Task-Driven Deep Canonical Correlation Analysis

Anonymous Authors[1]

Abstract

Canonical Correlation Analysis (CCA) is widely used for multimodal data analysis and, more recently, for discriminative tasks such as cross-modal classification; however, it makes no use of class label information. Recent CCA methods have begun to address this weakness but are limited in that they do not simultaneously optimize the discriminative capability of the CCA projection and the CCA projection itself, or they are linear only. We extend recent deep variants of CCA by adding a task-driven component, and we reformulate the CCA component to better suit the deep learning framework for end-to-end training. Together, these components learn a non-linear CCA projection to a shared latent space that is both highly correlated and discriminative. Our method is validated on real data, showing a significant improvement over previous state-of-the-art.

1. Introduction

Canonical Correlation Analysis (CCA) is a popular data analysis technique that projects two modalities into a space in which they are maximally correlated (Hotelling, 1936; Bie et al., 2005). It was initially used for unsupervised data analysis to gain insights into the shared components of two modalities (Andrew et al., 2013; Wang et al., 2015a; 2016). However, it has also found utility in computing a shared latent space for cross-modal classification (Kan et al., 2015; Wang et al., 2015a; Chandar et al., 2016; Chang et al., 2018). On some data sets, CCA-based methods can also find a more discriminative feature set for multimodal classification (Dorfer et al., 2016b). While some of the correlated features extracted by CCA are useful for discriminative tasks, many represent properties that are of no use for classification tasks and obscure correlated information that is beneficial. This problem is magnified with recent non-linear extensions of CCA using deep learning that make significant strides in improving correlation (Andrew et al., 2013; Wang et al., 2015a; 2016; Chang et al., 2018) but often at the expense of discriminative capability, as will be demonstrated empirically. Therefore, we present a new deep learning technique to project the data from two modalities to a shared space that is also discriminative.

Most prior work that boosts the discriminative capability of CCA is linear only (Lee et al., 2015; Singanamalli et al., 2014; Duan et al., 2016). More recent work using deep learning still remains limited in that itoptimizes discriminative capability for an intermediate representation rather than the final CCA projection (Dorfer et al., 2016b) or focuses on a task objective without optimizing the CCA component (Dorfer et al., 2018). We jointly optimize CCA and a discriminative objective by computing the CCA projection within a network layer and applying a task-driven operation such as classification. Experimental results will show that our method significantly improves upon previous deep and discriminative approaches (Dorfer et al., 2016b; 2018) due to its focus on both the shared latent space and task-driven objective. The task-driven component is particularly important in achieving this on small training set sizes.

While alternative multimodal approaches to CCA exist, they are typically focused on a reconstruction objective. That is, they transform the input data into a shared space such that the input could be reconstructed - either individually or reconstructing one modality from the other. Variations of coupled dictionary learning (Shekhar et al., 2014; Xu et al., 2015; Cha et al., 2015; Bahrampour et al., 2015) and auto-encoders (Wang et al., 2015a; Bhatt et al., 2017) have been used, along with further extensions to extract components shared by both modalities and individual components (Ray et al., 2014; Lock et al., 2013; Zhou et al., 2015; Yang & Michailidis, 2015). CCA-based objectives, such as the model used in this work, instead learn a transformation to a shared space, without the need for reconstructing the input. This task may be easier and sufficient in producing a representation for cross-modal classification (Wang et al., 2015a). We will show that the CCA objective can equivalently be expressed as an ℓ_2 distance minimization in the shared space, plus an orthogonality constraint. In a deep network, an orthogonality constraint provides a means of regularization (Huang et al., 2018) and we present three

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

techniques to accomplish this. While our method is derived from CCA, through manipulating the orthogonality constraints it becomes a means to compute a shared latent space that is also discriminative.

Our method combines the CCA and task-driven objectives in a novel way that enables end-to-end training and supports mini-batch training. We demonstrate the effectiveness of our model on a multimodal variation of MNIST adapted for study across different training set sizes, showing that task-driven deep CCA significantly improves cross-modal classification accuracy and is more robust to a small training set size than alternative methods. We provide further validation on a cancer imaging and genomic data set with only 1000 samples to show how our technique can regularize a model when two modalities are available for training but only one at test time. Finally, we demonstrate task-driven deep CCA in a semi-supervised setting for speech recognition. These experiments on real data show the effectiveness of our method in learning a shared space that is much more discriminative than previous state-of-the-art.

Contributions. 1) A task-driven deep CCA method that integrates the CCA operation into the network for end-to-end mini-batch training. 2) Three variations of this method that relax the orthogonality constraints of CCA by applying regularization in different ways. 3) Validation of this method showing a significant improvement in cross-modal classification accuracy over existing state-of-the-art methods and robustness to small training set sizes. 4) Validation of the importance of small-batch training for deep CCA methods. 5) Experiments on two real data sets demonstrating alternative applications of multimodal regularization during training and semi-supervised learning.

2. Background

This section introduces some fundamental CCA methods as background material before we present our task-driven method in the next section. CCA and its non-linear variants are unsupervised methods that find the shared signal between a pair of modalities by maximizing the sum correlation between projections of the two modalities. Let $X_1 \epsilon \mathbb{R}^{d_1 \times n}$ and $X_2 \epsilon \mathbb{R}^{d_2 \times n}$ be mean centered input data matrices from two different modalities with n samples and d_1 or d_2 features.

CCA. CCA maximizes the correlation between linear projections $a_1 = w_1^T X_1$ and $a_2 = w_2^T X_2$, where w_1 and w_2 are projection vectors (Hotelling, 1936). The first canonical direction is found by maximizing the correlation

$$\underset{w_1, w_2}{\text{argmax}} \, \text{corr}(w_1^T X_1, w_2^T X_2) \; .$$

Subsequent projections are found by maximizing the same correlation but in orthogonal directions.

Combining the projection vectors into matrices $W_1 = [w_1^{(1)}, ..., w_1^{(k)}]$ and $W_2 = [w_2^{(1)}, ..., w_2^{(k)}]$ ($k \leq \min(d_1, d_2)$), CCA can be reformulated as a trace with orthonormality constraints on the projections:

$$\underset{W_1, W_2}{\text{argmax}} \, \text{tr}(W_1^T \Sigma_{12} W_2) \quad \text{s.t. } W_1^T \Sigma_1 W_1 = W_2^T \Sigma_2 W_2 = I \tag{1}$$

for covariance matrices $\Sigma_1 = \frac{1}{n-1} X_1 X_1^T$ and $\Sigma_2 = \frac{1}{n-1} X_2 X_2^T$ and cross-covariance matrix $\Sigma_{12} = \frac{1}{n-1} X_1 X_2^T$. Let $T = \Sigma_1^{-1/2} \Sigma_{12} \Sigma_2^{-1/2}$ and its singular value decomposition (SVD) be $T = U_1 \text{diag}(\sigma) U_2^T$ with singular values $\sigma = [\sigma_1, ..., \sigma_{\min(d_1, d_2)}]$ in descending order. W_1 and W_2 are computed from the top k left and right singular vectors of T:

$$W_1 = \Sigma_1^{-1/2} U_1^{(1:k)} \quad W_2 = \Sigma_2^{-1/2} U_2^{(1:k)} \tag{2}$$

where $U^{(1:k)}$ is the k first columns of matrix U. The sum correlation in the projection space is equivalent to the sum of the top k singular values:

$$\sum_{i=1}^{k} \text{corr}((w_1^{(i)})^T X_1, (w_2^{(i)})^T X_2) = \sum_{i=1}^{k} \sigma_i^2 \; . \tag{3}$$

A regularized variation of CCA (RCCA) ensures that the covariance matrices are positive definite by computing the covariance matrices as $\hat{\Sigma}_1 = \frac{1}{n-1} X_1 X_1^T + rI$ and $\hat{\Sigma}_2 = \frac{1}{n-1} X_2 X_2^T + rI$ for regularization parameter r and identity matrix I (Bilenko & Gallant, 2016).

DCCA. Deep CCA adds non-linear projections to the CCA formulation by replacing the input data with deep networks and training end-to-end. The input data X_1 and X_2 are replaced with feed-forward networks f_1 and f_2, using parameters θ_1 and θ_2 and producing activations $A_1 = f_1(X_1; \theta_1)$ and $A_2 = f_2(X_2; \theta_2)$, respectively (assumed to be mean centered) (Andrew et al., 2013). Matrices $A_1, A_2 \epsilon \mathbb{R}^{d_o \times n}$ are the output activations from the networks f_1 and f_2 with d_o features on the output layer. Figure 1a shows the network structure.

DCCA optimizes the same objective as CCA (Equation 1) but using activations A_1 and A_2 as inputs. Regularized covariance matrices are computed as $\Sigma_1 = \frac{1}{n-1} A_1 A_1^T + rI$, $\Sigma_2 = \frac{1}{n-1} A_2 A_2^T + rI$, and $\Sigma_{12} = \frac{1}{n-1} A_1 A_2^T$ where the regularization parameter r is used in the same way as RCCA. The solution for W_1 and W_2 can be computed using SVD just as with linear CCA (Equation 2). When $k = d_o$ (the number of CCA components is equal to the number of features in A_1 and A_2), optimizing the sum correlation in the projection space (Equation 3) is equivalent to optimizing the matrix trace norm

$$\mathcal{L}_{\text{TNO}}(A_1, A_2) = ||T||_{\text{tr}} = \text{tr}(T^T T)^{1/2}$$

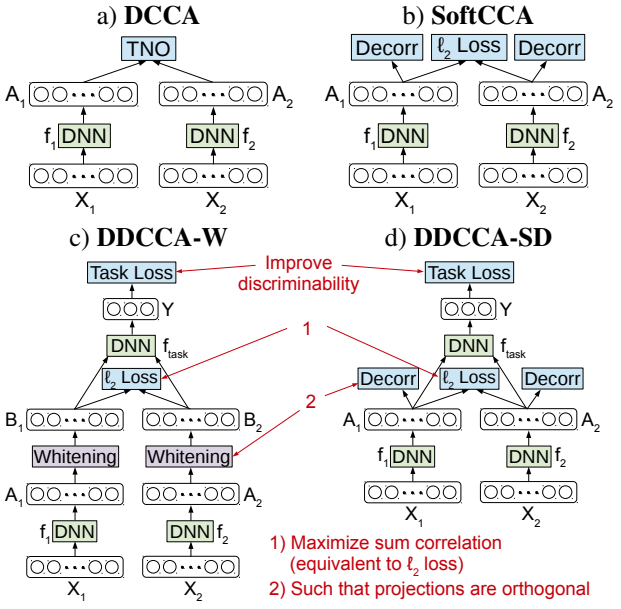

Figure 1. Deep CCA architectures used in this work. a) DCCA maximizes the sum correlation in the projection space but optimizes an equivalent loss, the trace norm objective (TNO) (Andrew et al., 2013). b) SoftCCA relaxes the orthogonality constraints by regularizing with Decorr and optimizes the ℓ_2 distance in the projection space (equivalent to the sum correlation when activations are normalized to a variance of one) (Chang et al., 2018). Our DDCCA methods add a task-driven loss and apply the orthogonality constraints of CCA by regularizing in one of two ways: c) DDCCA-W uses whitening and d) DDCCA-SD using soft decorrelation (Decorr). The third method that we propose, DDCCA-ND, simply removes the Decorr components of DDCCA-SD.

where $T = \Sigma_1^{-1/2}\Sigma_{12}\Sigma_2^{-1/2}$ as for CCA (Andrew et al., 2013). DCCA optimizes this trace norm objective (TNO) directly, without a need to compute the CCA projection within the network. The TNO is optimized first, followed by a linear CCA operation before downstream tasks like classification are performed.

SoftCCA. While DCCA enforces orthogonality constraints on projections $W_1^T A_1$ and $W_2^T A_2$, SoftCCA relaxes these constraints by applying them as regularization (Chang et al., 2018). First, the objective is modified and final projection matrices W_1 and W_2 are incorporated into f_1 and f_2. The trace objective for DCCA (Equation 1) can be rewritten as minimizing the ℓ_2 distance between the projections

$$\mathcal{L}_{\ell_2 \text{ dist}}(A_1, A_2) = ||A_1 - A_2||_F^2 \tag{4}$$

as long as each feature in A_1 and A_2 is normalized to a variance of one (Li et al., 2003). This new objective is now separable across mini-batches. We use this ℓ_2 distance objective in our formulation.

SoftCCA relaxes the orthogonality constraints of CCA by using regularization to penalize the off-diagonal elements of the covariance matrix. A running average of the covariance matrix is computed over mini-batches as $\hat{\Sigma}$, and the Decorr loss for A_1 and A_2 is

$$\mathcal{L}_{\text{Decorr}}(A) = \sum_{i \neq i}^{d_o} |\hat{\Sigma}_{i,j}| \ . \tag{5}$$

Together, the ℓ_2 distance and Decorr loss make the SoftCCA formulation:

$$\mathcal{L}_{\ell_2 \text{ dist}}(A_1, A_2) + \lambda \left(\mathcal{L}_{\text{Decorr}}(A_1) + \mathcal{L}_{\text{Decorr}}(A_2)\right) \ .$$

Supervised CCA Methods. CCA, DCCA, and SoftCCA are all unsupervised methods to learn a projection to a space in which the data is maximally correlated. Although these methods have shown utility for discriminative tasks, a CCA decomposition is not necessarily optimal for classification purposes because it may extract features that are correlated but not discriminative. Our experiments will make it clear that maximizing the correlation objective too much can degrade discriminative tasks.

CCA has previously been extended to the supervised case by maximizing the total correlation between each modality and the training labels in addition to each pair of modalities (Lee et al., 2015; Singanamalli et al., 2014) and by maximizing the separation of classes (Kan et al., 2015; Dorfer et al., 2016b). Although these methods incorporate the class labels, they do not directly optimize for the classification task with end-to-end training. Dorfer et. al's CCA Layer (CCAL) is the closest to our method. It optimizes a task-driven loss operating on a CCA projection; however, the CCA loss itself is only optimized during pre-training, not during end-to-end training (Dorfer et al., 2018). Other supervised CCA methods are linear only (Singanamalli et al., 2014; Lee et al., 2015; Kan et al., 2015; Duan et al., 2016). Our focus is on deep methods; in particular, our method simultaneously optimizes CCA and a task-driven objective in a deep network to capture non-linear representations. Instead of simply computing the CCA projection within the network as the CCAL method does, we optimize for the shared space with the CCA part of our objective.

3. Task-driven Deep CCA

In order to compute a shared latent space that is also discriminative, we start with the deep CCA formulation and add a task-driven term to the optimization objective. Our methods and related DNN ones from the literature are summarized in Table 1 and schematic diagrams are provided in Figure 1.

While DCCA provides a means to optimize the sum correlation through an equivalent loss function, the TNO, the CCA projection itself is computed only *after* optimization is complete. This means that the projections cannot be used to optimize another task simultaneously with end-to-end training. The main challenge in developing a task-driven

Method	Objective	
CCA	$\text{tr}(W_1^T \Sigma_{12} W_2)$	s.t. $W_1^T \Sigma_1 W_1 = W_2^T \Sigma_2 W^2 = I$
DCCA	$-\|\|\Sigma_1^{-1/2} \Sigma_{12} \Sigma_2^{-1/2}\|\|_{\text{tr}}$ TNO where $\|\|T\|\|_{\text{tr}} = \text{tr}(T^T T)^{1/2}$	equivalent to CCA objective $\text{CCA}(W_1^T A_1, W_2^T A_2)$ computed after optimization complete
SoftCCA	$\mathcal{L}_{\ell_2 \text{ dist}}(A_1, A_2) + \lambda \left(\mathcal{L}_{\text{Decorr}}(A_1) + \mathcal{L}_{\text{Decorr}}(A_2) \right)$	
CCAL-$\mathcal{L}_{\text{rank}}$	$\mathcal{L}_{\text{rank}}(B_1, B_2)$	where $B_1, B_2 = \text{CCA}(A_1, A_2)$, $\mathcal{L}_{\text{rank}}$ is pairwise rank loss
DDCCA-W	$\text{Task}(B_1, B_2, Y) + \lambda \ \mathcal{L}_{\ell_2 \text{ dist}}(B_1, B_2)$	where $B_1 = U_1 A_1, B_2 = U_2 A_2$ s.t. $B_1^T B_1 = B_2^T B_2 = I$
DDCCA-SD	$\text{Task}(A_1, A_2, Y) + \lambda_1 \mathcal{L}_{\ell_2 \text{ dist}}(A_1, A_2) + \lambda_2 \left(\mathcal{L}_{\text{Decorr}}(A_1) + \mathcal{L}_{\text{Decorr}}(A_2) \right)$	Whitening
DDCCA-ND	$\text{Task}(A_1, A_2, Y) + \lambda \ \mathcal{L}_{\ell_2 \text{ dist}}(A_1, A_2)$	

Loss functions

ℓ_2 dist:	$\mathcal{L}_{\ell_2 \text{ dist}}(A_1, A_2) = \|\|A_1 - A_2\|\|_F^2$
Decorr:	$\mathcal{L}_{\text{Decorr}}(A) = \sum_{i \neq j} \lvert\hat{\Sigma}_{i,j}\rvert$ where $\hat{\Sigma}$ is running mean across mini-batches of $\Sigma = A^T A$
Task:	$\text{Task}(A_1, A_2, Y) = \mathcal{L}_{\text{task}}(f_{\text{task}}(A_1), Y) + \mathcal{L}_{\text{task}}(f_{\text{task}}(A_2), Y)$ where $\mathcal{L}_{task}$ can be cross-entropy or any other task-driven loss

Table 1. A comparison of our proposed task-driven deep CCA methods with other related ones from the literature, including DCCA (Andrew et al., 2013), SoftCCA (Chang et al., 2018), and CCAL-$\mathcal{L}_{\text{rank}}$ (Dorfer et al., 2018). A_1 and A_2 are assumed to mean centered outputs from two feed-forward networks. $\Sigma = A^T A$ is computed from a single (large) batch (used in DCCA), $\hat{\Sigma}$ is computed as a running mean over batches (used in all other methods).

form of deep CCA that discriminates based on the CCA projection is in computing this projection within the network - a necessary step to enable end-to-end training. We tackle this by focusing on the two components of deep CCA: maximizing the sum correlation between activations A_1 and A_2 and enforcing orthonormality constraints within A_1 and A_2. We achieve both by transforming the CCA objective and present three methods that progressively relax the orthogonality constraints.

We further improve upon DCCA by enabling computation across mini-batches for improved flexibility and test performance. DCCA was developed for very large batches because correlation is not separable across batches. While large mini-batch implementations of stochastic gradient optimization increase the opportunity for parallelism and therefore increased computational efficiency, small batch training provides a more up-to-date gradient calculation, increasing the range of suitable learning rates and improving test accuracy (Masters & Luschi, 2018). By reformulating the correlation objective as the ℓ_2 distance (following SoftCCA), it becomes separable across mini-batches. We ensure a normalization to one by using batch normalization without the scale and shift parameters (Ioffe & Szegedy, 2015). The remaining complication for mini-batch optimization is the orthogonality constraints, which each of our three solutions handles differently.

Task-driven Objective. We start with feed-forward networks $A_1 = f_1(X_1; \theta_1)$ and $A_2 = f_2(X_2; \theta_2)$ applied to each modality X_1 and X_2, and a task-driven objective that operates on the outputs A_1 and A_2, respectively. Task-driven functions $f_{\text{task1}}(A_1; \theta_{task})$ and $f_{\text{task2}}(A_2; \theta_{\text{task}})$, then perform the discriminative task using the activations A_1 and A_2. Networks f_1 and f_2 are optimized so that the ℓ_2 distance between A_1 and A_2 is minimized; therefore, f_{task1} and f_{task2} are trained to operate on equivalent inputs and can share parameters θ_{task}. We combine the CCA and task-driven objectives as a weighted sum with a hyperparameter for tuning. This model is flexible, in that the task-driven goal can be for classification (Krizhevsky et al., 2012; Dorfer et al., 2016a), regression (Katzman et al., 2016), clustering (Caron et al., 2018), or some other task entirely.

Orthogonality Constraints. We present three variations of Discriminative Deep CCA, each handling the orthogonality constraints of CCA in a different way: 1) using whitening to achieve orthogonality (DDCCA-W), 2) relaxing the orthogonality with soft decorrelation (DDCCA-SD), and 3) removing all explicit decorrelation (DDCCA-ND).

1) DDCCA-W: Whitening. CCA applies orthogonality constraints to A_1 and A_2. We accomplish this with a linear transformation called whitening that transforms the activations such that their covariance becomes the identity matrix - that is, the features are uncorrelated. Decorrelated Batch Normalization (DBN) has previously been used to regularize a deep model by decorrelating features (Huang et al., 2018) and inspired our solution. We apply a transformation $B = UA$ to make B orthonormal: $BB^T = I$.

The Zero-phase Component Analysis (ZCA) whitening transform that we use performs three steps: rotate the data to decorrelate it, rescale each axis, and rotate back to the original space. Each of these transformations is learned from the data. The covariance matrix of A is computed as $\Sigma = \frac{1}{n-1}AA^T$. Any matrix $U \epsilon \mathbb{R}^{d_o \times d_o}$ that satisfies the condition $U^T U = \Sigma^{-1}$ whitens the data; however, U is only defined up to a rotation, so it is not unique. PCA whitening follows the first two steps and uses the eigendecomposition of covariance matrix Σ: $U_{PCA} = \Lambda^{-1/2} V^T$ for $\Lambda = \text{diag}(\lambda_1, ..., \lambda_{d_o})$ and $V = [v_1, ..., v_1]$ where (λ_i, v_i) are the eigenvalue, eigenvector pairs of Σ. However, PCA whitening suffers from stochastic axis swapping in which the neurons are not stable from one batch to the next (Huang et al., 2018). We instead use ZCA whitening as recommended by Huang et al. and used in their DBN method (Huang et al., 2018). ZCA uses the transformation $U_{ZCA} = V \Lambda^{-1/2} V^T$ in which PCA whitening is first applied, followed by a rotation back to the original space. Adding the rotation V brings the whitened data B as close as possible to the original data A (Kessy et al., 2015).

Computation of U_{ZCA} is clearly dependent upon covariance matrix Σ. While Huang et al. used a running average of transformation matrix U_{ZCA} over mini-batches (Huang et al., 2018), we apply this stochastic approximation to covariance matrix Σ for each modality using the update

$$\Sigma^{(k)} = \alpha \Sigma^{(k-1)} + (1 - \alpha)\Sigma^b \tag{6}$$

for batch k where Σ^b is the covariance matrix for the current batch. We then compute the ZCA transformation from $\Sigma^{(k)}$ to do whitening:

$$B = f_{\text{ZCA}}(A) = U_{\text{ZCA}}^{(k)} A \, .$$

At test time, matrix $U^{(k)}$ from the last training batch is used. Algorithm 1 describes the forward pass for ZCA whitening in more detail.

Algorithm 1 Whitening layer for orthogonality.

Input: activations $A \epsilon \mathbb{R}^{d_o \times n}$
Hyperparameters: mini-batch size m, momentum α
Parameters of layer: mean μ, covariance Σ
if training **then**
  $\mu \leftarrow \alpha\mu + (1-\alpha)\frac{1}{m} A \, 1_{n \times 1}$ // Update mean
  $\bar{A} = A - \mu$ // Mean center data
  $\Sigma \leftarrow \alpha\Sigma + (1-\alpha)\frac{1}{m-1}\bar{A}_1 \bar{A}_2^T$ // Update covariance
  $\hat{\Sigma} \leftarrow \Sigma + \epsilon I$ // Add ϵI for numerical stability
  $\Lambda, V \leftarrow \text{eig}(\hat{\Sigma})$ // Compute eigendecomposition
  $U \leftarrow V \Sigma^{-1/2} V^T$ // Compute transformation matrix
else
  $\bar{A} \leftarrow A - \mu$ // Mean center data
end if
$B \leftarrow U\bar{A}$ // Apply ZCA whitening transform
return B

The loss function for DDCCA-W integrates both the correlation and task-driven objectives with decorrelation performed by whitening:

$$\mathcal{L}_{\text{task}}(f_{\text{task}}(B_1), Y) + \mathcal{L}_{\text{task}}(f_{\text{task}}(B_2), Y) + \lambda \, \mathcal{L}_{\ell_2 \text{ dist}}(B_1, B_2)$$

where B_1 and B_2 are whitened outputs of A_1 and A_2, respectively. The whole network is trained end-to-end.

2) DDCCA-SD: Soft Decorrelation. While fully independent components may be beneficial in regularizing a DNN on some data sets, a softer decorrelation may be more suitable on others. In this formulation we relax the orthogonality constraints using regularization, following the Decorr loss of SoftCCA (Chang et al., 2018). The loss function for this formulation is then

$$\mathcal{L}_{\text{task}}(f_{\text{task}}(A_1), Y) + \mathcal{L}_{\text{task}}(f_{\text{task}}(A_2), Y) + \lambda_1 \mathcal{L}_{\ell_2 \text{ dist}}(A_1, A_2) + \lambda_2 \left(\mathcal{L}_{\text{Decorr}}(A_1) + \mathcal{L}_{\text{Decorr}}(A_2)\right) \, .$$

3) DDCCA-ND: No Decorrelation. When CCA is used in an unsupervised manner, some sort of orthogonality constraints or decorrelation is necessary to ensure that networks f_1 and f_2 do not simply produce multiple copies of the same feature. While this result could maximize the sum correlation, it is not helpful in capturing useful projections. In the task-driven setting, the decorrelation term helps to ensure that the features in f_1 and f_2 capture properties that are discriminative and therefore not replicates of the same information. This formulation removes the decorrelation term entirely, forming an even simpler objective:

$$\mathcal{L}_{\text{task}}(f_{\text{task}}(A_1), Y) + \mathcal{L}_{\text{task}}(f_{\text{task}}(A_2), Y) + \lambda \mathcal{L}_{\ell_2 \text{ dist}}(A_1, A_2) \, .$$

This allows us to test whether the soft decorrelation term provides a beneficial regularization in a task-driven model.

4. Experiments

We validated our method on three different data sets: MNIST handwritten digits, the Carolina Breast Cancer Study (CBCS) using imaging and genomic features, and speech data from the Wisconsin X-ray Microbeam Database (XRMB). Our experiments show the utility of our method for cross-modal classification, regularization during training when only one modality is available at test time, and semi-supervised learning. Further, we demonstrate robustness to small training set size and small mini-batch size.

4.1. Implementation Details

Each layer of our network consists of a fully connected layer, followed by the ReLu activation function and batch normalization (Ioffe & Szegedy, 2015). Our implementations of DCCA, SoftCCA, and Joint DCCA/DeepLDA (Dorfer et al., 2016b) also use ReLu activation and batch normalization. We also modified the CCAL approach to use a softmax function and cross entropy loss for classification instead of their

pairwise ranking loss for retrieval (Dorfer et al., 2018). We used the Nadam optimizer and trained for 200 epochs on MNIST, 400 on CBCS, and 100 on XRMB. Hidden layer size was set to 500 for MNIST, 200 for CBCS, and 1000 for XRMB, with 0 to 4 layers for each modality. Output layer size was set to 50 for MNIST and CBCS; it was set to 112 for XRMB. We learned the hyperparameters on a validation set by random search, including number of hidden layers, loss function weight λ, momentum α, ℓ_2 regularizer, learning rate, and batch size. We used Keras with the Theano backend.

4.2. Cross-modal Classification on MNIST Digits

We formed a multimodal data set from the MNIST handwritten digit image data set (LeCun, 1998). Following Andrew et al., we split each 28×28 image in half horizontally, creating left and right modalities that are each 14×28 pixels (Andrew et al., 2013). Each modality was flattened into a vector with 392 features. The full data set consists of 60k training images and 10k test images. We used a random set of up to 50k for training and the remaining training images for validation. We used the full 10k image test set.

We tested the cross-modal classification accuracy by first using each model to compute the projection for each modality, then we trained a linear SVM on one modality projection, and finally we used the other modality projection at test time. While the task-driven method presented in this work learns a classifier within the model, this test setup enables a comparison with the unsupervised forms of CCA and validates the discriminativity of the features learned. We report the mean sum correlation or classification accuracy over five randomly selected training/validation sets; the test set always remained the same.

Correlation vs. Classification Accuracy. We first demonstrate the importance of adding a task-driven component to deep CCA by showing that maximizing the sum correlation between modalities is not sufficient and can even be misleading. Figure 2 plots the sum correlation vs. cross-modal classification accuracy across many different hyperparameter settings for DCCA (Andrew et al., 2013) and SoftCCA (Chang et al., 2018). We used 50 components for each, and thus the maximum possible sum correlation was 50. The sum correlation was measured after applying linear CCA to the network projections to ensure that components are independent.

With DCCA, a larger correlation tended to produce a larger classification accuracy, but there was still a large variance in classification accuracy amongst hyperparameter settings that produced a similar sum correlation. Take, for example, the two farthest right points in the plot (colored red): their classification accuracy differs by 10%, and they are not even the points with the best classification accuracy (colored purple). Further, some settings with a very small sum correlation resulted in a very high classification accuracy. The pattern is rather different for SoftCCA. There was an increase in classification accuracy as sum correlation increased but only up to a point. For higher sum correlations, the classification accuracy varied even more from 20% to 80%. Further experiments (not shown) have indicated that when the sole objective is correlation, some of the projection directions are simply not discriminative, particularly when there are a large number of classes. Optimizing for sum correlation alone does not guarantee a model with the highest cross-modal classification accuracy.

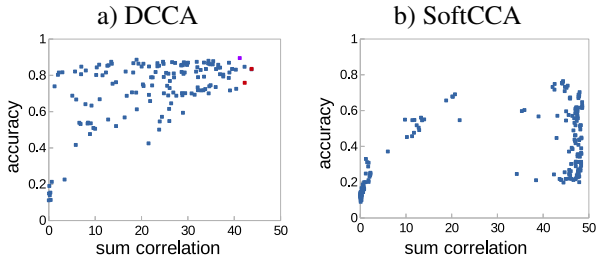

Figure 2. Sum correlation vs. cross-modal classification accuracy across many different hyperparameter settings on a training set size of 10,000. a) DCCA (Andrew et al., 2013), b) SoftCCA (Chang et al., 2018)

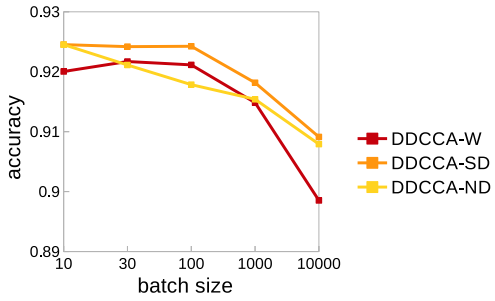

Figure 3. The effect of batch size on classification accuracy for each task-driven multimodal method on MNIST split with a training set size of 10,000.

Mini-batch Size. Our DDCCA method was designed to be run on any batch size in order get the best performance from small batches. This experiment verifies that small batch training is best for DDCCA. Figure 3 plots the batch size vs. classification accuracy for a training set size of 10,000. Batch sizes of 10, 30, 100, 1000, and 10,000 were tested and showed that a batch size of 10 or 30 produced the best result for all three variations of DDCCA. This is in line with previous work by Masters and Luschi that found the best performance with a batch size between 2 and 32 (Masters & Luschi, 2018).

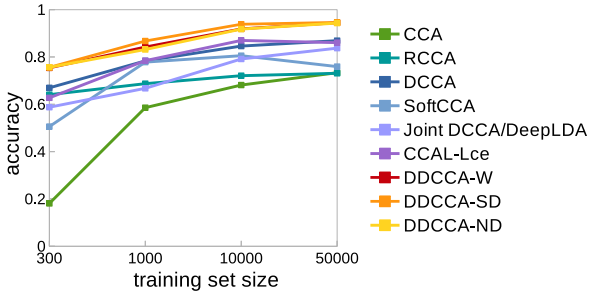

Figure 4. The effect of training set size on classification accuracy for each multimodal method on MNIST split.

Training Set Size. We manipulated the training set size in order to study the robustness of our method to smaller data sets. Figure 4 shows the cross-modal classification accuracy for training set sizes of $n = 300$, 1000, $10,000$, and $50,000$. While it is expected that performance will decrease for smaller training set sizes, some methods are more susceptible to this degradation than others. As expected, the classification accuracy with CCA dropped significantly for $n = 300$ and 1000 as the covariance and cross-covariance matrices were not stable and the training data was overfit. SoftCCA was also particularly susceptible for $n = 300$. Prior work on this method did not test such small training set sizes (Chang et al., 2018).

Across all training set sizes, our DDCCA variations were consistently the top performers, for example, increasing classification accuracy from 78.3% to 86.7% for $n = 1000$. Increases in classification accuracy over DDCCA-ND were small, indicating that the different decorrelation schemes have only a small effect on this data set; the task-driven component is the main reason for the success of our method. In particular, the classification accuracy with $n = 1000$ did better than the unsupervised method DCCA on $n = 10,000$. Further, DDCCA with $n = 300$ did better than linear methods on $n = 50,000$, showing the benefits of both the task-driven and deep components of our model.

Visualization. We also examined the CCA projections qualitatively by plotting them in 2D with t-distributed Stochastic Neighbor Embedding (t-SNE) (Van Der Maaten & Hinton, 2008). Figure 5 shows the CCA projection of the left modality for each method. As expected, the task-driven method produces more clearly separated classes.

4.3. Regularization for Imaging and Genomic Data Classification

We further studied our method on a small data set with only 1003 patient samples using image and genomic data from CBCS, Phase 3 (Troester et al., 2018). Images consisted of four cores per patient from a tissue microarray that was stained with hematoxylin and eosin. Image features were

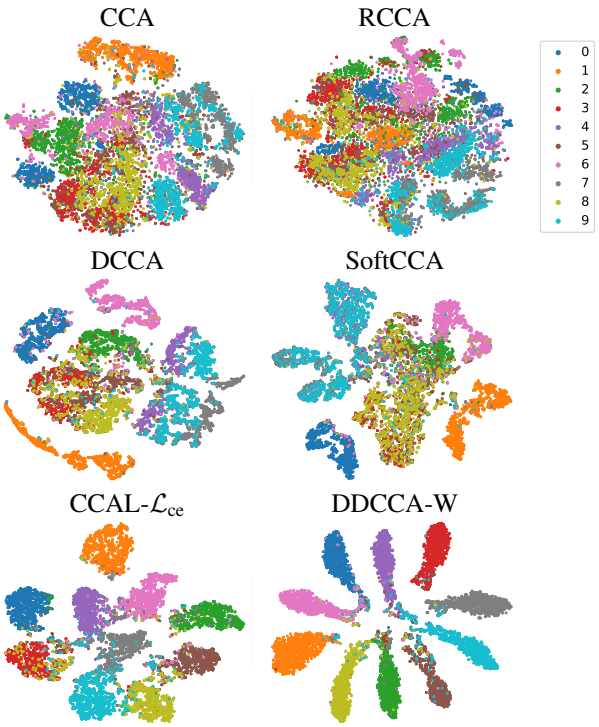

Figure 5. t-SNE plots for CCA methods on MNIST split. Each of the methods was used to compute projections for the left and right modalities (left and right sides of the images) using 10,000 training examples. The plots show a 2D visualization of the projection for the left modality computed with t-SNE with each digit colored differently. The samples for each digit are better clustered with the task-driven methods than with the unsupervised ones. DDCCA-SD and DDCCA-ND (not shown) produced similar results to DDCCA-W.

extracted with the Convolutional Neural Network VGG16 (Simonyan & Zisserman, 2015) by taking the mean of the 512-dimensional output of the fourth set of convolutional layers across the tissue region and further averaging across all core images for the same patient. For gene expression, we used the set of 50 genes in the PAM50 array (Parker et al., 2009) or a larger set of 163 genes (referred to as GE163). The data set was randomly split into half for training, one quarter for validation, and one quarter for testing. Classification tasks included predicting Basal vs. non-Basal genomic subtype, estrogen receptor (ER) status positive vs. negative, and grade 1 vs. 3.

We tested DDCCA as a method of regularization. If two modalities are available for training but only one at test time, the additional modality may help to regularize the model. We tested different classifier training methods when only images were available at test time: a) a linear SVM trained on image features, b) a DNN trained on image features, c) DDCCA trained on image features and PAM50, d) DDCCA trained on image features and GE163. Table 2 provides the classification accuracy for each method. While

Method	Training data	Basal	ER	Grade
Linear SVM	Image only	0.785 (0.004)	0.838 (0.003)	0.897 (0.006)
DNN	Image only	0.796 (0.007)	**0.852 (0.008)**	0.907 (0.009)
DDCCA-W	Image+PAM50	0.827 (0.006)	0.839 (0.007)	**0.911 (0.012)**
DDCCA-SD	Image+PAM50	0.820 (0.010)	0.826 (0.009)	0.859 (0.021)
DDCCA-W	Image+GE163	**0.840 (0.010)**	0.838 (0.009)	0.910 (0.017)
DDCCA-SD	Image+GE163	0.812 (0.011)	0.815 (0.020)	0.891 (0.020)

Table 2. Classification accuracy for different methods to predict from images only at test time. Linear SVM and DNN were trained on only images. DDCCA-W and DDCCA-SD were used to regularize with PAM50 or GE163 during training. The standard error is in brackets.

ER and grade only showed a small improvement beyond a linear SVM or DNN, or sometimes no improvement at all, genomic subtype Basal showed a much larger improvement of up to 5%. In particular, PAM50 and GE163 helped to regularize the DDCCA model. This demonstrates that having additional information at training time can boost model performance at test time - dependent upon the additional data available and what is being predicted.

This experiment used a static set of pre-trained CNN image features in order to assess the utility of the method. The CNN itself could be fine-tuned end-to-end with our DDCCA model, providing an easy opportunity for data augmentation and likely further improvements in classification accuracy.

4.4. Semi-supervised Learning for Speech Recognition

Our final set of experiments uses speech data from XRMB, consisting of simultaneously recorded acoustic and articulatory measurements. Prior work has shown that CCA-based algorithms can improve phonetic recognition (Wang et al., 2015b;a; 2016; Dorfer et al., 2016b). The 45 speakers are split into 35 for training, 2 for validation, and 8 for testing, providing a total of 1,429,236 samples for training, 85,297 for validation, and 111,314 for testing.[1] The acoustic features are 112D and the articulatory ones are 273D. We removed the per-speaker mean and variance for both modalities. Each sample is annotated with one of 38 phonetic labels.

Our goal on this data set is not cross-modal classification, but multimodal classification - this is, using both modalities of data to find a discriminative feature set. We trained each model using both modalities and their labels. To test each CCA model, we followed prior work and concatenated the original input features from both modalities with the projections from both modalities. Due to the large training set size, we used a Linear Discriminant Analysis (LDA) classifier for efficiency. The same feature construction was used at test time. This setup was used to test whether a task-driven deep CCA model can improve feature discriminativity. We tested DDCCA with a task-driven loss of LDA (Dorfer et al., 2016a) or softmax to demonstrate the flexibility of our model.

[1] http://ttic.uchicago.edu/~klivescu/XRMB_data/full/README

Method	Task	Accuracy
Baseline	-	0.591
CCA	-	0.589
RCCA	-	0.588
DCCA	-	0.620
SoftCCA	-	0.635
Joint DCCA/DeepLDA	LDA	0.633
CCAL-$\mathcal{L}_{ce}$	Softmax	0.598
DDCCA-W	LDA	0.710
DDCCA-SD	LDA	0.677
DDCCA-ND	LDA	0.677
DDCCA-W	Softmax	**0.795**
DDCCA-SD	Softmax	0.785
DDCCA-ND	Softmax	0.785

Table 3. XRMB classification results. Using each method, the projections were computed for each modality. The original input data and the projections from both modalities were concatenated and used to train an LDA classifier. The methods were tested with the same feature construction on the test data.

Labeled data	100%	30%	10%	3%	1%
Accuracy	0.795	0.762	0.745	0.684	0.637

Table 4. Semi-supervised classification results on XRMB using DDCCA-W. The model was trained using only a percentage of the sample labels, selected randomly.

Table 3 compares the classification accuracy for each method. The baseline result used only the original input features in training LDA. Deep methods DCCA and SoftCCA improved upon the linear methods. All DDCCA variations significantly outperformed previous state-of-the-art methods. The softmax task consistently beat LDA by a large margin. DDCCA-SD and DDCCA-ND produced equivalent results as a weight of 0 on the decorrelation term performed best. However, DDCCA-W showed the best result, with an improvement of 16% over the best unsupervised method.

DDCCA can also be used in a semi-supervised manner when labels are available for only some samples. Table 4 shows the results for DDCCA-W in this setting. With 0% labeled data, the result would be similar to DCCA. A large improvement over the unsupervised results in Table 3 is seen even with labels for only 10% of the training samples.

5. Discussion

We proposed a method to find a shared latent space that is also discriminative by adding a task-driven component to deep CCA while enabling end-to-end training. This was accomplished by replacing the CCA projection with ℓ_2 distance minimization and orthogonality constraints on the activations. We presented three different techniques for applying the orthogonality constraints: learning a whitening transformation (DDCCA-W), decorrelating the data by penalizing the off-diagonal elements of the covariance matrix (DDCCA-SD), or applying no explicit decorrelation (DDCCA-ND). DDCCA-W or DDCCA-SD performed the best, dependent on the data set - both of which include some means of decorrelation to provide an extra regularizing effect to the model and thereby outperforming DDCCA-ND.

DDCCA showed large improvements over state-of-the-art methods in cross-modal classification accuracy on MNIST and significantly increased robustness when the training set size was small. On a cancer imaging and genomic data set, DDCCA provided a regularizing effect when both modalities were available for training but only one was available at test time. DDCCA also produced a large increase over state-of-the-art in classification accuracy on a much larger data set, XRMB. On this data set we also demonstrated a semi-supervised approach to get a large increase in classification accuracy with only a small proportion of the labels. Using a similar technique, our method could also be applied when some samples are missing a second modality.

Classification tasks using a softmax operation or LDA were explored in this work; however, the formulation presented can also be used with other tasks such as regression or clustering. Another possible avenue for future work entails extending beyond the current focus of shared features. While shared features are essential for cross-modal classification, further insight into the data can be gained from also learning features that are unique to each modality. This approach has been developed for dictionary learning (Lock et al., 2013; Ray et al., 2014) but could be extended to deep CCA-based methods. Finally, we have yet to apply data augmentation to the proposed framework; this could provide a significant benefit for small training sets.

References

Andrew, G., Arora, R., Bilmes, J., and Livescu, K. Deep Canonical Correlation Analysis. In *Proc. ICML*, 2013.

Bahrampour, S., Nasrabadi, N. M., Ray, A., and Jenkins, W. K. Multimodal Task-Driven Dictionary Learning for Image Classification. *arXiv preprint: 1502.01094*, 2015.

Bhatt, G., Jha, P., and Raman, B. Common Representation Learning Using Step-based Correlation Multi-Modal CNN. *arXiv preprint: 1711.00003*, 2017.

Bie, T. D., Cristianini, N., and Rosipal, R. Eigenproblems in pattern recognition. In *Handbook of Geometric Computing*, pp. 129–167. Springer Berlin Heidelberg, 2005.

Bilenko, N. Y. and Gallant, J. L. Pyrcca: regularized kernel canonical correlation analysis in Python and its applications to neuroimaging. *Frontiers in Neuroinformatics*, 10, nov 2016.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In *Proc. ECCV*, 2018.

Cha, M., Gwon, Y., and Kung, H. T. Multimodal sparse representation learning and applications. *arXiv preprint: 1511.06238*, 2015.

Chandar, S., Khapra, M. M., Larochelle, H., and Ravindran, B. Correlational Neural Networks. *Neural Computation*, 28(2):257–285, feb 2016.

Chang, X., Xiang, T., and Hospedales, T. M. Scalable and Effective Deep CCA via Soft Decorrelation. In *Proc. CVPR*, 2018.

Dorfer, M., Kelz, R., and Widmer, G. Deep linear discriminant analysis. In *Proc. ICLR*, 2016a.

Dorfer, M., Widmer, G., and At, G. W. Towards Deep and Discriminative Canonical Correlation Analysis. In *Proc. ICML Workshop on Multi-view Representaiton Learning*, 2016b.

Dorfer, M., Schlüter, J., Vall, A., Korzeniowski, F., and Widmer, G. End-to-end cross-modality retrieval with CCA projections and pairwise ranking loss. *International Journal of Multimedia Information Retrieval*, 7(2):117–128, jun 2018.

Duan, K., Zhang, H., and Wang, J. J. Y. Joint learning of cross-modal classifier and factor analysis for multimedia data classification. *Neural Computing and Applications*, 27(2):459–468, feb 2016.

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

Huang, L., Yang, D., Lang, B., and Deng, J. Decorrelated Batch Normalization. In *Proc. CVPR*, 2018.

Ioffe, S. and Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. ICML*, 2015.

Kan, M., Shan, S., Zhang, H., Lao, S., and Chen, X. Multi-view Discriminant Analysis. *IEEE PAMI*, 2015.

Katzman, J., Shaham, U., Cloninger, A., Bates, J., Jiang, T., and Kluger, Y. Deep Survival: A Deep Cox Proportional Hazards Network. *arxiv preprint: 1606.00931*, 2016.

Kessy, A., Lewin, A., and Strimmer, K. Optimal whitening and decorrelation. *arXiv preprint: 1512.00809*, 2015.

Krizhevsky, A., Sutskever, I., and Hinton, G. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1106–1114, 2012.

LeCun, Y. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

Lee, G., Singanamalli, A., Wang, H., Feldman, M. D., Master, S. R., Shih, N. N. C., Spangler, E., Rebbeck, T., Tomaszewski, J. E., and Madabhushi, A. Supervised multi-view canonical correlation analysis (sMVCCA): integrating histologic and proteomic features for predicting recurrent prostate cancer. *IEEE Transactions on Medical Imaging*, 34(1):284–97, jan 2015.

Li, D., Dimitrova, N., Li, M., and Sethi, I. K. Multimedia content processing through cross-modal association. In *Proc. ACM International Conference on Multimedia*, 2003.

Lock, E. F., Hoadley, K. A., Marron, J. S., and Nobel, A. B. Joint and Individual Variation Explained (JIVE) for Integrated Analysis of Multiple Data Types. *The Annals of Applied Statistics*, 7(1):523–542, mar 2013.

Masters, D. and Luschi, C. Revisiting Small Batch Training for Deep Neural Networks. *arxiv preprint: 1804.07612*, 2018.

Parker, J. S., Mullins, M., Cheang, M. C., Leung, S., Voduc, D., Vickery, T., Davies, S., Fauron, C., He, X., et al. Supervised risk predictor of breast cancer based on intrinsic subtypes. *Journal of Clinical Oncology*, 27(8): 1160–1167, 2009.

Ray, P., Zheng, L., Lucas, J., and Carin, L. Bayesian joint analysis of heterogeneous genomics data. *Bioinformatics*, 30(10):1370–6, may 2014.

Shekhar, S., Patel, V. M., Nasrabadi, N. M., and Chellappa, R. Joint sparse representation for robust multimodal biometrics recognition. *IEEE PAMI*, 36(1):113–26, jan 2014.

Simonyan, K. and Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proc. ICLR*, 2015.

Singanamalli, A., Wang, H., Lee, G., Shih, N., Rosen, M., Master, S., Tomaszewski, J., Feldman, M., and Madabhushi, A. Supervised multi-view canonical correlation analysis: fused multimodal prediction of disease diagnosis and prognosis. In *Proc. SPIE Medical Imaging*, 2014.

Troester, M., Sun, X., Allott, E. H., Geradts, J., Cohen, S. M., Tse, C. K., Kirk, E. L., Thorne, L. B., Matthews, M., Li, Y., Hu, Z., Robinson, W. R., Hoadley, K. A., Olopade, O. I., Reeder-Hayes, K. E., Earp, H. S., Olshan, A. F., Carey, L., and Perou, C. M. Racial differences in PAM50 subtypes in the Carolina Breast Cancer Study. *Journal of the National Cancer Institute*, 2018.

Van Der Maaten, L. and Hinton, G. Visualizing high-dimensional data using t-sne. journal of machine learning research. *Journal of Machine Learning Research*, 9:26, 2008.

Wang, W., Arora, R., Livescu, K., and Bilmes, J. On deep multi-view representation learning. In *Proc. ICML*, 2015a.

Wang, W., Arora, R., Livescu, K., and Bilmes, J. A. Unsupervised learning of acoustic features via deep canonical correlation analysis. In *Proc. ICASSP*, 2015b.

Wang, W., Arora, R., Livescu, K., and Srebro, N. Stochastic optimization for deep CCA via nonlinear orthogonal iterations. In *Proc. Allerton Conference on Communication, Control, and Computing*, 2016.

Xu, X., Shimada, A., Taniguchi, R.-i., and He, L. Coupled dictionary learning and feature mapping for cross-modal retrieval. In *Proc. International Conference on Multimedia and Expo*, 2015.

Yang, Z. and Michailidis, G. A Non-negative Matrix Factorization Method for Detecting Modules in Heterogeneous Omics Multi-modal Data. *Bioinformatics*, 2015.

Zhou, G., Cichocki, A., Zhang, Y., and Mandic, D. P. Group Component Analysis for Multiblock Data: Common and Individual Feature Extraction. *IEEE Transactions on Neural Networks and Learning Systems*, 2015.