# Extracting online information from Dual and Multi Data Streams

Zeeshan Khawar Malik, Amir Hussain and Jonathan Wu[1]
University of Stirling, Scotland, UK
and
University of Windsor, Ontario, Canada[1].
email:zkm,ahu@cs.stir.ac.uk and jwu@uwindsor.ca[1]

**Abstract**

In this paper, we consider the challenging problem of finding shared information in multiple data streams simultaneously. The standard statistical method for doing this is the well-known canonical correlation analysis (CCA) approach. We begin by developing an online version of the CCA and apply it to reservoirs of an Echo State Network (ESN) in order to capture shared temporal information in two data streams. We further develop the proposed method by forcing it to ignore shared information that is created from static values using derivative information. We finally develop a novel multi-set CCA method which can identify shared information in more than two data streams simultaneously. The comparative effectiveness of the proposed methods is illustrated using artificial and real benchmark data sets.

## 1 Introduction

In this paper, we consider the problem of extracting common information from 2 or more data streams simultaneously. The standard statistical technique for identifying common structure in 2 data streams is known as Canonical Correlation Analysis (CCA). We first employ an existing online method for solving the generalized eigenproblem in order to solve the standard CCA problem. However, we are interested in extracting canonical information from two streams of time series.

$$\begin{array}{l} abbabc***cdcdcda*cacdcd \\ \text{Direction of time} \longrightarrow \\ abb**abccd**cdcdaca*****cdcd \end{array} \tag{1}$$

Series (1) shows a particular temporal pattern (*abbabccdcdcdacacdcd*) from an alphabet of 4 symbols which is to be found in two distinct time series. However

both series contain stray values which are not part of the pattern (shown by '*'s in the figure to indicate these are sections whose value we don't know or care about). There is a clear and direct relationship between the two time series but to identify automatically the relevant pattern, we would typically require a technique such as dynamic time warping [16].

We are interested in a generalization of this problem in which the relationship between the elements of the time series is not as direct as the above but in fact can be characterized by finding the canonical variates of the two time series. However as can be seen from the above, a direct method will fail since the corresponding pairs do not necessarily appear at the same time instant. We therefore use the technique of reservoir computing to get a representation of the time series which contains information about the history of the time series. Thus, for example, at position 11 in the time series (1), the partial pattern, *abbabccd*, will exist in the reservoir's representation, albeit mixed with a representation of the don't care values, '*'.

Reservoir computing [17] [8] is a relatively new artificial neural network technique for processing temporal data. There are two main strands, the Echo State Network [23] and the Liquid State Machine [22], each of which has a set of recurrent connections [27], forming the reservoir, which have fixed strength or weight. In our experiments, we consider the problem of extracting information from time series data streams and so we use the activations from two or more reservoirs, each corresponding to one data stream, as the inputs to our generalized CCA method. The resulting technique is known as Temporal CCA.

We then, again in an online manner, remove from the data streams, data which does not change much in the individual streams. We do this using the derivative information from the data streams, similar to the technique for Slow Feature Analysis [26]. The resulting method is known as High Variance CCA. We combine this with the Temporal CCA to create a Temporal High Variance CCA algorithm.

Finally, we consider the extraction of information from more than 2 data sets simultaneously: Multi-set CCA. This is easily encompassed by our generalized eigenproblem method and can be combined with the other techniques to give Multi-set Temporal High Variance CCA.

## 2 Background

Canonical Correlation Analysis is a well-known technique since its first formulation [10] used primarily for finding filters between two streams of data in a way that correlation between those two filters get maximized. This is now a standard technique in the data analysis repertoire.

One of us has previously been interested in creating online incremental versions of CCA [6, 7, 14, 25] and very recently Laplacian Eigenmaps [32], often based on artificial neural networks. Some of these techniques involved minimizing the squared difference between the outputs of two twinned neural networks [14], each devoted to one of the data streams. This was done by gradient de-

scent. This was extended in [25] to gradient descent on the Bregman divergences between the two data streams; in that paper too, we used reservoirs for preprocessing but we did not consider time series and the reservoir was used merely to give a nonlinear representation of the data.

Recently [29] has proposed an adaptive formulation of the classical CCA algorithm based on matrix manifolds. The authors solved the optimization problem on matrix manifolds using classical gradient algorithm and designed the adaptive CCA algorithm based on rationale that the algorithm should be capable enough to detect the exact time stamp when change occurs in the subspace.

An incremental approach based on the recursive least square algorithm have also been proposed [24] for rank-one CCA problem that can cope with multiple orthogonal projections using a deflation scheme. The emphasis in this paper is the extension of CCA to cope with more than two data sets simultaneously; the authors perform a valuable comparative study of various algorithms but do not consider kernel CCA nor specific time series adaptations.

Many researchers have contributed in a different manner to further maximize the correlations by introducing kernels [14, 1], and very recently a temporal kernel CCA approach was proposed [2] in which a novel method based on kernels were introduced which computes multivariate temporal filters between data sources containing different dimensionality and temporal resolutions. The core idea behind using kernels is that the raw data is transformed to get a representation of the data in an implicit high-dimensional latent space. We say implicit because usually the actual representation in this space is not used since the kernel trick enables us to manipulate algorithms by using only the dot product of the implicit representations i.e. if we know these dot products, we never need to investigate the actual representations themselves. These methods are especially useful if we have a relatively small number of high dimensional samples; it is also useful in that a nonlinear problem in the data space is converted to a linear problem in the latent space. One problem which is particularly important to address in kernel CCA is the problem of overfitting and so some form of regularization is often employed. An application of kernel CCA is shown in [9]; this deals with the problem of learning semantics of multimedia content by combining image and text data. The nature of the data is such that the authors propose to use a sparse version of KCCA.

Many researchers have also contributed other effective processing techniques for both textual and numeric data. In [21] the authors have presented a new framework for effective commonsense reasoning by taking into account a number of correlation-based similarity measures, including point-wise mutual information and emotional affinity. Similarly in [3] the authors have proposed an ELM-based emotion categorization architecture that is able to remap any concept represented according to the affectivespace into a suitable space defined by four affective dimensions which includes pleasantness, attention, sensitivity and aptitude. Both the proposed methods in [21] and [3] shows how an ensemble of concept-based sentiment analysis and machine learning techniques could emulate the cognitive process of affective analogical reasoning; in order to quickly,

dynamically and effectively infer semantic and sentics associated with natural language concepts. In [5] and [28] the authors have exploited the standard extreme learning machine (ELM) [11] method, one of the most popular and fastest methods used for classification nowadays. In [5] the authors have introduced a novel random projection based model for ELM that helped in reducing the number of hidden neurons without effecting the generalization performance in prediction accuracy. Similarly, in [28] the authors have proposed a parent-offspring progressive learning method that works by separating the data points into various parts, and then multiple extreme learning machine learn and identify the clustered parts separately. This helps in improving the generalization performance, but at the same time decreases the computational efficiency too.

Time Series information have also been considered for finding the temporal structure in a correlation framework [4, 30]. However [4] considered autocorrelation within a single data stream rather than correlation between 2 (or more) data streams. [30] uses an extension of reverse-correlation methods based on canonical correlation analysis in order to identify the properties of receptive fields of a group of neurons; they also capture nonlinear stimulus-response relationships using kernel canonical correlation analysis. We will directly attack the problem of identifying correlations in two or more data streams where each data stream corresponds to a single time series. As our toy example in Section 1 illustrates, in real data sets we cannot guarantee that the temporal features of 2 time series march in step and thus we use reservoir activations to maintain a representation of historical values of the time series.

The remainder of this paper is organized as follows: Section 3 discusses the method, which we will use for solving eigenproblems. Section 4 reviews echo state networks. In section 5, we discusses existing CCA for dual streams by solving the generalized eigenproblem and our new Temporal CCA method. In Section 6 we further exploited our new Temporal CCA method and derive new High Variance CCA method (HVCCA) and Temporal High Variance CCA method. In Section 7 we continue to focus on multiple streams of data and derive enhanced versions of our Temporal CCA method, High Variance CCA and Temporal High Variance CCA method for multiple streams of data. Finally in Section 8 we apply all our newly proposed MCCA-based methods to MNIST digit dataset and compared the results.

# 3 Generalized Eigenproblems: incremental solutions

[31] show that one method of finding the maximum eigenvalue of the generalised eigenproblem

$$A\mathbf{w} = \lambda B\mathbf{w}, \tag{2}$$

is to iteratively use

$$\begin{aligned} \Delta\mathbf{w} &= A\mathbf{w} - f(\mathbf{w})B\mathbf{w}, \\ \mathbf{w} &= \mathbf{w} + \eta\Delta\mathbf{w}, \end{aligned} \tag{3}$$

where $\eta$ is a learning rate or step size. The function $f(\mathbf{w}) : R^n - \{0\} \rightarrow R$ satisfies

1. $f(\mathbf{w})$ is locally Lipschitz continuous
2. $\exists M_1 > M_2 > 0 : f(\mathbf{w}) > \lambda_1, \forall \mathbf{w} :\| \mathbf{w} \| \geq M_1$ and$f(\mathbf{w}) < \lambda_n, \forall \mathbf{w} : 0 <\| \mathbf{w} \| \leq M_2$
3. $\forall \mathbf{w} \in R^n - \{0\}, \exists N_1 > N_2 > 0 : f(\theta \mathbf{w}) > \lambda_1, \forall \theta : \theta \geq N_1$ and$f(\theta \mathbf{w}) < \lambda_n, \forall \theta : 0 \leq \theta \leq N_2$ and $f(\theta \mathbf{w})$ is a strictly monotonically increasing function of $\theta$ in $[N_1, N_2]$.

where $\lambda_1$ is the greatest generalised eigenvalue and $\lambda_n$ is the least eigenvalue.

Intuitively, what these criteria mean are that

1. The function is rather smooth.
2. It is always possible to find values of $\mathbf{w}_i, i = 1, 2$ large enough so that the functions of the weights exceed the greatest eigenvalue.
3. It is always possible to find values of $\mathbf{w}_i, i = 1, 2$ small enough so that the functions of the weights are smaller than the least eigenvalue.
4. For any particular value of $\mathbf{w}_i, i = 1, 2$, it is possible to multiply $\mathbf{w}_i, i = 1, 2$ by a scalar and apply the function to the result to get a value greater than the greatest eigenvalue.
5. Similarly, we can find another scalar so that, multiplying the $\mathbf{w}_i, i = 1, 2$, by this scalar and taking the function of the result gives us a value less than the smallest eigenvalue.
6. The function of this product is monotonically increasing between the scalars defined in 4 and 5.

This method has been used in [6, 7] to perform extensions of canonical correlation analysis. In this paper we have further used this method to extend canonical correlation analysis and performed multi-set canonical correlation using our extended approach.

# 4 Echo State Network

Echo State Network [17] is one of the most popular types in Reservoir Computing. Echo State Networks are mainly composed of three layers of 'neurons': an input layer which is connected with random and fixed weights to the next layer which forms the reservoir. The neurons of the reservoirs are connected with each other with a fixed, random, sparse matrix of weights. Normally only 10% of the weights in the reservoirs are non-zero. The weights from reservoir to the output neurons are trained using error descent. Only weights from the reservoirs to

the output are trainable and this makes reservoir really very efficient and easily trained.

We first define the idea of reservoir. $W_{in}$ indicates the weight from the $N_u$ inputs $\mathbf{u}$ to the $N_x$ reservoir units $\mathbf{x}$, $W$ indicates the $N_x \times N_x$ reservoir weight matrix, and $W_{out}$ indicates the $(N_x + 1) \times N_y$ weight matrix connecting the reservoir units to the output units, denoted by $\mathbf{y}$. Typically $N_x \gg N_u$. $W_{in}$ is fully connected with the neurons inside the reservoirs and fixed (i.e the weights are non-trainable). $W$ is also fixed. $W_{out}$ is fully connected and the weights are trainable.

The network's dynamics are governed by

$$\mathbf{x}(t) = f(W_{in}\mathbf{u}(t) + W\mathbf{x}(t-1)), \tag{4}$$

where $f(.) = \tanh(.)$ and $t$ is the time index. The feed forward stage is given by

$$\mathbf{y} = W_{out}\mathbf{x}. \tag{5}$$

This is followed by supervised learning of the output weights, $W_{out}$. A simple least mean square method is used for online learning which gives

$$W_{out} = W_{out} + \eta(\mathbf{y}_{target} - \mathbf{y})\mathbf{x}^T, \tag{6}$$

where $\eta$ is a learning rate (step size) and $\mathbf{y}_{target}$ is the target output corresponding to the current input.

In this paper, we will maintain the first two sets of weights fixed but update the third set, $W_{out}$, using methods suggested by canonical correlation analysis.

# 5 Dual data streams

Consider the problem of extracting information from two data streams simultaneously when these data streams contain information about each other which may be used to assist with on-going information gathering. These methods may be useful in a number of cases: for example we may be seeing the same underlying signal through different sensors which will often happen with the various scans of the human brain and heart. We may be examining the correlation between different signals when there is an underlying hidden reason for the different signals.

We envisage first the situation where we have two sets of data samples, $\mathbf{x}_1$ and $\mathbf{x}_2$, which are then passed through a set of weights, $\mathbf{w}_1$ and $\mathbf{w}_2$, to give outputs $y_1 = \mathbf{w}_1^T\mathbf{x}_1$ and $y_2 = \mathbf{w}_2^T\mathbf{x}_2$. We will adjust the weights $\mathbf{w}_1$ and $\mathbf{w}_2$ to optimise the CCA criterion.

It may be shown that one method [7] of finding the canonical correlation directions is to solve the generalised eigenvalue problem

$$\begin{bmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = \rho \begin{bmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}, \tag{7}$$

where $\Sigma_{ij}$ is the covariance matrix between $\mathbf{x}_i$ and $\mathbf{x}_j$.

Using this formulation we have previously [6] shown that the canonical correlation directions $\mathbf{w}_1$ and $\mathbf{w}_2$ may be found using

$$\begin{aligned}\frac{d\mathbf{w}_1}{dt} &= \Sigma_{12}\mathbf{w}_2 - f(\mathbf{w})\Sigma_{11}\mathbf{w}_1, \\ \frac{d\mathbf{w}_2}{dt} &= \Sigma_{21}\mathbf{w}_1 - f(\mathbf{w})\Sigma_{22}\mathbf{w}_2.\end{aligned}$$

Using the fact that $\Sigma_{ij} = E(\mathbf{x}_i\mathbf{x}_j^T), i, j = 1, 2$,where T denotes the transpose, we derive the instantaneous versions

$$\begin{aligned}\Delta\mathbf{w}_1 &= \eta(\mathbf{x}_1 y_2 - f(\mathbf{w})\mathbf{x}_1 y_1), \\ \Delta\mathbf{w}_2 &= \eta(\mathbf{x}_2 y_1 - f(\mathbf{w})\mathbf{x}_2 y_2),\end{aligned}$$

which was shown to provide a family of networks capable of performing CCA.

However canonical correlation analysis is a linear method. It is more interesting to consider methods by which we may find temporal relationships between pairs of data sets. We may use the above method but use the reservoir activations for a pair of related time series and $W^1_{out}$ and $W^2_{out}$ in place of $\mathbf{w}_1$ and $\mathbf{w}_2$.

We can use the reservoirs to extract information from two data streams simultaneously: we simply have two reservoirs with fixed weights between inputs and reservoirs and fixed weights internal to the reservoirs but have two sets of trainable weights which are simultaneously adjusted so that they learn to predict each other's output as shown in figure 1.

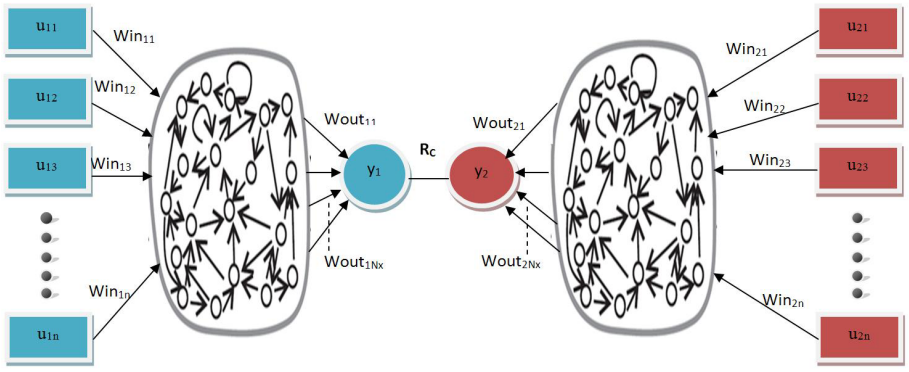


Figure 1: Dual Reservoir Streams

Thus we have simultaneously for paired inputs $\mathbf{u}_1$ and $\mathbf{u}_2$,

$$\begin{aligned}
\mathbf{x}_1(t) &= f(W_{in}^1 \mathbf{u}_1(t) + W^1 \mathbf{x}_1(t-1)), \\
\mathbf{x}_2(t) &= f(W_{in}^2 \mathbf{u}_2(t) + W^2 \mathbf{x}_2(t-1)), \\
y_1 &= W_{out}^1 \mathbf{x}_1, \\
y_2 &= W_{out}^2 \mathbf{x}_2, \\
\Delta W_{out}^1 &= \eta(\mathbf{x}_1 y_2 - f(W_{out}^1)\mathbf{x}_1 y_1), \\
\Delta W_{out}^2 &= \eta(\mathbf{x}_2 y_1 - f(W_{out}^2)\mathbf{x}_2 y_2).
\end{aligned} \tag{8}$$

The resulting technique is Temporal CCA though clearly it can be used with e.g. image data where the relationship between subsequent pixels or lines is spatial rather than temporal.

## 5.1 Artificial Data

We illustrate on an artificial data set which has two related sources but the relation is maximised by discovering a temporal mapping. Let $\mathbf{u}_1 = \{u_1(1), u_1(2)\}$ and $\mathbf{u}_2 = \{u_2(1), u_2(2)\}$. Then our artificial data set has

$$\begin{aligned}
u_1(1) &= \sin(t), \\
u_1(2) &= \cos(t), \\
u_2(1) &= t, \\
u_2(2) &= \tanh(t),
\end{aligned} \tag{9}$$

where $t$ increases from $-\pi$ to $\pi$ in steps of 0.01. We created a 2-dimensional input vector and thus we have 1000 samples. In our experiment the learning rate was 0.0001 and the number of iterations was 10000. We get a temporal correlation of 0.85 whereas the standard linear non-temporal value was 0.623 [13].

## 5.2 Real Data

In order to compare our proposed method with those reported earlier we use data taken from [19]. The data set consists of 88 students who sat 5 exams, 2 of which were closed book exams while the other 3 were open book exams. Thus each student comprises a single sample over the two data sets and we have a two dimensional $\mathbf{u}_1$ (the closed book marks) and a three dimensional $\mathbf{u}_2$ (the open book marks). Since we are investigating a temporal technique, we must ensure that the students were presented in a specific order: we used the average mark over the 5 examinations and sampled the students in descending order from highest overall mark to lowest. In our experiment, the learning rate was 0.0001, the size of reservoir was 50 and the number of iterations were 50000. The temporal correlation on student's data is 0.7687925 whereas the linear correlation using the standard statistical technique was 0.6630.

# 6 Extracting High Variance Features

We have, in a sister paper [18], used the incremental solution of the generalised eigenproblem on the slow feature analysis criterion [26]. This tries to identify invariances in a data set and is based on the idea that we wish to find the minimal eigenvalue of the covariance of a (single stream) data set while maximising the eigenvalue of the covariance of the derivatives. This suggests a twist to standard CCA: what we wish is to maximise the cross covariance while keeping constant the variance within each data stream *while simultaneously* maximising the rate of change of variances within each data set. Intuitively this is saying that we are less interested in correlations which are based on constant values than correlations which occur when the rate of change is higher.

This is implemented as

$$\left(\begin{bmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{bmatrix}\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = \rho\left(\begin{bmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{bmatrix} - \begin{bmatrix} \Sigma_{\dot{1}\dot{1}} & 0 \\ 0 & \Sigma_{\dot{2}\dot{2}} \end{bmatrix}\right)\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix},$$

where $\Sigma_{ij}$ is the covariance matrix and $\Sigma_{\dot{i}\dot{j}}$ is the covariance of derivatives of the data with respect to time.

This can also be written as

$$\begin{bmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{bmatrix}\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = \rho\begin{bmatrix} \Sigma_{11} - \Sigma_{\dot{1}\dot{1}} & 0 \\ 0 & \Sigma_{22} - \Sigma_{\dot{2}\dot{2}} \end{bmatrix}\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}. \tag{10}$$

The method of finding canonical correlation directions $\mathbf{w}_1$ and $\mathbf{w}_2$ would be then

$$\begin{aligned} \frac{d\mathbf{w}_1}{dt} &= \Sigma_{12}\mathbf{w}_2 - f(\mathbf{w})(\Sigma_{11} - \Sigma_{\dot{1}\dot{1}})\mathbf{w}_1, \\ \frac{d\mathbf{w}_2}{dt} &= \Sigma_{21}\mathbf{w}_1 - f(\mathbf{w})(\Sigma_{22} - \Sigma_{\dot{2}\dot{2}})\mathbf{w}_2. \end{aligned}$$

Using the fact that $\Sigma_{ij} = E(\mathbf{x}_i\mathbf{x}_j^T), i, j = 1, 2$ and that $y_1 = \mathbf{w}_1.\mathbf{x}_1$, we may propose the instantaneous rules

$$\begin{aligned} \Delta\mathbf{w}_1 &= \eta(\mathbf{x}_1 y_2 - f(\mathbf{w})((\mathbf{x}_1 y_1) - \left((\frac{d\mathbf{x_1}}{dt})(\frac{d\mathbf{x_1}}{dt}^T)\right)\mathbf{w}_1)), \\ \Delta\mathbf{w}_2 &= \eta(\mathbf{x}_2 y_1 - f(\mathbf{w})((\mathbf{x}_2 y_2) - \left((\frac{d\mathbf{x_2}}{dt})(\frac{d\mathbf{x_2}}{dt}^T)\right)\mathbf{w}_2)). \end{aligned}$$

In practice, to estimate $\frac{d\mathbf{x}}{dt}|_\tau$, we used $\mathbf{x}(\tau+1) - \mathbf{x}(\tau)$, where $\mathbf{x}(\tau)$ is the value of $\mathbf{x}$ at time $\tau$.

We have also tried to minimize the rate of change from both the covariance and cross-covariance between datasets X and Y but only maximizing the covariance from within the datasets gives better results.

This technique is useful to find CCA for moving objects inside 2 or more images by extracting only high variance features where the rate of change is maximum and calculating CCA. This technique we call High Variance CCA, HVCCA.

## 6.1 Real Data

In order to compare our proposed method with those reported earlier we use the student exam data [19]. The correlation vectors of the new and previous methods which includes that standard statistical method and the one reported in [6] are shown in Table 1. The learning rate was 0.0001 and the number of iterations was 50000.

| | | |
|---|---|---|
| 1 | Standard Statistics Maximum Correlation<br>$\mathbf{w}_1$(0.0260 0.0518)<br>$\mathbf{w}_2$(0.0824 0.00081 0.0035) | 0.6630 |
| 2 | Existing Neural Network Maximum Correlation<br>$\mathbf{w}_1$ (0.0270 0.0518)<br>$\mathbf{w}_2$ (0.0810 0.0090 0.0040) | 0.6790 |
| 3 | New Neural Network Maximum Correlation<br>$\mathbf{w}_1$ (0.026 0.0518)<br>$\mathbf{w}_2$ (0.0609 0.0084 0.0042) | 0.68125 |

Table 1: Correlations and Weights of Real Data Experiment

Note that we are not using the reservoir to pre-process the data at this stage and note also that we are not achieving as high a correlation as previously when we were using the reservoir: this is only High Variance CCA.

## 6.2 Real Images

In order to compare our method with those reported earlier we take the data from two similar images. The datasets will be taken by extracting first 150 pixels from both the images of 150 dimensions each. This means that both the datasets are of equal length consisting of 150 rows and 150 columns each. The learning rate was 0.0001 and the number of iterations were 50000. The experiment is conducted on the previous method reported in [6]. The images which are used for this particular experiment are shown below. The first 150 $\times$ 150 chunk of pixel data is read from both the images. The results are displayed in Table 2.

| | |
|---|---|
| Existing Neural Network Maximum Correlation | 0.7833631 |
| High Variance CCA | 0.7935415 |

Table 2: Correlations of Real Image Data Experiment

Note again that we are not using the reservoirs in this section.

Figure 2: Two Real Images Used in the Experiment

### 6.3 Temporal High Variance CCA

We may use the above High Variance method but use the reservoir activations for a pair of related times series and $W^1_{out}$ and $W^2_{out}$ in place of $\mathbf{w_1}$ and $\mathbf{w_2}$.

We illustrate on an artificial data set which has two related sources but the relation is maximised by discovering a nonlinear mapping. Let $\mathbf{u}_1 = \{u_1(1), u_1(2)\}$ and $\mathbf{u}_2 = \{u_2(1), u_2(2)\}$. Then our artificial data set has

$$\begin{aligned} u_1(1) &= \sin(t), \\ u_1(2) &= \cos(t), \\ u_2(1) &= t, \\ u_2(2) &= \tanh(t), \end{aligned} \tag{11}$$

where $t$ increases from $-\pi$ to $\pi$ in steps of 0.01. The learning rate was 0.0001 and the number of iterations was 10000. The size of the reservoir is equal to 50. We get correlations of 0.87 which may be compared with a correlation of 0.85 with the online CCA method of section 5.

## 7 Multi-Set Canonical Correlation Analysis

Multi-set Canonical Correlation Analysis (MCCA)[12, 20] is a technique through which we can analyse linear relationship between more (than 2) sets of variables. It is considered as a generalized extension of CCA in essence.

Consider firstly three variables $\mathbf{x}_1$,$\mathbf{x}_2$ and $\mathbf{x}_3$ the method for finding canonical correlations of these three variables can be extended for $n$ terms easily. These three variables are then passed through a set of weights, $\mathbf{w}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$ to give outputs $y_1 = \mathbf{w}_1^T\mathbf{x}_1$, $y_2 = \mathbf{w}_2^T\mathbf{x}_2$ and $y_3 = \mathbf{w}_3^T\mathbf{x}_3$.

The criteria for finding Multi-set canonical correlations of three variables will be to find the greatest eigenvalue of:

$$\begin{bmatrix} 0 & \Sigma_{12} & \Sigma_{13} \\ \Sigma_{21} & 0 & \Sigma_{23} \\ \Sigma_{31} & \Sigma_{32} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{bmatrix} = \rho \begin{bmatrix} \Sigma_{11} & 0 & 0 \\ 0 & \Sigma_{22} & 0 \\ 0 & 0 & \Sigma_{33} \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{bmatrix}, \tag{12}$$

where $\Sigma_{ij}$ is the covariance matrix between $\mathbf{x}_i$ and $\mathbf{x}_j$.

The canonical correlation directions $\mathbf{w}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$ may be found using

$$\begin{aligned}
\frac{d\mathbf{w}_1}{dt} &= \Sigma_{12}\mathbf{w}_2 + \Sigma_{13}\mathbf{w}_3 - f(\mathbf{w})\Sigma_{11}\mathbf{w}_1, \\
\frac{d\mathbf{w}_2}{dt} &= \Sigma_{21}\mathbf{w}_1 + \Sigma_{23}\mathbf{w}_3 - f(\mathbf{w})\Sigma_{22}\mathbf{w}_2, \\
\frac{d\mathbf{w}_3}{dt} &= \Sigma_{31}\mathbf{w}_1 + \Sigma_{32}\mathbf{w}_2 - f(\mathbf{w})\Sigma_{33}\mathbf{w}_3.
\end{aligned}$$

As before, we may derive the instantaneous versions

$$\begin{aligned}
\Delta\mathbf{w}_1 &= \eta\mathbf{x}_1(y_2 + y_3 - f(\mathbf{w})y_1), \\
\Delta\mathbf{w}_2 &= \eta\mathbf{x}_2(y_1 + y_3 - f(\mathbf{w})y_2), \\
\Delta\mathbf{w}_3 &= \eta\mathbf{x}_3(y_1 + y_2 - f(\mathbf{w})y_3).
\end{aligned}$$

The generalized Multi-set CCA criteria for $n$ terms is given as

$$\begin{bmatrix} 0 & \Sigma_{12} & \Sigma_{13} & ... & \Sigma_{1n} \\ \Sigma_{21} & 0 & \Sigma_{23} & ... & \Sigma_{2n} \\ \Sigma_{31} & \Sigma_{32} & 0 & ... & \Sigma_{3n} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ \Sigma_{n1} & \Sigma_{n2} & \Sigma_{n3} & ... & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \\ . \\ . \\ . \\ \mathbf{w}_n \end{bmatrix} = \rho \begin{bmatrix} \Sigma_{11} & 0 & 0 & ... & 0 \\ 0 & \Sigma_{22} & 0 & ... & 0 \\ 0 & 0 & \Sigma_{33} & ... & 0 \\ . & . & . & ... & . \\ . & . & . & ... & . \\ 0 & 0 & 0 & ... & \Sigma_{nn} \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \\ . \\ . \\ . \\ \mathbf{w}_n \end{bmatrix}, \tag{13}$$

from which we get the obvious generalisation

$$\Delta\mathbf{w}_i = \eta\mathbf{x}_i(\sum_{j\neq i} y_j - f(\mathbf{w})y_i). \tag{14}$$

## 7.1 Artificial Data

We illustrate on an artificial data set which has three related sources but the relation is maximised by discovering a linear relationship among the three datasets. Let $\mathbf{u}_1 = \{u_1(1), u_1(2)\}$,$\mathbf{u}_2 = \{u_2(1), u_2(2)\}$ and $\mathbf{u}_3 = \{u_3(1), u_3(2)\}$. Then our artificial dataset has

$$\begin{aligned}
u_1(1) &= \text{Gaussian Noise (Mean = 0, standard deviation = 0.1)} \\
u_1(2) &= \sin(t) + \text{Gaussian Noise (Mean = 0, standard deviation = 0.1)} \\
u_2(1) &= 1 - (2.6 - t) * (2.6 - t) + \text{Gaussian Noise (Mean = 0, standard deviation = 0.1)} \\
u_2(2) &= \text{Gaussian Noise (Mean = 0, standard deviation = 0.1)} \\
u_3(1) &= -(t - 3) * (t - 2) + \text{Gaussian Noise (Mean = 0 and standard deviation = 0.1)} \\
u_3(2) &= \text{Gaussian Noise (Mean = 0 and standard deviation = 0.1)}
\end{aligned} \tag{15}$$

where t increases from 0 to 3.33 in steps of $\frac{1}{300}$ i.e. we have a 3 stream data set

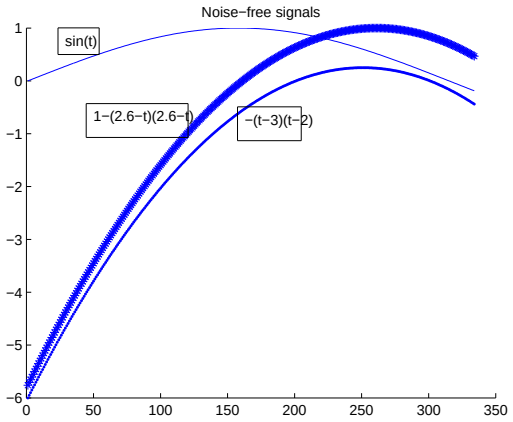


Figure 3: Artificial Noise free signals

of 1000 samples of two dimensional data. The learning rate was 0.0001 and the number of iterations were 10000. Noise-free versions of the underlying signals of this dataset are shown in Figure 3. The multi-set correlation among the three variables is shown in Table 3.

| | $\mathbf{u}_1$ | $\mathbf{u_2}$ | $\mathbf{u_3}$ |
|---|---|---|---|
| $\mathbf{u_1}$ | 1.0000000 | 0.3351417 | 0.4862097 |
| $\mathbf{u_2}$ | 0.3351417 | 1.0000000 | 0.8666971 |
| $\mathbf{u_3}$ | 0.4862097 | 0.8666971 | 1.0000000 |

Table 3: Multi-Set Correlations Between $u_1$, $u_2$ and $u_3$

| | | |
|---|---|---|
| $\mathbf{w_1}$ | 0.01548205 | **0.7630209** |
| $\mathbf{w_2}$ | **1.05575** | 0.0176861 |
| $\mathbf{w_3}$ | **0.7339293** | 0.08696326 |

Table 4: Weights $w_1$, $w_2$ and $w_3$ of $u_1$, $u_2$ and $u_3$

We note from Table 4 that the parts of each data stream which contain true covariance information are those which the weights are identifying: the other dimensions which contain only noise have weight values which are two orders of magnitude less. The correlations in Table 3 illustrate the very strong correlation between the first elements in each of the second and third data streams. The second element of the first data stream contains a signal which is similar to

these two correlated signals but not so close as they are to each other and hence the correlation found between signal 1 and the other 2 is somewhat less.

## 7.2 Temporal MCCA

We consider methods by which we may find non-linear relationships between pairs of data sets. We may use the above MCCA method but use the reservoir activations for a pair of related time series and $W^1_{out}$, $W^2_{out}$ and $W^3_{out}$ in place of $\mathbf{w_1}$, $\mathbf{w_2}$ and $\mathbf{w_3}$. Since we have three data streams we have three separate reservoirs and hence three sets of output weights to update. Results are shown in Table 5.

| | $\mathbf{u}_1$ | $\mathbf{u_2}$ | $\mathbf{u_3}$ |
|---|---|---|---|
| $\mathbf{u_1}$ | 1.0000000 | 0.3543267 | 0.5176265 |
| $\mathbf{u_2}$ | 0.3543267 | 1.0000000 | 0.8899055 |
| $\mathbf{u_3}$ | 0.5176265 | 0.8899055 | 1.0000000 |

Table 5: Multi-Set Non-Linear Correlations Between $u_1$, $u_2$ and $u_3$

We see that the use of the reservoirs has given us larger values in the non-diagonal weights.

## 7.3 High Variance Multi-Set CCA

The idea remains the same as for multi streams of data but aiming to maximise the changes within each data stream separately.

The criteria for finding High Variance Multi-set Canonical Correlations of three variables will be given as:

$$\begin{bmatrix} 0 & \Sigma_{12} & \Sigma_{13} \\ \Sigma_{21} & 0 & \Sigma_{23} \\ \Sigma_{31} & \Sigma_{32} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{bmatrix} = \rho \left( \begin{bmatrix} \Sigma_{11} - \Sigma_{1\dot{1}} & 0 & 0 \\ 0 & \Sigma_{22} - \Sigma_{2\dot{2}} & 0 \\ 0 & 0 & \Sigma_{33} - \Sigma_{3\dot{3}} \end{bmatrix} \right) \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{bmatrix}. \tag{16}$$

The method of finding High Variance canonical correlation directions $w_1$, $w_2$ and $w_3$ is then

$$\begin{aligned} \frac{d\mathbf{w}_1}{dt} &= \Sigma_{12}\mathbf{w}_2 + \Sigma_{13}\mathbf{w}_3 - f(\mathbf{w})(\Sigma_{11} - \Sigma_{1\dot{1}})\mathbf{w}_1, \\ \frac{d\mathbf{w}_2}{dt} &= \Sigma_{21}\mathbf{w}_1 + \Sigma_{23}\mathbf{w}_3 - f(\mathbf{w})(\Sigma_{22} - \Sigma_{2\dot{2}})\mathbf{w}_2, \\ \frac{d\mathbf{w}_3}{dt} &= \Sigma_{31}\mathbf{w}_2 + \Sigma_{32}\mathbf{w}_2 - f(\mathbf{w})(\Sigma_{33} - \Sigma_{3\dot{3}})\mathbf{w}_3. \end{aligned}$$

Using the fact that $\Sigma_{ij} = E(\mathbf{x}_i \mathbf{x}_j^T), i, j = 1, 2$ and that $y_1 = \mathbf{w}_1.\mathbf{x}_1$, we may propose the instantaneous rules

$$\begin{aligned}
\Delta \mathbf{w}_1 &= \eta(\mathbf{x}_1 y_2 + \mathbf{x}_1 y_3 - f(\mathbf{w}) \left( \mathbf{x}_1 y_1 - \left( (\frac{d\mathbf{x_1}}{dt})(\frac{d\mathbf{x_1}}{dt}^T) \right) \right), \\
\Delta \mathbf{w}_2 &= \eta(\mathbf{x}_2 y_1 + \mathbf{x}_2 y_3 - f(\mathbf{w}) \left( \mathbf{x}_2 y_2 - \left( (\frac{d\mathbf{x_2}}{dt})(\frac{d\mathbf{x_2}}{dt}^T) \right) \right), \\
\Delta \mathbf{w}_3 &= \eta(\mathbf{x}_3 y_1 + \mathbf{x}_3 y_2 - f(\mathbf{w}) \left( \mathbf{x}_3 y_3 - \left( (\frac{d\mathbf{x_3}}{dt})(\frac{d\mathbf{x_3}}{dt}^T) \right) \right).
\end{aligned}$$

We have used the same artificial datasets for the High Variance approach. Table 6 shows that the High Variance method has produced slightly higher correlations as compared to the generalized approach. It can be seen more clearly from the values of the weight vectors shown in Table 7 that the method is ignoring the noise parts of each data stream and concentrating on the signal parts.

| | $\mathbf{u_1}$ | $\mathbf{u_2}$ | $\mathbf{u_3}$ |
|---|---|---|---|
| $\mathbf{u_1}$ | 1.0000000 | 0.3357379 | 0.5029659 |
| $\mathbf{u_2}$ | 0.3357379 | 1.0000000 | 0.8704137 |
| $\mathbf{u_3}$ | 0.5029659 | 0.8704137 | 1.0000000 |

Table 6: Multi-Set Correlations Between $u_1$, $u_2$ and $u_3$

Again we emphasise that these results are without the use of a reservoir.

| | | |
|---|---|---|
| $\mathbf{w_1}$ | -0.04771852 | **0.7397458** |
| $\mathbf{w_2}$ | **1.073291** | 0.006790616 |
| $\mathbf{w_3}$ | **0.7530229** | -0.08743803 |

Table 7: Weights $w_1$, $w_2$ and $w_3$ of $u_1$, $u_2$ and $u_3$

## 7.4 Temporal High Variance MCCA

We may follow the above criteria by using reservoir activations to create a new method by which we can compute High Variance canonical correlations among multi-set data. It can be seen from the Table 8 that the correlations are a bit higher with reservoirs as compared to the generalized technique but not as high as with temporal CCA using the reservoirs.

| | $\mathbf{u_1}$ | $\mathbf{u_2}$ | $\mathbf{u_3}$ |
|---|---|---|---|
| $\mathbf{u_1}$ | 1.0000000 | 0.3443723 | 0.5158995 |
| $\mathbf{u_2}$ | 0.3443723 | 1.0000000 | 0.9183531 |
| $\mathbf{u_3}$ | 0.5158995 | 0.9183531 | 1.0000000 |

Table 8: Multi-Set Non-Linear Correlations Between $u_1$, $u_2$ and $u_3$

# 8 Comparative Analysis on Real Data

We have used the MNIST data set [15] consisting of 60000 training patterns containing 0-9 handwritten digits and 10000 test patterns of the same digits (0-9). Each digit consists of 784 pixels which are of $28 \times 28$ pixels enclosed in a bounding box. Every digit of the same type is slightly different from every other in terms of position, size and shape. In order to compute multi-set canonical correlation using our method we have chosen one digit randomly from every class (0-9) and find the generalized multi-set canonical correlations between digits belonging to different classes. The learning rate of our algorithm is 0.0001 and the total number of iterations for learning all set of digits are 100000. We are displaying a combined comparative results of all the methods related to MCCA (Multi-set Canonical Correlation Analysis) that we have proposed. The combined results are shown in Table 9

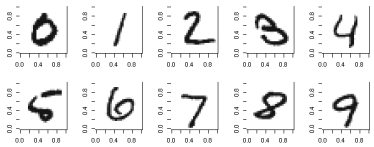

Figure 4: Ten Digit Used in the Experiment

### 8.0.1 Discussion

We can see in Table 9, that some figures have a high correlation with each other e.g. 6 and 8 while others have a lower correlation e.g. 3 and 1. The reasons for these should be obvious.

In Table 10, we have used 2-tailed t-values to compare the performance of the various methods, comparing them in pairs: when we write $a > b, 99\%$, we mean that the improvement in performance of **a** over **b** is significant at the 99% confidence level; similarly $\mathbf{a} < \mathbf{b}, 99\%$ means that **b** improves **a** with **a** significance value greater than the 99% confidence level.

According to the performance measurement chart shown in Table 10, we can see that the addition of reservoirs to the method always improves the classification accuracy with respect to the identification of individual figures. Note

that HVMCCA is better than GMCCA (with 99% confidence) but the addition of reservoirs reverses this. Our conjecture is that the reservoirs themselves are adding variance though this is a feature which requires further analysis. The rationale behind deriving all these new methods is to extract selected features from the data which can further maximize the correlation between two and more streams in comparison with the previously derived techniques in a completely unsupervised manner. All the techniques performed consistently well for different kinds of data. Temporal CCA is good on numeric time series data. Similarly High Variance CCA (HVCCA) works well on image data. Temporal High Variance CCA proves useful to extract time series information from image data. In other words, each technique is specifically designed to work for a particular kind of data stream.

# 9 Conclusion

We have developed an extension of a method to find the canonical correlation analysis of a data set. In particular we have

1. Used reservoir activations to capture information on temporal or image data and subsequently used the online weight adaptation algorithm to create a novel method known as Temporal CCA.
2. We used a technique suggested by the method of Slow Feature Analysis [26] to ensure that our correlations do not come from static signals.
3. We have developed an online Multi-set CCA method which is computationally inexpensive.
4. We have combined the above techniques and shown results on real and artificial data sets.

We have concluded that the generalised online method for finding canonical correlations is more appropriate for numeric data (our artificial data as well as the student exam data) whereas the temporal high variance method is more appropriate for image data sets (MNIST digit data) because in images most of the time the constant data needs to be ignored.

Future work will concentrate on finding relationships between the derivative information in 2 or more data sets simultaneously. We will also use a weighted approach of higher order derivatives on image and temporal time series. We will also investigate how different structures in reservoirs can help to extract different information from multiple data streams.

# 10 Acknowledgements

The first author is grateful to Professor Colin Fyfe, formerly with the University of The West of Scotland, for his insightful suggestions which helped improve the writing of this paper.

# References


[1] S. Akaho. A kernal method for canonical correlation analysis. In *Proceedings of the International Meeting of the Psychometric Society.* (IMPS2001), Springer (arXiv preprint cs/0609071.), 2006.

[2] Meinecke F. C. Gretton A. Rauch A. Rainer G. Logothetis N. K. Biessmann, F. and K. R. Mueller. Temporal kernal cca and its application in multimodal neuronal data analysis. *Machine Learning*, 79(1-2):5–27, 2010.

[3] Gastaldo P. Bisio F. Cambria, E. and R. Zunino. An elm-based model for affective analogical reasoning. *Neurocomputing*, 149:1–18, doi:10.1016/j.neucom.2014.01.064, 2014.

[4] Broga M. Lundberg P. Knutsson H. Friman, O. Exploratory fmri analysis by autocorrelation maximization. *Neuroimage*, 16(2):454–464, 2002.

[5] Zunino R. Cambria E. Gastaldo, P. and S. Decherchi. Combining elm with random projections. *IEEE Intelligent Systems*, 28(6):1, 2013.

[6] Z. Gou and C. Fyfe. A family of networks which perform canonical correlation analysis. *International Journal of Knowledge-based Intelligent Engineering Systems*, 5(2):76–82, April 2001.

[7] Z. K. Gou and C. Fyfe. A canonical correlation neural network for multicollinearity and functional data. *Neural Networks*, 17(2):285–293, 2003.

[8] C. Gros. Cognitive computation with autonomously active neural networks: An emerging field. *Cognitive Computation*, 1(1):77, 2009.

[9] Szedmak S. Hardoon, D.R. and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[10] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[11] G.B. Huang. An insight into extreme learning machines: Random neurons, random features and kernels. *Cognitive Computation*, 6(3):376, 2014.

[12] J.R. Kettenring. Canonical analysis of several sets of variables. *Biometrika*, 58(3):433, 1971.

[13] P. L. Lai. *Neural Implementations of Canonical Correlation Analysis.* PhD thesis, University of Paisley, 2000.

[14] P.L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(5):365–377, 2001.

[15] Y. LeCun and C. Cortes. The mnist database of handwritten digits. *The Dataset is available at http://yann.lecun.com/exdb/mnist/.*

[16] D. Lemire. Faster retrieval with a two-pass dynamic-time-warping lower bound. *Pattern Recognition*, 42(9):2169, 2009.

[17] M. Lukosevicius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Computing Science Review*, 3(3):127–149, 2009.

[18] Z.K. Malik, A. Hussain, and J. Wu. Novel biologically inspired approaches to extracting online information from temporal data. *Cognitive Computation*, 6(3):1–13, 2014.

[19] Kent J.T. Mardia, K.V. and Bibby J.M. Multivariate analysis. In *Academic Press*, 1979.

[20] A.A. Nielsen. Multi-set canonical correlations analysis and multispectral, truely multitemporal remote sensing data. *IEEE Transactions on Image Processing*, 11(3):293, 2002.

[21] Gelbukh A. Cambria E. Hussain A. Poria, S. and G.B. Huang. Emosenticspace: A novel framework for affective common-sense reasoning. *Knowledge-Based Systems*, 69:108–123, doi:10.1016/j.knosys.2014.06.011, 2014.

[22] B. Schrauwen, D. Verstraeten, and J. Van Campenhout. An overview of reservoir computing: theory, application and implementation. In *Proceedings of the 15th European Symposium on Artificial Neural Networks*, pages 471–482, 2007.

[23] J. J. Steil. Online reservoir adaptation by intrinsic plasticity for backpropagation-decorrelation and echo state learning. *Neural networks*, 20(3):353–364, 2007.

[24] J. Via. A learning algorithm for adaptive canonical correlation analysis of several data sets. *Neural Networks*, 20(1):139–152, 2007.

[25] X. Wang, M. Crowe, and C. Fyfe. Dual stream data exploration. *International Journal of Data Mining, Modelling and Management*, 4(2):188–202, 2012.

[26] L. Wiskott and T. J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002.

[27] Eyben F. Graves A. Schuller B. Wollmer, M. and G. Rigoll. Bidirectional lstm networks for context-sensitive keyword detection in a cognitive virtual agent framework. *Cognitive Computation*, 2(3):180, 2010.

[28] Y. Yang, Q.M. Wu, Y. Wang, M. Zeeshan, K., X. Lin, and X. Yuan. Data partition learning with multiple extreme learning machines. *Cybernetics, IEEE Transactions*, PP(99):18–20, DOI:10.1109/TCYB.2014.2352594, 2014.

[29] F. Yger, M. Berar, G. Gasso, and A. Rakotomamonjy. Adaptive canonical correlation analysis based on matrix manifolds. In *Proc. of the Int'l Conf. on Machine Learning*. New york, NY, ACM, 2012.

[30] Guenther Zeck, Matthias Bethge, and Jakob H. Macke. Receptive fields without spike-triggering. In J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 969–976. Curran Associates, Inc., 2008.

[31] Q. Zhang and Y. W. Leung. A class of learning algorithms for principal component analysis and minor component analysis. *IEEE Transactions on Neural Networks*, 11(2):200–204, 2000.

[32] Z.K., Malik, A., Hussain, and J., Wu, "An Online Generalized Eigenvalue Version of Laplacian Eigenmap for Visual Big Data," *Neurocomputing*, 2015, (in press).

| | Method | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | GMCCA | 1.000 | 0.571 | 0.817 | 0.749 | 0.693 | 0.765 | 0.882 | 0.851 | 0.828 | 0.826 |
| | GMCCA(R) | 1.000 | 0.903 | 0.979 | 0.957 | 0.942 | 0.966 | 0.979 | 0.982 | 0.965 | 0.986 |
| | HVMCCA | 1.000 | 0.641 | 0.876 | 0.842 | 0.825 | 0.856 | 0.855 | 0.927 | 0.875 | 0.925 |
| | HVMCCA(R) | 1.000 | 0.759 | 0.934 | 0.955 | 0.974 | 0.973 | 0.954 | 0.961 | 0.961 | 0.989 |
| 1 | GMCCA | 0.571 | 1.000 | 0.705 | 0.566 | 0.135 | 0.735 | 0.431 | 0.792 | 0.714 | 0.599 |
| | GMCCA(R) | 0.903 | 1.000 | 0.866 | 0.816 | 0.822 | 0.841 | 0.896 | 0.910 | 0.854 | 0.906 |
| | HVMCCA | 0.641 | 1.000 | 0.721 | 0.719 | 0.292 | 0.795 | 0.518 | 0.743 | 0.795 | 0.564 |
| | HVMCCA(R) | 0.759 | 1.000 | 0.697 | 0.723 | 0.679 | 0.742 | 0.697 | 0.746 | 0.747 | 0.749 |
| 2 | GMCCA | 0.817 | 0.705 | 1.000 | 0.779 | 0.314 | 0.632 | 0.785 | 0.704 | 0.876 | 0.592 |
| | GMCCA(R) | 0.979 | 0.866 | 1.000 | 0.981 | 0.935 | 0.987 | 0.971 | 0.973 | 0.987 | 0.982 |
| | HVMCCA | 0.876 | 0.721 | 1.000 | 0.839 | 0.553 | 0.767 | 0.793 | 0.798 | 0.886 | 0.707 |
| | HVMCCA(R) | 0.934 | 0.697 | 1.000 | 0.903 | 0.928 | 0.888 | 0.917 | 0.867 | 0.926 | 0.939 |
| 3 | GMCCA | 0.749 | 0.566 | 0.779 | 1.000 | 0.387 | 0.747 | 0.736 | 0.724 | 0.934 | 0.601 |
| | GMCCA(R) | 0.957 | 0.816 | 0.981 | 1.000 | 0.919 | 0.978 | 0.957 | 0.956 | 0.989 | 0.953 |
| | HVMCCA | 0.955 | 0.723 | 0.903 | 1.000 | 0.944 | 0.987 | 0.935 | 0.961 | 0.981 | 0.954 |
| | HVMCCA(R) | 0.955 | 0.723 | 0.903 | 1.000 | 0.944 | 0.987 | 0.935 | 0.961 | 0.981 | 0.954 |
| 4 | GMCCA | 0.693 | 0.135 | 0.314 | 0.387 | 1.000 | 0.594 | 0.505 | 0.636 | 0.367 | 0.826 |
| | GMCCA(R) | 0.942 | 0.822 | 0.935 | 0.919 | 1.000 | 0.943 | 0.957 | 0.944 | 0.932 | 0.941 |
| | HVMCCA | 0.825 | 0.292 | 0.553 | 0.595 | 1.000 | 0.744 | 0.674 | 0.748 | 0.558 | 0.913 |
| | HVMCCA(R) | 0.974 | 0.679 | 0.928 | 0.944 | 1.000 | 0.958 | 0.975 | 0.937 | 0.947 | 0.972 |
| 5 | GMCCA | 0.765 | 0.735 | 0.632 | 0.747 | 0.594 | 1.000 | 0.631 | 0.844 | 0.733 | 0.878 |
| | GMCCA(R) | 0.967 | 0.841 | 0.987 | 0.978 | 0.943 | 1.000 | 0.962 | 0.963 | 0.989 | 0.967 |
| | HVMCCA | 0.855 | 0.795 | 0.767 | 0.872 | 0.745 | 1.000 | 0.714 | 0.903 | 0.846 | 0.901 |
| | HVMCCA(R) | 0.973 | 0.742 | 0.888 | 0.987 | 0.958 | 1.000 | 0.951 | 0.962 | 0.966 | 0.969 |
| 6 | GMCCA | 0.882 | 0.431 | 0.785 | 0.736 | 0.505 | 0.631 | 1.000 | 0.611 | 0.802 | 0.576 |
| | GMCCA(R) | 0.979 | 0.896 | 0.971 | 0.957 | 0.957 | 0.962 | 1.000 | 0.985 | 0.967 | 0.976 |
| | HVMCCA | 0.855 | 0.518 | 0.793 | 0.722 | 0.674 | 0.714 | 1.000 | 0.737 | 0.799 | 0.723 |
| | HVMMCA(R) | 0.975 | 0.697 | 0.917 | 0.935 | 0.975 | 0.951 | 1.000 | 0.951 | 0.958 | 0.959 |
| 7 | GMCCA | 0.851 | 0.792 | 0.704 | 0.724 | 0.636 | 0.844 | 0.611 | 1.000 | 0.815 | 0.874 |
| | GMCCA(R) | 0.982 | 0.910 | 0.973 | 0.956 | 0.944 | 0.963 | 0.985 | 1.000 | 0.973 | 0.987 |
| | HVMCCA | 0.927 | 0.743 | 0.798 | 0.914 | 0.748 | 0.903 | 0.737 | 1.000 | 0.914 | 0.917 |
| | HVMCCA(R) | 0.954 | 0.746 | 0.867 | 0.961 | 0.937 | 0.962 | 0.951 | 1.000 | 0.965 | 0.943 |
| 8 | GMCCA | 0.828 | 0.714 | 0.876 | 0.934 | 0.367 | 0.733 | 0.802 | 0.815 | 1.000 | 0.628 |
| | GMCCA(R) | 0.965 | 0.854 | 0.987 | 0.989 | 0.932 | 0.989 | 0.967 | 0.973 | 1.000 | 0.969 |
| | HVMCCA | 0.875 | 0.795 | 0.886 | 0.955 | 0.558 | 0.846 | 0.799 | 0.914 | 1.000 | 0.764 |
| | HVMCCA(R) | 0.961 | 0.747 | 0.926 | 0.981 | 0.948 | 0.966 | 0.958 | 0.965 | 1.000 | 0.945 |
| 9 | GMCCA | 0.826 | 0.599 | 0.592 | 0.601 | 0.826 | 0.878 | 0.576 | 0.874 | 0.628 | 1.000 |
| | GMCCA(R) | 0.986 | 0.906 | 0.983 | 0.953 | 0.941 | 0.967 | 0.976 | 0.987 | 0.969 | 1.000 |
| | HVMCCA | 0.925 | 0.564 | 0.707 | 0.787 | 0.913 | 0.901 | 0.723 | 0.917 | 0.764 | 1.000 |
| | HVMCCA(R) | 0.989 | 0.749 | 0.939 | 0.954 | 0.973 | 0.969 | 0.959 | 0.943 | 0.945 | 1.000 |

Table 9: Multi-Set Correlations Between digit 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. GMCCA(Generalized Multi-Set Canonical Correlation Analysis), GMCCA(R)(Generalized Multi-Set Canonical Correlation Analysis with Reservoir), HVMCCA(High Variance Multi-Set Canonical Correlation Analysis), HVMCCA(R)(High Variance Multi-Set Canonical Correlation Analysis with Reservoir).

| | |
|---|---|
| GMCCA-GMCCA(R) | $< 99$ % |
| HVMCCA-GMCCA | $> 99$ % |
| HVMCCA(R)-GMCCA | $> 99$ % |
| GMCCA(R)-HVMCCA | $> 99$ % |
| GMCCA(R)-HVMCCA(R) | $> 99$ % |
| HVMCCA(R)-HVMCCA | $> 99$ % |

Table 10: Performance Measurement. Paired-wise Comparison with a confidence interval of 99 %