of processing noise and according to Eq. 10, maximizing the expression for $I(\mathbf{p};\mathbf{x})$ is achieved by maximizing the determinant of $\mathbf{R_{pp}}$ and consequently that of $\mathbf{W}$. If the determinant of $\mathbf{W}$ is allowed to grow indefinitely, then $I(\mathbf{p};\mathbf{x})$ will not reach a stationary point. Therefore, the maximization is constrained to a normalized $\mathbf{W}$ ($|\mathbf{W}\mathbf{W}^T| = 1$), which can be achieved using Lagrange multipliers. The computational cost involved is not significant. For example, adjusting the weight matrices of 2-channel EEG data, as described in section 4.2, for 100 iterations using Matlab routine running under a pentium III PC would take less than one second (mainly 4 matrix inversions of size $M \times M$).

[2] Refer to appendix A.2 for the full derivation.

### 3.3 Maximizing MI Between Three Output Sets

In this section, we derive the learning rules for maximizing MI between three different sets, with inputs $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, and outputs $\mathbf{p}$, $\mathbf{q}$, $\mathbf{r}$, and weights $\mathbf{W}$, $\mathbf{V}$, $\mathbf{U}$ respectively. We will first present the expression of $I(\mathbf{p};\mathbf{q};\mathbf{r})$, then determine its derivative with respect to the weights.

$$\begin{aligned} I(\mathbf{p};\mathbf{q};\mathbf{r}) &= H(\mathbf{p}) + H(\mathbf{q}) + H(\mathbf{r}) - H(\mathbf{p},\mathbf{q}) - H(\mathbf{p},\mathbf{r}) - H(\mathbf{q},\mathbf{r}) + H(\mathbf{p},\mathbf{q},\mathbf{r}) \\ &= \frac{1}{2}\log[(2\pi e)^M|\mathbf{R_{pp}}|] + \frac{1}{2}\log[(2\pi e)^M|\mathbf{R_{qq}}|] + \frac{1}{2}\log[(2\pi e)^M|\mathbf{R_{rr}}|] - \\ &\quad \frac{1}{2}\log[(2\pi e)^{2M}|\mathbf{R_a}|] - \frac{1}{2}\log[(2\pi e)^{2M}|\mathbf{R_b}|] - \frac{1}{2}\log[(2\pi e)^{2M}|\mathbf{R_c}|] + \\ &\quad \frac{1}{2}\log[(2\pi e)^{2M}|\mathbf{R_d}|] \end{aligned} \tag{18}$$

where,

$$\begin{aligned} |\mathbf{R_a}| &= |\mathbf{R_{pp}}||\mathbf{R_{qq}} - \mathbf{R_{qp}}{\mathbf{R_{pp}}}^{-1}\mathbf{R_{pq}}| \\ |\mathbf{R_b}| &= |\mathbf{R_{qq}}||\mathbf{R_{rr}} - \mathbf{R_{rq}}{\mathbf{R_{qq}}}^{-1}\mathbf{R_{qr}}| \\ |\mathbf{R_c}| &= |\mathbf{R_{rr}}||\mathbf{R_{pp}} - \mathbf{R_{pr}}{\mathbf{R_{rr}}}^{-1}\mathbf{R_{rp}}| \\ |\mathbf{R_d}| &= \left|\begin{bmatrix} \mathbf{R_{pp}} & \mathbf{R_{pq}} \\ \mathbf{R_{qp}} & \mathbf{R_{qq}} \end{bmatrix}\right| \Bigg|\underbrace{\mathbf{R_{rr}} - \begin{bmatrix} \mathbf{R_{rp}} & \mathbf{R_{rq}} \end{bmatrix} \begin{bmatrix} \mathbf{R_{pp}} & \mathbf{R_{pq}} \\ \mathbf{R_{qp}} & \mathbf{R_{qq}} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{R_{pr}} \\ \mathbf{R_{qr}} \end{bmatrix}}_{\mathbf{G_r}}\Bigg| \\ |\mathbf{R_d}| &= |\mathbf{R_{pp}}||\mathbf{R_{qq}} - \mathbf{R_{qp}}{\mathbf{R_{pp}}}^{-1}\mathbf{R_{pq}}||\mathbf{G_r}| \end{aligned}$$

$$\therefore\ I(\mathbf{p};\mathbf{q};\mathbf{r}) = \frac{1}{2}\log\left[\frac{|\mathbf{R_{pp}}||\mathbf{G_r}|}{|\mathbf{R_{rr}} - \mathbf{R_{rq}}{\mathbf{R_{qq}}}^{-1}\mathbf{R_{qr}}||\mathbf{R_{pp}} - \mathbf{R_{pr}}{\mathbf{R_{rr}}}^{-1}\mathbf{R_{rp}}|}\right] \tag{19}$$

Let $\mathbf{H_p} = (\mathbf{R_{pp}} - \mathbf{R_{pr}}{\mathbf{R_{rr}}}^{-1}\mathbf{R_{rp}})^{-1}$, $\mathbf{H_q} = (\mathbf{R_{qq}} - \mathbf{R_{qp}}{\mathbf{R_{pp}}}^{-1}\mathbf{R_{pq}})^{-1}$, and $\mathbf{H_r} = (\mathbf{R_{rr}} - \mathbf{R_{rq}}{\mathbf{R_{qq}}}^{-1}\mathbf{R_{qr}})^{-1}$, then

$$\begin{aligned} \mathbf{G_r} &= \mathbf{R_{rr}} - \mathbf{R_{rp}}{\mathbf{R_{pp}}}^{-1}\mathbf{R_{pr}} - \mathbf{R_{rp}}{\mathbf{R_{pp}}}^{-1}\mathbf{R_{pq}}\mathbf{H_q}\mathbf{R_{qp}}{\mathbf{R_{pp}}}^{-1}\mathbf{R_{pr}} + \\ &\quad \mathbf{R_{rq}}\mathbf{H_q}\mathbf{R_{qp}}{\mathbf{R_{pp}}}^{-1}\mathbf{R_{pr}} + \mathbf{R_{rp}}{\mathbf{R_{pp}}}^{-1}\mathbf{R_{pq}}\mathbf{H_q}\mathbf{R_{qr}} - \mathbf{R_{rq}}\mathbf{H_q}\mathbf{R_{qr}} \end{aligned}$$

Maximizing MI between $\mathbf{p}$, $\mathbf{q}$ and $\mathbf{r}$ can be achieved by updating the weighting matrix $\mathbf{U}$ according to the learning rule[3]:

[3] Refer to appendix A.3 for the full derivation.

$$\frac{\partial I(\mathbf{p};\mathbf{q};\mathbf{r})}{\partial \mathbf{U}} = \mathbf{G_r}^{-1}[\mathbf{U R_{zz}} - \mathbf{R_{rp}} \mathbf{R_{pp}}^{-1} \mathbf{W R_{xz}} - \mathbf{R_{rp}} \mathbf{R_{pp}}^{-1} \mathbf{R_{pq}} \mathbf{H_q} \mathbf{R_{qp}} \mathbf{R_{pp}}^{-1} \mathbf{W R_{xz}} + \mathbf{R_{rp}} \mathbf{R_{pp}}^{-1} \mathbf{R_{pq}} \mathbf{H_q} \mathbf{V R_{yz}} + \mathbf{R_{rq}} \mathbf{H_q} \mathbf{R_{qp}} \mathbf{R_{pp}}^{-1} \mathbf{W R_{xz}} - \mathbf{R_{rq}} \mathbf{H_q} \mathbf{V R_{yz}}] - \mathbf{H_r}[\mathbf{U R_{zz}} - \mathbf{R_{rq}} \mathbf{R_{qq}}^{-1} \mathbf{V R_{yz}}] - \mathbf{R_{rr}}^{-1} \mathbf{R_{rp}} \mathbf{H_p} [\mathbf{R_{pr}} \mathbf{R_{rr}}^{-1} \mathbf{U R_{zz}} - \mathbf{W R_{xz}}] \quad (20)$$

The formulas for updating $\mathbf{W}$ and $\mathbf{V}$ can be obtained using similar approach.

### 3.4 Extension to Higher Number of Sets

It is clear that the expression of MI for 3 channels has more terms than that of the 2 channel case (7 and 3 respectively), and some of these terms exhibit a high degree of complexity. As the number of channels increases, the number of terms increase exponentially (15 and 31 for 4 and 5 channels respectively), and the overall expression becomes more complex. One possible way to overcome this degree of complexity is to generalize the HIM algorithm to any number of channels by maximizing MI between *all possible pairs* of channels (or subsets of three channels). For example, in the case of 4 channels, the HIM algorithm can be implemented according to one of the following two approaches for channel 1:

$$\mathbf{W} = \mathbf{W} + \alpha \left[ \frac{\partial I(\mathbf{p};\mathbf{q})}{\partial \mathbf{W}} + \frac{\partial I(\mathbf{p};\mathbf{r})}{\partial \mathbf{W}} + \frac{\partial I(\mathbf{p};\mathbf{s})}{\partial \mathbf{W}} + \frac{\partial I(\mathbf{q};\mathbf{r})}{\partial \mathbf{W}} + \frac{\partial I(\mathbf{q};\mathbf{s})}{\partial \mathbf{W}} + \frac{\partial I(\mathbf{r};\mathbf{s})}{\partial \mathbf{W}} \right] + \beta \frac{\partial I(\mathbf{p};\mathbf{x})}{\partial \mathbf{W}}, \quad \text{(using subsets of 2 channels)} \quad (21)$$

or

$$\mathbf{W} = \mathbf{W} + \alpha \left[ \frac{\partial I(\mathbf{p};\mathbf{q};\mathbf{r})}{\partial \mathbf{W}} + \frac{\partial I(\mathbf{p};\mathbf{q};\mathbf{s})}{\partial \mathbf{W}} + \frac{\partial I(\mathbf{p};\mathbf{r};\mathbf{s})}{\partial \mathbf{W}} + \frac{\partial I(\mathbf{q};\mathbf{r};\mathbf{s})}{\partial \mathbf{W}} \right] + \beta \frac{\partial I(\mathbf{p};\mathbf{x})}{\partial \mathbf{W}}, \quad \text{(using subsets of 3 channels)} \quad (22)$$

The learning rules for the other three channels are obtained in a similar fashion. Even though this may not result in the optimal solution, our extensive simulations have shown that most of the MI between input and output sets is preserved, while still maintaining a reasonable amount of mutual information between output sets for the different channels, as will be explained in the next section.

# 4 Experimental Results

In this section, the performance of HIM is first analyzed using synthetic data by evaluating the amount of information loss that occurs when reducing the dimensionality of input sets. The issue of choosing appropriate learning rates using real data is then discussed. Finally, comparisons between the HIM, PCA and CCA in terms of MI and classification accuracy are presented. In all experiments, the data are assumed to be Gaussian and the MIs are calculated using Eqs. 10, 12 and 18. Since the noise is virtual, we can assign an arbitrary value to the determinant of its covariance matrix or just remove it, which is the case in the experiments conducted in this section.

## 4.1 Performance Using Synthetic Data

The purpose of this experiment is to analyze the performance of HIM for the case of two channels and study the effect of reducing the dimensionality of two input sets. Consider the following two input sets $\mathbf{x} = [x_1, x_2, x_3, x_4]$ and $\mathbf{y} = [y_1, y_2, y_3, y_4]$. We defined the MI between the pairs of $\mathbf{x}$, and the MI between the pairs of $\mathbf{y}$, such that $(x_2, x_3)$, and $(y_2, y_3)$, exhibit some degree of dependency, while the MI between $x_4$ and $y_4$ is small compared to the other three pairs, as given below:

$$I(x_i; x_j) = \begin{bmatrix} 1.4189 & 0.0603 & 0.0687 & 0.0631 \\ 0.0603 & 1.4189 & 0.5712 & 0.0550 \\ 0.0687 & 0.5712 & 1.4189 & 0.0568 \\ 0.0631 & 0.0550 & 0.0568 & 1.4189 \end{bmatrix}$$

$$I(x_i; y_i) = \begin{bmatrix} 0.5743 & 0.3509 & 0.3640 & 0.1643 \end{bmatrix}$$

The overall entropy of the two input sets and MI between them are: $H(\mathbf{x}) = 4.9384$, $H(\mathbf{y}) = 4.9312$, and $I(\mathbf{x}; \mathbf{y}) = 1.1600$. Firstly, the HIM algorithm was applied to extract three variables. The obtained MI between the output pairs are:

$$I(p_i; p_j) = \begin{bmatrix} 1.4189 & 0.0446 & 0.0801 \\ 0.0446 & 1.4189 & 0.0340 \\ 0.0801 & 0.0340 & 1.4189 \end{bmatrix}$$

$$I(p_i; q_i) = \begin{bmatrix} 0.5922 & 0.1232 & 0.4704 \end{bmatrix}$$

with overall entropy and MI: $H(\mathbf{p}) = 4.3839$, $H(\mathbf{q}) = 4.1954$, and $I(\mathbf{p}; \mathbf{q}) = 1.0970$. The dependency between the extracted variables has dramatically dropped since the values of the off-diagonal

elements of $I(p_i; p_j)$ are close to zero. In addition, there is a slight improvement on the average MI between the output pairs of the two sets, reflected by the increase of the average $I(p_i; q_i)$. Even though the size of the output sets is reduced by 25%, the information loss of the transformation was only 11%, while MI between $\mathbf{p}$ and $\mathbf{q}$ is almost similar to that between $\mathbf{x}$ and $\mathbf{y}$.

The HIM algorithm was then applied to extract two variables from the original four input ones. The resulting MI values became:

$$I(p_i; p_j) = \begin{bmatrix} 1.4189 & 0.0943 \\ 0.0943 & 1.4189 \end{bmatrix}$$

$$I(p_i; q_i) = \begin{bmatrix} 0.5883 & 0.5395 \end{bmatrix}$$

the overall entropy and MI became: $H(\mathbf{p}) = 3.1028$, $H(\mathbf{q}) = 2.9986$, and $I(\mathbf{p}; \mathbf{q}) = 1.0362$. The results show that this transformation maintains high degree of independency between the output elements as well as enhancing the average MI between pairs of the two output sets. The information loss of the transformation increased to 37%, while MI between the two sets has been slightly reduced. In summary, the HIM algorithm enabled us to maintain most of the information within and between the two sets when mapping the 4-dimensional input vectors into 3-dimensional output vectors. When the dimension of the output sets is reduced to 2, we could still maintain most of the information between the two sets, however the amount of information loss within each set increased, which may not be acceptable in some applications. Therefore, this experiment illustrates that the HIM has the ability to minimize the dependency within each set and maximize it between different sets, given that appropriate number of output elements are chosen.

## 4.2 Choosing Appropriate Learning Rates

We will investigate here the issue of choosing $\alpha$ and $\beta$, the learning rates of the two parts of HIM for the case of two channels (Eqs. 16 and 17), such that the MI between the input and output sets as well as the MI between the two output sets is maximized. Experiments were carried on two types of data. The first one is obtained from two neighboring channels of an 8-second EEG segments that represent two mental tasks: left and right hand movements[4]. From the 8-second segment of each channel, 13 features were extracted and used as an input set. The extracted features were: dominant frequency and its amplitude, average power in main lobe, energy, zero crossing and number of extrema of each segment, average half-waves amplitude and duration, and poles of AR model. The second type of data is obtained from speech signal with two different

[4]The authors would like to thank Prof. G. Pfurtscheller, Technical University, Graz, Austria, for providing the EEG data.

added noise levels ($SNR = 20$ and $SNR = 25$), to simulate the case of microphone array. Each input set consisted of the following features: 16 mel frequency spectral coefficients and 10 energy wavelet bands. These features were extracted from each speech frame of 256 millisecond length. For both experiments, the HIM algorithm is used to map the input of each channel to a lower subspace, 10 and 22 elements respectively. Based on Eqs. 16 and 17, we compute the values of MI between the input and output sets, $I(\mathbf{p}; \mathbf{x})$, and between the two output sets, $I(\mathbf{p}; \mathbf{q})$ for different values of $\alpha$ and $\beta$ ranging between 0 and 1. The obtained results are shown in Figs. 4 to 8. 300 iterations were used to update the weights for each value of $\alpha$ and $\beta$.

Insert Figures 4, 5 here

From these figures, we notice that the maximum value of $I(\mathbf{p}; \mathbf{x})$ obtained by certain values of $\alpha$ and $\beta$ will not lead to a maximum value of $I(\mathbf{p}; \mathbf{q})$ and vice versa. For example, when $\alpha = 0$ and $\beta = 1$, $I(\mathbf{p}; \mathbf{x})$ will reach a maximum value, whereas $I(\mathbf{p}; \mathbf{q})$ does not achieve that. This is expected, because according to Eq. 11, the maximization of $I(\mathbf{p}; \mathbf{x})$ does not depend on $H(\mathbf{p}; \mathbf{q})$, which is part of Eq. 12. The same thing is applied to $I(\mathbf{q}; \mathbf{y})$. Also, the maximization of $I(\mathbf{p}; \mathbf{q})$ may be achieved by maximizing the overlap between $H(\mathbf{p})$ and $H(\mathbf{q})$, and not necessarily their values. For the case of EEG data, it can be inferred from Fig. 4 that $I(\mathbf{p}; \mathbf{x})$ reaches steady state when $\beta/\alpha > 0.2$ with a maximum value of 13.82, while from Fig. 5, it can be seen that most of the MI between $\mathbf{p}$ and $\mathbf{q}$ is preserved when $\beta/\alpha < 0.2$ with a maximum value of 8.2. To make an appropriate choice for both $\alpha$ and $\beta$, Fig. 6 shows the plots of $I(\mathbf{p}; \mathbf{q})$ versus $I(\mathbf{p}; \mathbf{x})$ for various values of $\alpha$ and $\beta$. Choosing the values of $\alpha$ and $\beta$ as 1.0 and 0.2 respectively, preserves approximately 95% of the maximum values of $I(\mathbf{p}; \mathbf{q})$ and $I(\mathbf{p}; \mathbf{x})$.

Insert Figure 6 here

The same thing is applied to the speech data, except that there is a sharper increase and decrease in $I(\mathbf{p}; \mathbf{x})$ and $I(\mathbf{p}; \mathbf{q})$ respectively, as shown in Figs. 7 and 8. This makes the optimal choice of $\alpha$ and $\beta$ that preserves most of the information within and between the two transformations be 1.0 and 0.07 respectively.

Insert Figures 7 and 8 here

We can conclude from these two experiments that the appropriate choice of $\alpha$ and $\beta$ is application dependent, but in general setting the values of $\alpha$ and $\beta$ to 1.0 and 0.1 respectively, is reasonable, where more than 90% of the maximum values of $I(\mathbf{p}; \mathbf{q})$ and $I(\mathbf{p}; \mathbf{x})$ for both speech and EEG data are preserved. However, for a specific application, better results can be achieved through fine tuning these variables.

## 4.3 Comparison With PCA and CCA

In this section, we will compare the performance of HIM to that of PCA and CCA. We will present first an information-based comparison for two channels, then extend that to multiple channels. Also, a two-channel classification problem will be used to compare the performance of the three methods with respect to their classification accuracy.

### 4.3.1 Two Channel Comparison in Terms of Mutual Information

We will compare here the performance of HIM to that of the PCA and CCA algorithms[5] in terms of MI between input and output sets and between output sets of two-channel data. We have used both the EEG and speech data in this experiment. First, we set the values of $\alpha$ and $\beta$ to 0.0 and 1.0 respectively (only maximizing MI within each transformation). Figs. 9 and 10 show the value of $I(\mathbf{p};\mathbf{x})$ obtained for different number of output elements. It is clear that in both cases the performance of HIM is equivalent to that of PCA, while the CCA is much poorer. On the other hand, when we set the values of $\alpha$ and $\beta$ to 1.0 and 0.0 respectively (only maximizing MI between the different sets), and measure $I(\mathbf{p};\mathbf{q})$, we found that CCA and HIM have a very comparable performance and both outperform the PCA, as shown in Figs 11 and 12. After that, we set the values of $\alpha$ and $\beta$ to 1.0 and 0.2 (EEG data) and 1.0 and 0.07 (speech data) respectively, where we want to retain most of the information within the two transformations and between their output elements. Figs. 13 and 14 show the performance of the three techniques with respect to $I(\mathbf{p};\mathbf{x})$, while Figs. 15 and 16 show the performance of the three techniques with respect to $I(\mathbf{p};\mathbf{q})$. It is clear that without loosing much information within each transformation ($I(\mathbf{p};\mathbf{x})$ and $I(\mathbf{q};\mathbf{y})$), the HIM algorithm retained most of the information with respect to $I(\mathbf{p};\mathbf{q})$, for both EEG and speech data.

Insert Figures 9, 10, 11, 12, 13, 14, 15 and 16 here

In conclusion, the HIM algorithm has the ability to replicate the performance of PCA and CCA when we set the learning rates appropriately. In addition, it has the ability to adapt between maximizing MI within the two transformations and maximizing it between their outputs.

### 4.3.2 Multi-Channel Comparison in Terms of Mutual Information

Unlike CCA that has a limitation of only two channels, the HIM algorithm can maximize MI between more than two channels, as explained in sections 3.3 and 3.4. For the case of three

[5] PCA can be examined from an information theoretic standpoint [20], which is also true for CCA.

channels, the HIM algorithm has the ability to maximize $I(\mathbf{p};\mathbf{q};\mathbf{r})$ while preserving most of MI within each transformation. Fig. 17 shows the MI between three EEG channels for various number of output elements obtained using both PCA and HIM. Note that CCA has not been included in the comparison because it cannot be used for more than two channels. It is clear that HIM outperforms PCA regardless the number of output elements considered. This is due to the fact that PCA treats each channel individually.

Insert Figure 17 here

For the case of more than three channels, we carried an experiment using five channels and compared the performance of the *approximated* HIM, using subsets of 2 and 3 channels, with that of PCA (see Fig. 18).

Insert Figure 18 here

It is clear that these two versions of HIM both outperform the PCA in terms of inter-channel MI, while they preserve more than 90% of the MI within each transformation. Even though these two approximated HIM might only achieve near optimal $I(\mathbf{p};\mathbf{q};\mathbf{r};\mathbf{s};\mathbf{t})$, they still outperform, by far, the PCA. In addition, they allow us to implement HIM for any number of channels without the need to derive complex formulas for the learning rules, and still achieve very promising results.

### 4.3.3 Comparison in Terms of Classification Accuracy

The purpose of the experiment conducted here is to classify speech segments obtained from a simulated two channels, as described in section 4.2, according to their manner of articulation. Six classes are considered, namely: vowel, semi-vowel, nasal, closure, stop, fricative, lateral, rhotic, and silence. The features obtained by applying PCA, CCA, and HIM with number of output elements ranging from 1 to 14 were used to represent speech segments that are fed to an artificial neural network. Fig. 19 shows the average classification accuracy of the two simulated speech channels for different number of output elements. It can be seen from this figure that the performance of the PCA is better than that of CCA when the number of output elements ($M < 8$). However, the effect of maximizing the correlation between the two channels starts to make clear impact when $M > 9$, which makes the CCA outperforms the PCA. Regarding the HIM algorithm, it approaches the PCA when $M \leq 4$ and, on average, outperforms it when there are more than 4 elements. In addition, it outperforms the CCA when $M < 10$ and their performance become similar when $M > 10$.

Insert Figure 19 here

In conclusion, the HIM algorithm allows us to achieve excellent performance for any number of elements, which can not be achieved by either PCA or CCA. Further research in relation to speech recognition from multi-channels sources will be carried in the future. Since the Gaussian model can only cater for second order statistics, we are currently extending the HIM to the more general multivariate generalized Laplace distribution (MGLD).

# 5 Conclusion

A new subspace mapping algorithm based on the maximization of MI between the input and output sets of each channel and between the output sets of different channels has been developed. Derivations for two and three channels have been presented, and a possible extension to four or more channels has also been considered. We have shown that by assigning appropriate learning rates, $\alpha$ and $\beta$, to the learning rules, most of the information within and between the transformations of different channels can be preserved. Experimental results using synthetic data show the strength of the proposed algorithm by removing redundancy from each set and maximizing the MI between the different sets. In addition, real EEG and speech data were used to compare the performance of HIM, PCA and CCA techniques in terms of MI values (within and between transformations) for the cases of two, three and five channels. These experiments show the advantage of using HIM, which provides a good compromise between PCA and CCA for the two-channel case. When processing more than two channels, the HIM proved to be especially useful, as CCA is inapplicable. In a two-channel speech classification problem, PCA achieved better performance than CCA when the number of elements of the output sets is small, while CCA outperformed PCA for higher number of elements, which reflects the importance of inter-channel relationship. The HIM, on the other hand, achieved the optimal or near optimal performance, compared to PCA and CCA, regardless the number of output elements. The concept proposed here is novel with a great potential in developing a new framework for multi-channel subspace signal mapping.

# A Appendix

## A.1 Learning Rule to Maximize MI Between The Input and Output Sets

According to [21], $d|\mathbf{X}| = |\mathbf{X}|tr[\mathbf{X}^{-1}d\mathbf{X}]$.

$$d|\mathbf{R_{pp}}| = |\mathbf{R_{pp}}|tr[{\mathbf{R_{pp}}}^{-1}d(\mathbf{W}\mathbf{R_{xx}}\mathbf{W}^T + \mathbf{R_{nn}})]$$

$$
\begin{aligned}
&= |\mathbf{R_{pp}}| tr[\mathbf{R_{pp}}^{-1}(d\mathbf{W})\mathbf{R_{xx}}\mathbf{W}^T + \mathbf{R_{pp}}^{-1}\mathbf{W}\mathbf{R_{xx}}(d\mathbf{W})^T] \\
&= |\mathbf{R_{pp}}|[tr[\mathbf{R_{pp}}^{-1}(d\mathbf{W})\mathbf{R_{xx}}\mathbf{W}^T] + tr[\mathbf{R_{pp}}^{-1}\mathbf{W}\mathbf{R_{xx}}(d\mathbf{W})^T]] \\
&= |\mathbf{R_{pp}}|[tr[\mathbf{R_{pp}}^{-1}(d\mathbf{W})\mathbf{R_{xx}}\mathbf{W}^T] + tr[(d\mathbf{W})\mathbf{R_{xx}}\mathbf{W}^T\mathbf{R_{pp}}^{-1}]] \\
&= 2|\mathbf{R_{pp}}|[tr[\mathbf{R_{xx}}\mathbf{W}^T\mathbf{R_{pp}}^{-1}(d\mathbf{W})]
\end{aligned}
$$

As a result,

$$
\begin{aligned}
\frac{\partial|\mathbf{R_{pp}}|}{\partial\mathbf{W}} &= 2|\mathbf{R_{pp}}|\mathbf{R_{pp}}^{-1}\mathbf{W}\mathbf{R_{xx}} \\
\frac{\partial log|\mathbf{R_{pp}}|}{\partial\mathbf{W}} &= 2\mathbf{R_{pp}}^{-1}\mathbf{W}\mathbf{R_{xx}} \qquad (23)
\end{aligned}
$$

Which leads to

$$
\frac{\partial I(\mathbf{p};\mathbf{x})}{\partial\mathbf{W}} = (\mathbf{W}\mathbf{R_{xx}}\mathbf{W^T} + \mathbf{R_{nn}})^{-1}\mathbf{W}\mathbf{R_{xx}}
$$

## A.2 Learning Rule to Maxmize MI Between Two Output Sets

Let $\mathbf{N} = \mathbf{R_{qq}} - \mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}\mathbf{R_{pq}}$.

To derive the log of determinant of $\mathbf{N}$ with respect to $\mathbf{W}$,

$$
\begin{aligned}
d|\mathbf{N}| &= |\mathbf{N}| tr[\mathbf{N}^{-1}[-\mathbf{V}\mathbf{R_{yx}}(d\mathbf{W})^T\mathbf{R_{pp}}^{-1}\mathbf{R_{pq}} - \mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}(d\mathbf{W})\mathbf{R_{xy}}\mathbf{V}^T + \\
&\quad \mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}[(d\mathbf{W})\mathbf{R_{xx}}\mathbf{W}^T + \mathbf{W}\mathbf{R_{xx}}(d\mathbf{W})^T]\mathbf{R_{pp}}^{-1}\mathbf{R_{pq}}] \\
&= |\mathbf{N}|\{-tr[\mathbf{N}^{-1}\mathbf{V}\mathbf{R_{yx}}(d\mathbf{W})^T\mathbf{R_{pp}}^{-1}\mathbf{R_{pq}}] - tr[\mathbf{N}^{-1}\mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}(d\mathbf{W})\mathbf{R_{xy}}\mathbf{V}^T] + \\
&\quad tr[\mathbf{N}^{-1}\mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}(d\mathbf{W})\mathbf{R_{xx}}\mathbf{W}^T\mathbf{R_{pp}}^{-1}\mathbf{R_{pq}}] + \\
&\quad tr[\mathbf{N}^{-1}\mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}\mathbf{W}\mathbf{R_{xx}}(d\mathbf{W})^T\mathbf{R_{pp}}^{-1}\mathbf{R_{pq}}]\} \\
&= |\mathbf{N}|\{-tr[\mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}(d\mathbf{W})\mathbf{R_{xy}}\mathbf{V}^T\mathbf{N}^{-1}] - tr[\mathbf{N}^{-1}\mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}(d\mathbf{W})\mathbf{R_{xy}}\mathbf{V}^T] + \\
&\quad tr[\mathbf{N}^{-1}\mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}(d\mathbf{W})\mathbf{R_{xx}}\mathbf{W}^T\mathbf{R_{pp}}^{-1}\mathbf{R_{pq}}] + \\
&\quad tr[\mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}(d\mathbf{W})\mathbf{R_{xx}}\mathbf{W}^T\mathbf{R_{pp}}^{-1}\mathbf{R_{pq}}\mathbf{N}^{-1}]\} \\
&= 2|\mathbf{N}|\{-tr[\mathbf{R_{xy}}\mathbf{V}^T\mathbf{N}^{-1}\mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}(d\mathbf{W})] + \\
&\quad tr[\mathbf{R_{xx}}\mathbf{W}^T\mathbf{R_{pp}}^{-1}\mathbf{R_{pq}}\mathbf{N}^{-1}\mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}(d\mathbf{W})]\}
\end{aligned}
$$

As a result

$$
\frac{\partial log|\mathbf{N}|}{\partial\mathbf{W}} = 2\{-\mathbf{R_{pp}}^{-1}\mathbf{R_{pq}}\mathbf{N}^{-1}\mathbf{V}\mathbf{R_{yx}} + \mathbf{R_{pp}}^{-1}\mathbf{R_{pq}}\mathbf{N}^{-1}\mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}\mathbf{W}\mathbf{R_{xx}}\} \qquad (24)
$$

Eqs. 23 and 24 lead to:

$$
\frac{\partial I(\mathbf{p};\mathbf{q})}{\partial\mathbf{W}} = \mathbf{R_{pp}}^{-1}\mathbf{R_{pq}}(\mathbf{R_{qq}} - \mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}\mathbf{R_{pq}})^{-1}[\mathbf{V}\mathbf{R_{yx}} - \mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}\mathbf{W}\mathbf{R_{xx}}]
$$

### A.3 Learning Rule to Maxmize MI Between Three Output Sets

Let $\mathbf{A_p} = (\mathbf{R_{pp}})^{-1}$, $\mathbf{B_q} = (\mathbf{R_{qq}} - \mathbf{R_{qp}}\mathbf{A_p}\mathbf{R_{pq}})^{-1}$. To derive the log of determinant of $\mathbf{G_r}$ with respect to $\mathbf{U}$,

$$
\begin{aligned}
d|\mathbf{G_r}| &= |\mathbf{G_r}| tr\{\mathbf{G_r}^{-1} d[\mathbf{R_{rr}} - \mathbf{R_{rp}}\mathbf{A_p}\mathbf{R_{pr}} - \mathbf{R_{rp}}\mathbf{A_p}\mathbf{R_{pq}}\mathbf{B_q}\mathbf{R_{qp}}\mathbf{A_p}\mathbf{R_{pr}} + \\
&\quad \mathbf{R_{rq}}\mathbf{B_q}\mathbf{R_{qp}}\mathbf{A_p}\mathbf{R_{pr}} + \mathbf{R_{rp}}\mathbf{A_p}\mathbf{R_{pq}}\mathbf{B_q}\mathbf{R_{qr}} - \mathbf{R_{rq}}\mathbf{B_q}\mathbf{R_{qr}}]\} \\
&= |\mathbf{G_r}| tr\{\mathbf{G_r}^{-1}[(d\mathbf{U})\mathbf{R_{zz}}\mathbf{U}^T + \mathbf{U}\mathbf{R_{zz}}(d\mathbf{U})^T - (d\mathbf{U})\mathbf{R_{zx}}\mathbf{W}^T\mathbf{A_p}\mathbf{R_{pr}} - \mathbf{R_{rp}}\mathbf{A_p}\mathbf{W}\mathbf{R_{xz}}(d\mathbf{U})^T \\
&\quad -(d\mathbf{U})\mathbf{R_{zx}}\mathbf{W}^T\mathbf{A_p}\mathbf{R_{pq}}\mathbf{B_q}\mathbf{R_{qp}}\mathbf{A_p}\mathbf{R_{pr}} - \mathbf{R_{rp}}\mathbf{A_p}\mathbf{R_{pq}}\mathbf{B_q}\mathbf{R_{qp}}\mathbf{A_p}\mathbf{W}\mathbf{R_{xz}}(d\mathbf{U})^T + \\
&\quad (d\mathbf{U})\mathbf{R_{zy}}\mathbf{V}^T\mathbf{B_q}\mathbf{R_{qp}}\mathbf{A_p}\mathbf{R_{pr}} + \mathbf{R_{rq}}\mathbf{B_q}\mathbf{R_{qp}}\mathbf{A_p}\mathbf{W}\mathbf{R_{xz}}(d\mathbf{U})^T + (d\mathbf{U})\mathbf{R_{zx}}\mathbf{W}^T\mathbf{A_p}\mathbf{R_{pq}}\mathbf{B_q}\mathbf{R_{qr}} \\
&\quad +\mathbf{R_{rp}}\mathbf{A_p}\mathbf{R_{pq}}\mathbf{B_q}\mathbf{V}\mathbf{R_{yz}}(d\mathbf{U})^T - (d\mathbf{U})\mathbf{R_{zy}}\mathbf{V}\mathbf{B_q}\mathbf{R_{qr}} - \mathbf{R_{rq}}\mathbf{B_q}\mathbf{V}\mathbf{R_{yz}}(d\mathbf{U})^T]\} \\
&= |\mathbf{G_r}|\{tr[\mathbf{G_r}^{-1}(d\mathbf{U})\mathbf{R_{zz}}\mathbf{U}^T] + tr[\mathbf{G_r}^{-1}\mathbf{U}\mathbf{R_{zz}}(d\mathbf{U})^T] - tr[\mathbf{G_r}^{-1}(d\mathbf{U})\mathbf{R_{zx}}\mathbf{W}^T\mathbf{A_p}\mathbf{R_{pr}}] - \\
&\quad tr[\mathbf{G_r}^{-1}\mathbf{R_{rp}}\mathbf{A_p}\mathbf{W}\mathbf{R_{xz}}(d\mathbf{U})^T] - tr[\mathbf{G_r}^{-1}(d\mathbf{U})\mathbf{R_{zx}}\mathbf{W}^T\mathbf{A_p}\mathbf{R_{pq}}\mathbf{B_q}\mathbf{R_{qp}}\mathbf{A_p}\mathbf{R_{pr}}] - \\
&\quad tr[\mathbf{G_r}^{-1}\mathbf{R_{rp}}\mathbf{A_p}\mathbf{R_{pq}}\mathbf{B_q}\mathbf{R_{qp}}\mathbf{A_p}\mathbf{W}\mathbf{R_{xz}}(d\mathbf{U})^T] + tr[\mathbf{G_r}^{-1}(d\mathbf{U})\mathbf{R_{zy}}\mathbf{V}^T\mathbf{B_q}\mathbf{R_{qp}}\mathbf{A_p}\mathbf{R_{pr}}] + \\
&\quad tr[\mathbf{G_r}^{-1}\mathbf{R_{rq}}\mathbf{B_q}\mathbf{R_{qp}}\mathbf{A_p}\mathbf{W}\mathbf{R_{xz}}(d\mathbf{U})^T] + tr[\mathbf{G_r}^{-1}(d\mathbf{U})\mathbf{R_{zx}}\mathbf{W}^T\mathbf{A_p}\mathbf{R_{pq}}\mathbf{B_q}\mathbf{R_{qr}}] + \\
&\quad tr[\mathbf{G_r}^{-1}\mathbf{R_{rp}}\mathbf{A_p}\mathbf{R_{pq}}\mathbf{B_q}\mathbf{V}\mathbf{R_{yz}}(d\mathbf{U})^T] - tr[\mathbf{G_r}^{-1}(d\mathbf{U})\mathbf{R_{zy}}\mathbf{V}\mathbf{B_q}\mathbf{R_{qr}}] - \\
&\quad tr[\mathbf{G_r}^{-1}\mathbf{R_{rq}}\mathbf{B_q}\mathbf{V}\mathbf{R_{yz}}(d\mathbf{U})^T]\} \\
&= |\mathbf{G_r}|\{tr[\mathbf{R_{zz}}\mathbf{U}^T\mathbf{G_r}^{-1}(d\mathbf{U})] + tr[(d\mathbf{U})\mathbf{R_{zz}}\mathbf{U}^T\mathbf{G_r}^{-1}] - tr[\mathbf{R_{zx}}\mathbf{W}^T\mathbf{A_p}\mathbf{R_{pr}}\mathbf{G_r}^{-1}(d\mathbf{U})] - \\
&\quad tr[(d\mathbf{U})\mathbf{R_{zx}}\mathbf{W}^T\mathbf{A_p}\mathbf{R_{pr}}\mathbf{G_r}^{-1}] - tr[\mathbf{R_{zx}}\mathbf{W}^T\mathbf{A_p}\mathbf{R_{pq}}\mathbf{B_q}\mathbf{R_{qp}}\mathbf{A_p}\mathbf{R_{pr}}\mathbf{G_r}^{-1}(d\mathbf{U})] - \\
&\quad tr[(d\mathbf{U})\mathbf{R_{zx}}\mathbf{W}^T\mathbf{A_p}\mathbf{R_{pq}}\mathbf{B_q}\mathbf{R_{qp}}\mathbf{A_p}\mathbf{R_{pr}}\mathbf{G_r}^{-1}] + tr[\mathbf{R_{zy}}\mathbf{V}^T\mathbf{B_q}\mathbf{R_{qp}}\mathbf{A_p}\mathbf{R_{pr}}\mathbf{G_r}^{-1}(d\mathbf{U})] + \\
&\quad tr[(d\mathbf{U})\mathbf{R_{zx}}\mathbf{W}^T\mathbf{A_p}\mathbf{R_{pq}}\mathbf{B_q}\mathbf{R_{qr}}\mathbf{G_r}^{-1}] + tr[\mathbf{R_{zx}}\mathbf{W}^T\mathbf{A_p}\mathbf{R_{pq}}\mathbf{B_q}\mathbf{R_{qr}}\mathbf{G_r}^{-1}(d\mathbf{U})] + \\
&\quad tr[(d\mathbf{U})\mathbf{R_{zy}}\mathbf{V}^T\mathbf{B_q}\mathbf{R_{qp}}\mathbf{A_p}\mathbf{R_{pr}}\mathbf{G_r}^{-1}] - tr[\mathbf{R_{zy}}\mathbf{V}\mathbf{B_q}\mathbf{R_{qr}}\mathbf{G_r}^{-1}(d\mathbf{U})] - \\
&\quad tr[(d\mathbf{U})\mathbf{R_{zy}}\mathbf{V}^T\mathbf{B_q}\mathbf{R_{qr}}\mathbf{G_r}^{-1}]\}
\end{aligned}
$$

As a result,

$$
\begin{aligned}
\frac{\partial log|\mathbf{G_r}|}{\partial \mathbf{U}} &= 2\mathbf{G_r}^{-1}[\mathbf{U}\mathbf{R_{zz}} - \mathbf{R_{rp}}\mathbf{A_p}\mathbf{W}\mathbf{R_{xz}} - \mathbf{R_{rp}}\mathbf{A_p}\mathbf{R_{pq}}\mathbf{B_q}\mathbf{R_{qp}}\mathbf{A_p}\mathbf{W}\mathbf{R_{xz}} + \\
&\quad \mathbf{R_{rp}}\mathbf{A_p}\mathbf{R_{pq}}\mathbf{B_q}\mathbf{V}\mathbf{R_{yz}} + \mathbf{R_{rq}}\mathbf{B_q}\mathbf{R_{qp}}\mathbf{A_p}\mathbf{W}\mathbf{R_{xz}} - \mathbf{R_{rq}}\mathbf{B_q}\mathbf{V}\mathbf{R_{yz}}] \qquad (25)
\end{aligned}
$$

Eqs. 23, 24, and 25 lead to:

$$
\begin{aligned}
\frac{\partial I(\mathbf{p};\mathbf{q};\mathbf{r})}{\partial \mathbf{U}} &= \mathbf{G_r}^{-1}[\mathbf{U}\mathbf{R_{zz}} - \mathbf{R_{rp}}\mathbf{R_{pp}}^{-1}\mathbf{W}\mathbf{R_{xz}} - \mathbf{R_{rp}}\mathbf{R_{pp}}^{-1}\mathbf{R_{pq}}\mathbf{H_q}\mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}\mathbf{W}\mathbf{R_{xz}} \\
&\quad +\mathbf{R_{rp}}\mathbf{R_{pp}}^{-1}\mathbf{R_{pq}}\mathbf{H_q}\mathbf{V}\mathbf{R_{yz}} + \mathbf{R_{rq}}\mathbf{H_q}\mathbf{R_{qp}}\mathbf{R_{pp}}^{-1}\mathbf{W}\mathbf{R_{xz}} - \mathbf{R_{rq}}\mathbf{H_q}\mathbf{V}\mathbf{R_{yz}}] \\
&\quad -\mathbf{H_r}[\mathbf{U}\mathbf{R_{zz}} - \mathbf{R_{rq}}\mathbf{R_{qq}}^{-1}\mathbf{V}\mathbf{R_{yz}}] - \mathbf{R_{rr}}^{-1}\mathbf{R_{rp}}\mathbf{H_p}[\mathbf{R_{pr}}\mathbf{R_{rr}}^{-1}\mathbf{U}\mathbf{R_{zz}} - \mathbf{W}\mathbf{R_{xz}}]
\end{aligned}
$$

## Acknowledgment

The authors wish to thank The University of Western Australia and King Fahd University of Petroleum and Minerals for their support. The authors also wish to thank the annonmous reviewers for their valuable comments.

## References

[1] P.A. Devijver and J. Kittler. *Pattern recognition: A statistical approach*. Prentice–Hall, 1982.

[2] E. Oja. Neural networks, principal components, and subspaces. *International journal of neural systems*, 1:61–68, 1989.

[3] L. Xu. Least mean square error reconstruction principle for self-organizing neural nets. *Neural Networks*, 6:627–648, 1993.

[4] J. Karhunen and J. Joutesnsalo. Generalizations of principal component analysis, optimization problems, and neural networks. *Neural Networks*, 8:549–562, 1995.

[5] Y. Miao and Y. Hua. Fast subspace tracking and neural network learning by a novel information criterion. *IEEE Transactions on Signal Processing*, 46:1967–1979, 1998.

[6] P.L. Lai and C. Fyfe. A neural implementation of canonical correlation analysis. *Neural Networks*, 12:1391–1397, 1999.

[7] E. van der Werf. *Non–linear target based feature extraction by diabolo networks*. PhD thesis, Delft University of Technology, 1999.

[8] R. Linsker. Self–organization in perceptual network. *IEEE Computer*, 21:105–117, 1988.

[9] S. Becker. Mutual information maximization: models of cortical self–organization. *Network: Comp. in neural sys*, 7:7–31, 1996.

[10] A.J. Bell and t.J. Sejnowski. An information–maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7:1004–1034, 1995.

[11] P. Comon. Independent component analysis, A new concept? *Signal Processing*, 36:287–314, 1994.

[12] J.W. Fisher III and J.C. Principe. A methodology for information theoretic feature extraction. In *World Congress on Computational Intelligence*, 1998.

[13] J.C. Principe, D. Xu, and J.W. Fisher III. Information theoretic learning. In Simon Haykin, editor, *Unspervised adaptive filtering*. Wiley, New York, NY, 2000.

[14] K.V. Mardia, J. Ken, and J. Bibby. *Multivariate analysis*. Academic Press, 1979.

[15] W.E. Larimore. Generalized canonical variate analysis of nonlinear systems. In *Proc. of the IEEE Conference on Decision and Control*, pages 1720–1725, 1988.

[16] J. Kay. Feature discovery under contextual supervision using mutual information. In *International Joint Conference on Neural Networks*, pages 79–84, 1992.

[17] J. Kay. Information theoretic neural networks for contextually guided unsupervised learning: Mathematical and statistical considerations. Technical report, University of Stirling, 1994.

[18] S. Becker. *An information–theroretic unsupervised learning algorithm for neural networks*. PhD thesis, University of Toronto, 1992.

[19] M.E. Tipping and C.M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

[20] P.F. Baldi and K. Hornik. Learning in linear neural networks: a survey. *IEEE Transactions on Neural Networks*, 6:837–858, 1995.

[21] J.R. Magnus. *Matrix differential calculus with applications in statistics and econometrics*. Chichester [England] ; New York : Wiley, 1988.

# List of figures

- "Fig. 17. Comparison between PCA and HIM ($\alpha = 1.0, \beta = 0.1$) in terms of $I(\mathbf{p}; \mathbf{q}; \mathbf{r})$, using 3-channel EEG data, (CCA does not apply)"
- "Fig. 18. Comparison between PCA and HIM ($\alpha = 1.0, \beta = 0.1$) in terms of $I(\mathbf{p}; \mathbf{q}; \mathbf{r}; \mathbf{s}; \mathbf{t})$, using 5-channel EEG data, (CCA does not apply)"
- "Fig. 19. Comparison between PCA, CCA, and HIM ($\alpha = 1.0, \beta = 0.1$) in terms of classification accuracy (average accuracy of 2-channel Speech data)"

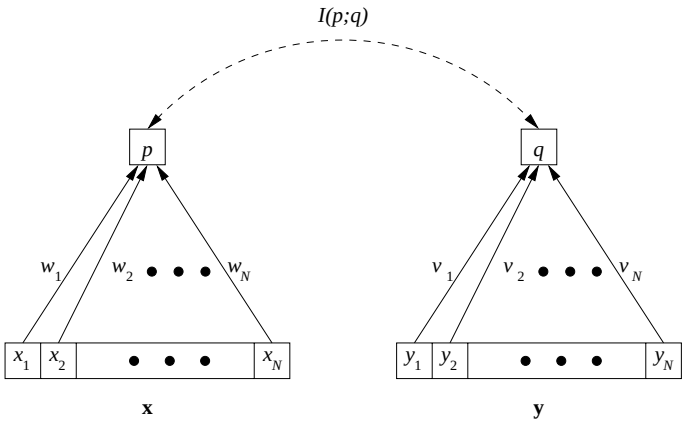


Figure 1:

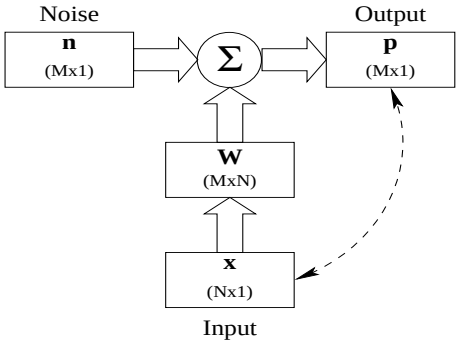


Figure 2:

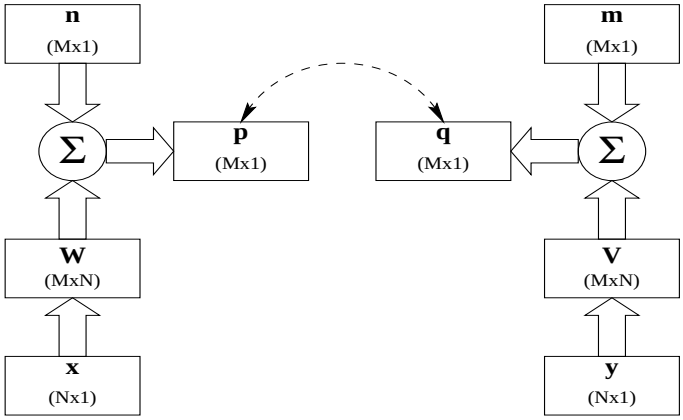


Figure 3:

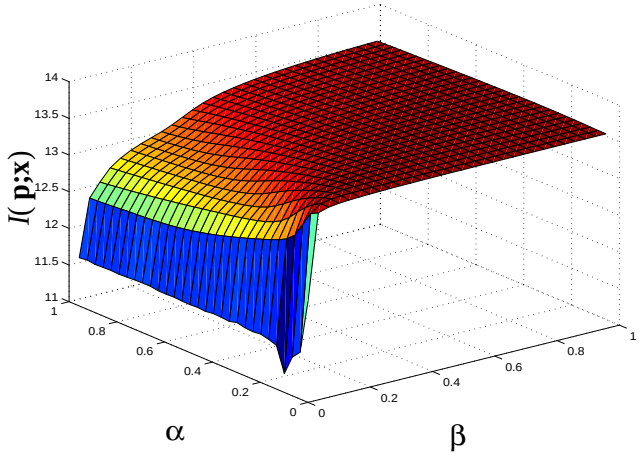


Figure 4:

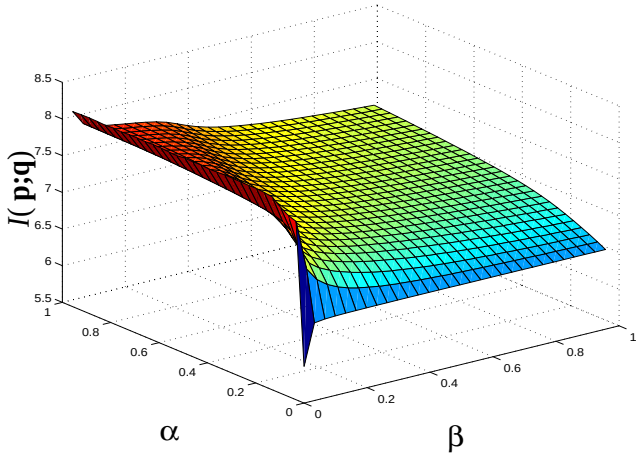


Figure 5:

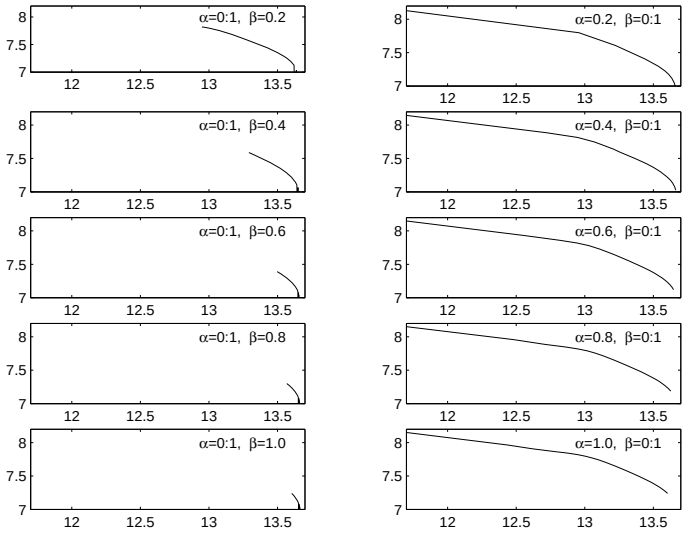


Figure 6:

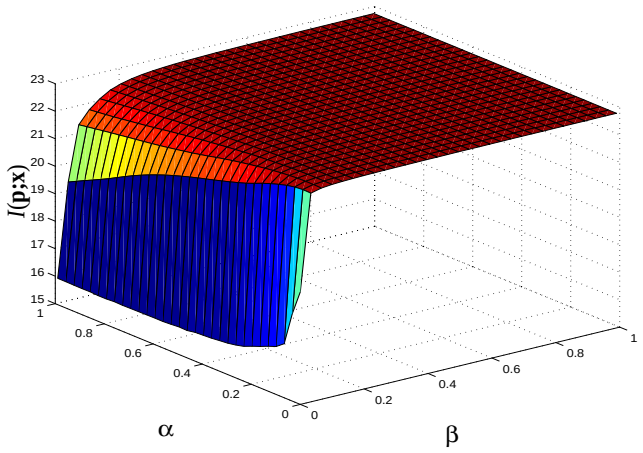


Figure 7:

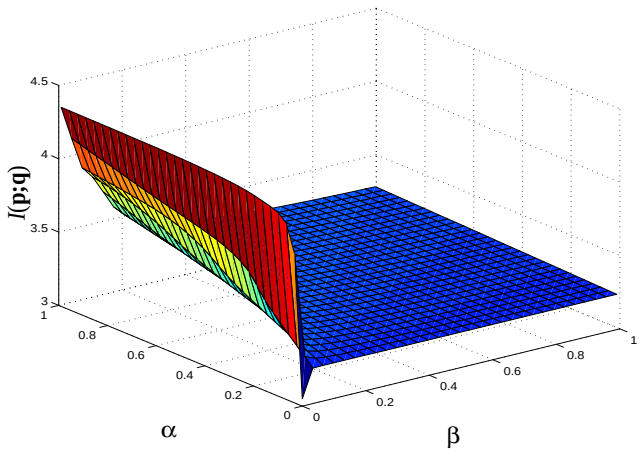


Figure 8:

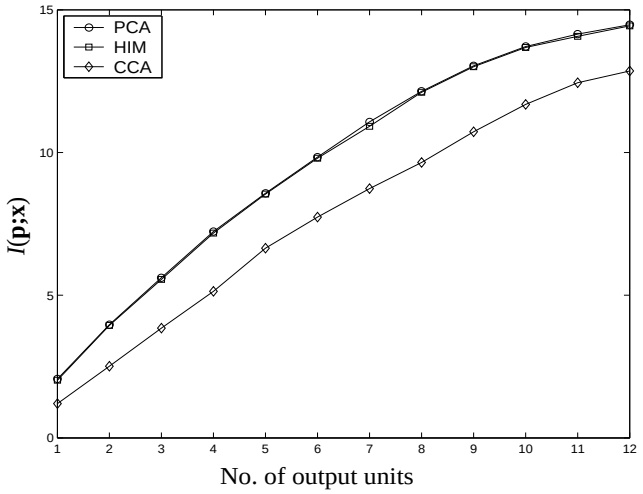


Figure 9:

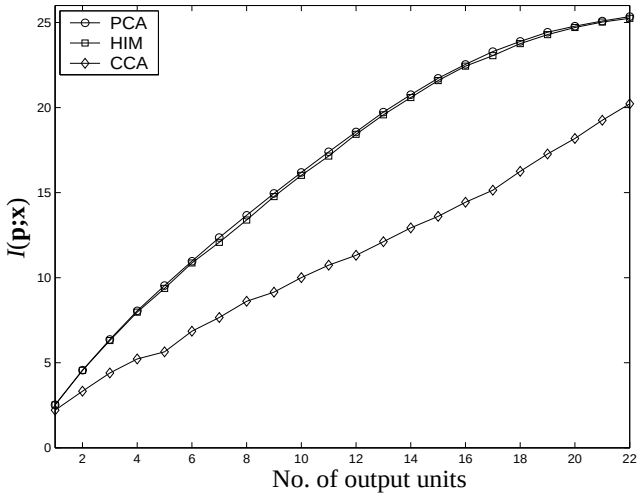


Figure 10:

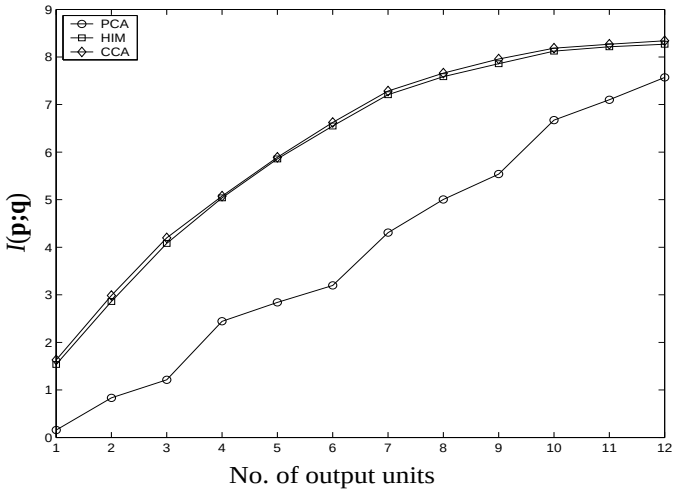


Figure 11:

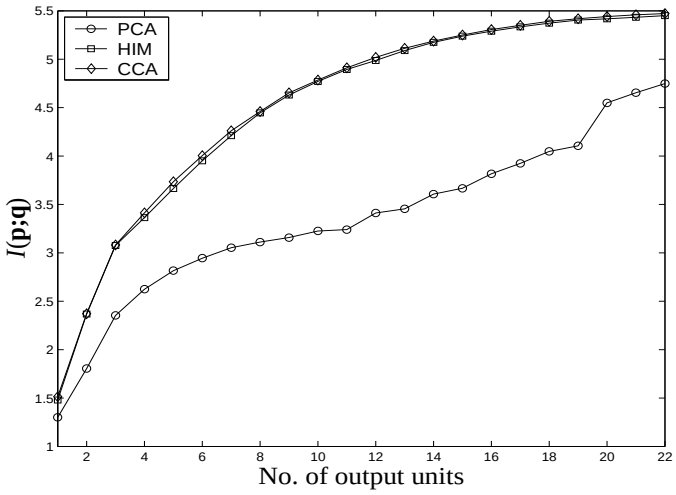


Figure 12:

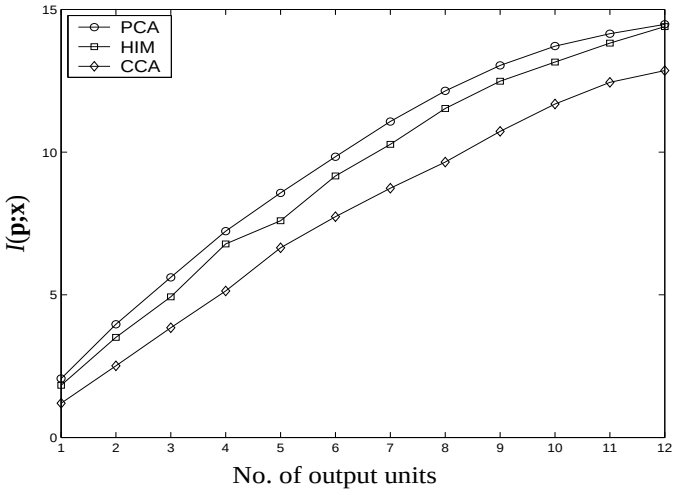


Figure 13:

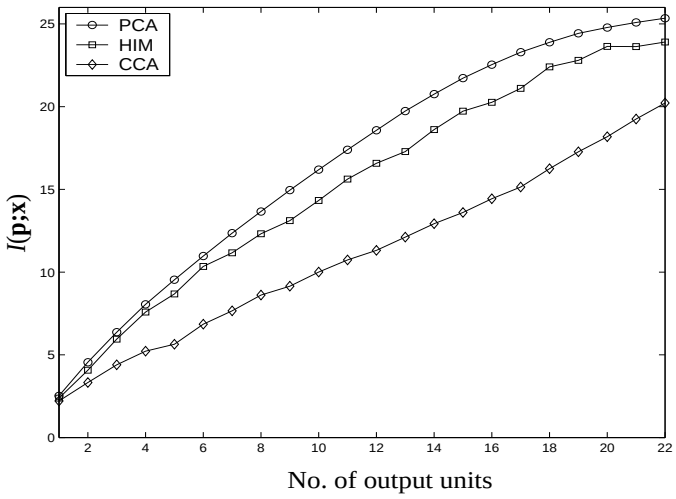


Figure 14:

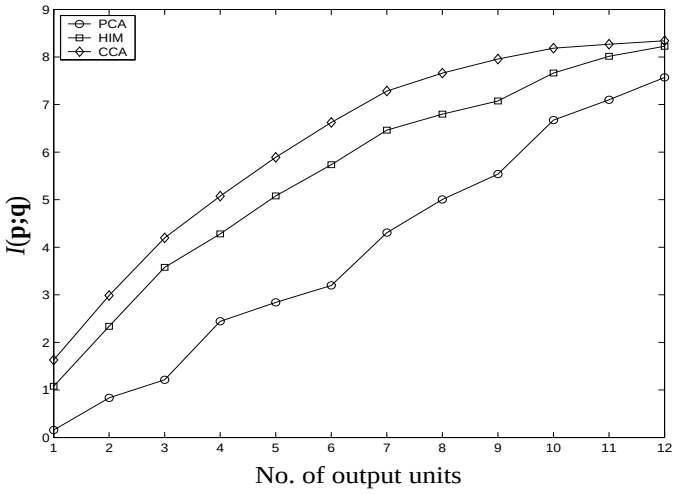


Figure 15:

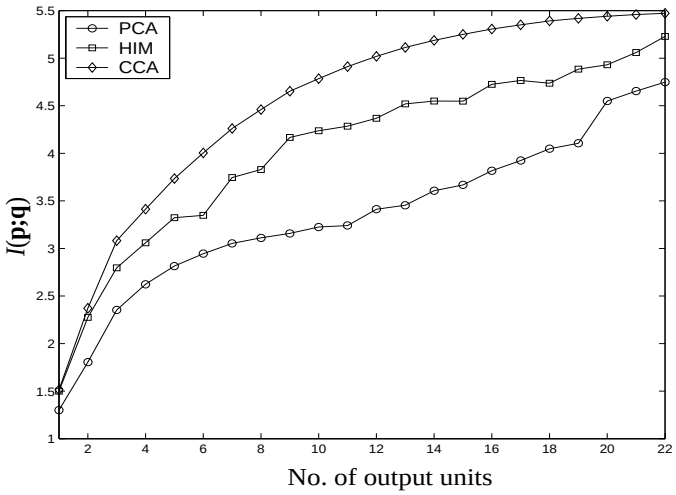


Figure 16:

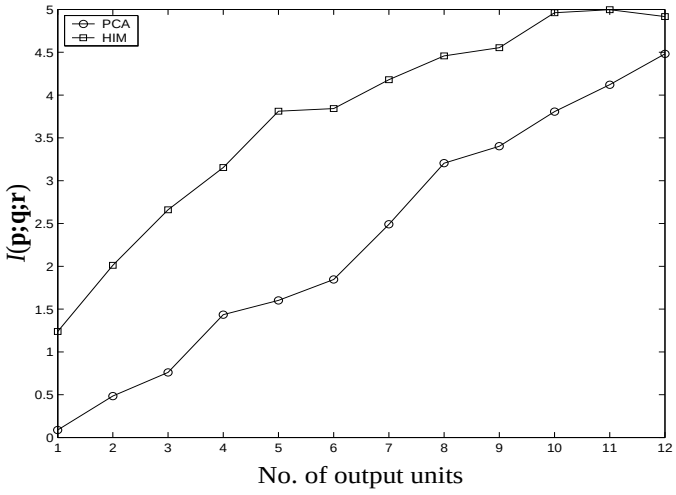


Figure 17:

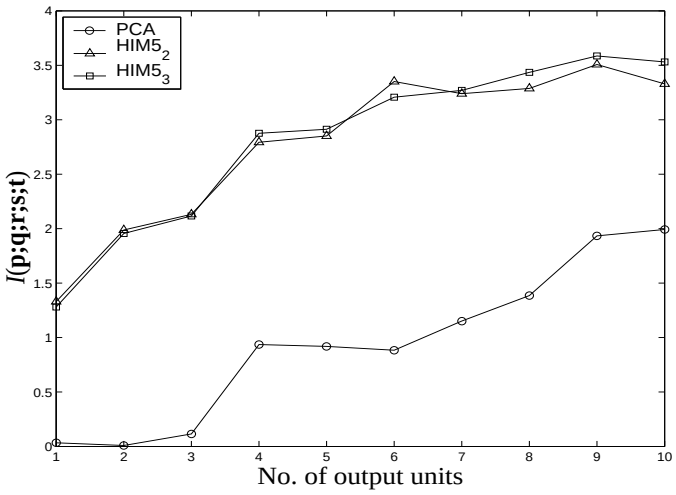


Figure 18:

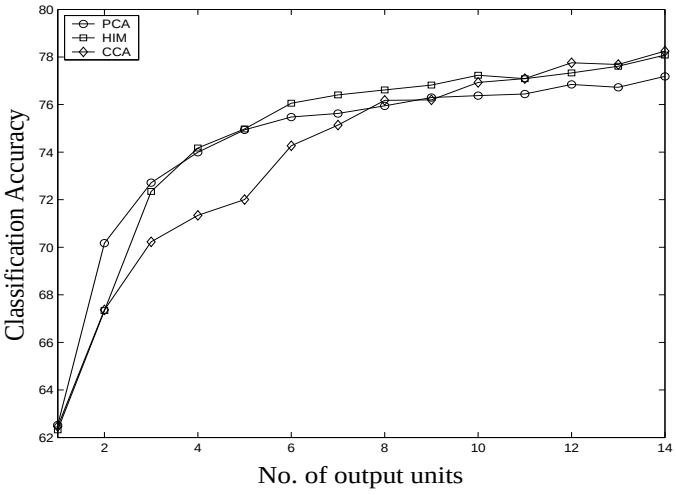


Figure 19: