RESEARCH

Open Access

Interpretable deep learning methods for multiview learning

Hengkang Wang[1], Han Lu[2], Ju Sun[1] and Sandra E. Safo[2*]

*Correspondence:
ssafo@umn.edu

[1] Department of Computer Science and Engineering, University of Minnesota, Minneapolis 55455, USA
[2] Division of Biostatistics and Health Data Science, University of Minnesota, Minneapolis 55414, USA

Abstract

Background: Technological advances have enabled the generation of unique and complementary types of data or views (e.g. genomics, proteomics, metabolomics) and opened up a new era in multiview learning research with the potential to lead to new biomedical discoveries.

Results: We propose iDeepViewLearn (Interpretable Deep Learning Method for Multiview Learning) to learn nonlinear relationships in data from multiple views while achieving feature selection. iDeepViewLearn combines deep learning flexibility with the statistical benefits of data and knowledge-driven feature selection, giving interpretable results. Deep neural networks are used to learn view-independent low-dimensional embedding through an optimization problem that minimizes the difference between observed and reconstructed data, while imposing a regularization penalty on the reconstructed data. The normalized Laplacian of a graph is used to model bilateral relationships between variables in each view, therefore, encouraging selection of related variables. iDeepViewLearn is tested on simulated and three real-world data for classification, clustering, and reconstruction tasks. For the classification tasks, iDeepViewLearn had competitive classification results with state-of-the-art methods in various settings. For the clustering task, we detected molecular clusters that differed in their 10-year survival rates for breast cancer. For the reconstruction task, we were able to reconstruct handwritten images using a few pixels while achieving competitive classification accuracy. The results of our real data application and simulations with small to moderate sample sizes suggest that iDeepViewLearn may be a useful method for small-sample-size problems compared to other deep learning methods for multiview learning.

Conclusion: iDeepViewLearn is an innovative deep learning model capable of capturing nonlinear relationships between data from multiple views while achieving feature selection. It is fully open source and is freely available at https://github.com/lasandrall/iDeepViewLearn.

Keywords: Data integration, Integrative analysis, Data fusion, Feature ranking or selection, Graph Laplacian

Background

Multiview learning has garnered considerable interest in biomedical research, thanks to advances in data collection and processing. Here, for the same individual, different sets of data or views (e.g., genomics, imaging) are collected, and the main interest lies in learning low-dimensional representation(s) common to all views or specific to each view that together explain the overall dependency structure among the different views. Downstream analyses typically use the learned representations in supervised or unsupervised algorithms. For example, if a categorical outcome is available, then the learned low-dimensional representations could be used for classification. If no outcome is available, the low-dimensional representations could be used in clustering algorithms to cluster the samples.

Existing methods

The literature on multiview learning is not scarce. Linear and nonlinear methods have been proposed to associate multiview data. For example, canonical correlation analysis (CCA) methods have been proposed to maximize the correlation between linear projections of two views [1, 2]. The kernel version of CCA (KCCA) has also been proposed to maximize the correlation between nonlinear functions of the views while restricting these nonlinear functions to reside in reproducing kernel Hilbert spaces [3, 4]. Deep learning methods, which offer more flexibility than kernel methods, have been proposed to learn flexible nonlinear representations of two or more views, via deep neural networks (DNNs). Examples of such methods include Deep CCA [5], Deep generalized CCA [for three or more views] [6] and DeepIMV [7].

Despite the success of DNN and kernel methods, their main limitation is that they do not yield interpretable findings. In particular, if these methods are applied to our motivating data, it will be difficult to determine the genes and CpG sites that contribute the most to the dependency structure in the data. This is important for interpreting the results of downstream analysis that use these methods and for determining key molecules that discriminate between those who died from breast cancer and those who did not.

Few interpretable deep-leaning methods for multiview learning have been proposed in the literature. In [8], a data integration and classification method (MOMA) was proposed for multiview learning that uses the attention mechanism for interpretability. Specifically, MOMA builds a module (e.g. gene set) for each view and uses the attention mechanism to identify modules and features relevant to a certain task.

In [9], a deep learning method was proposed to jointly associate data from multiple views and discriminate subjects that allows for feature ranking. The authors considered a homogeneous ensemble approach for feature selection that allowed the ranking of features based on their contributions to the overall dependency among views and the separation of classes within a view. It is noteworthy that variable selection in MOMA and Deep IDA is data driven, and the algorithm for MOMA is applicable to two views, which is very restrictive.

Our approach

In this article, we propose a deep learning framework to associate data from two or more views while achieving feature selection. Similar to deep generalized CCA [deep GCCA] [6] and unlike deep CCA [5], we learn low-dimensional representations that are common to all views. However, unlike deep GCCA, we assume that each view can be approximated by a nonlinear function of the shared low-dimensional representations. We use deep neural networks to model the nonlinear function and construct an optimization problem that minimizes the difference between the observed and the nonlinearly approximated data, while imposing a regularization penalty on the reconstructed data. This allows us to reconstruct each view using only the relevant variables in each view. As a result, the proposed method allows the selection of variables in the views and enhances our ability to identify features from each view that contribute to the association of the views. The results of our motivating data and simulations with small sample sizes suggest that the proposed method may be a useful method for small-sample-size problems compared to other deep learning methods for associating multiple views. Beyond the data-driven approach to feature selection, we also consider a knowledge-based approach to identify relevant features. In the statistical learning literature, the use of prior information (e.g., biological information in the form of variable–variable interactions) in variable selection methods has the potential to identify correlated variables with greater ability to produce interpretable results and improve prediction or classification estimates [10, 11]. As such, we use the normalized Laplacian of a graph to model bilateral relationships between variables in each view and to encourage the selection of variables that are connected.

In summary, we have three main contributions. First, we propose a deep learning method for learning nonlinear relationships in multiview data that is capable of identifying relevant features that contribute the most to the association among different views. Our approach can accommodate more than two views, in contrast to MOMA, which requires significant code modifications by users, for the same purpose. Second, we extend this method to incorporate prior biological information to yield more interpretable findings, distinguishing it from existing interpretable deep learning methods for multiview learning, such as MOMA and Deep IDA. To the best of our knowledge, this is one of the first nonlinear-based methods for multiview learning to do so. Third, we provide an efficient implementation of the proposed methods in Pytorch and interface them in R to increase the reach of our algorithm.

The remainder of the paper is organized as follows. In section "Methods", we introduce the proposed method. In section "Simulation experiments", we conduct simulation studies to assess the performance of our methods compared to several existing linear and nonlinear methods. In section "Real-world experiments", we apply our method to the Holm breast cancer study for classification and clustering; we further consider two additional applications: brain lower grade glioma (LGG) data, to demonstrate the use of our method for three views; and MNIST handwriting data, to demonstrate that handwritten digits can be reconstructed with few pixels while maintaining competitive classification accuracy.

Methods

Model formulation

Assume that $d = 1, \ldots, D$ different types of data or views are available from n individuals and organized in D matrices $\mathbf{X}^{(1)} \in \mathbb{R}^{n \times p^{(1)}}, \cdots, \mathbf{X}^{(D)} \in \mathbb{R}^{n \times p^{(D)}}$. For example, for the same set of n individuals in our motivating study, the matrix $\mathbf{X}^{(1)}$ consists of gene expression levels and $\mathbf{X}^{(2)}$ consists of CpG sites. Denote an outcome variable by $\mathbf{y}$, if available. In our motivating study, $\mathbf{y}$ is an indicator variable of whether or not an individual died from breast cancer. We wish to model complex nonlinear relationships between these views via an *informative* joint low-dimensional nonlinear embedding of the original high-dimensional data.

For the sake of clarity, we outline a linear framework which our nonlinear model emulates. Assume that there is a joint embedding (or common factors) $\mathbf{Z} \in \mathbb{R}^{n \times K}$ of the D views that drives the observed variation across the views so that each view is written as a linear function of the joint embedding plus some noise: $\mathbf{X}^{(d)} = \mathbf{Z}\mathbf{B}^{(d)^{\mathrm{T}}} + \mathbf{E}^{(d)}$. Here, K is the number of latent components and $\mathbf{B}^{(d)} \in \mathbb{R}^{p^{(d)} \times K}$ is the loading matrix for view d, each row corresponding to the coefficients K for a specific variable. $\mathbf{E}^{(d)}$ is a matrix of errors incurred by approximating $\mathbf{X}^{(d)}$ with $\mathbf{Z}\mathbf{B}^{(d)^{\mathrm{T}}}$. Let $\mathbf{z}_i \in \mathbb{R}^K$ be the ith row in $\mathbf{Z}$. The common factors $\mathbf{z}_i$ represent K different driving factors that predict all variables in all views for the ith subject, thus inducing correlations between views. When we write $\mathbf{X}^{(d)} \approx \mathbf{Z}\mathbf{B}^{(d)^{\mathrm{T}}}$ for $d = 1, \ldots, D$, we assume that there is an "intrinsic" space $\mathbb{R}^K$ so that each sample is represented as $\mathbf{z} \in \mathbb{R}^K$. For each $d = 1, \ldots, D$, $\mathbf{x}^{(d)}$ is an instance in $\mathbf{X}^{(d)}$, and $\mathbf{B}^{(d)}$ maps a low-dimensional representation $\mathbf{z}$ to this $\mathbf{x}^{(d)}$, i.e., restricting the mappings to be linear. Now, we generalize these mappings to be nonlinear, parameterized by neural networks.

For $d = 1, \ldots, D$, let G_d denote the neural network that generalizes $\mathbf{B}^{(d)\mathrm{T}}$ for the view d. As typical neural networks, each of the G_d's is composed of multilayer affine mapping followed by nonlinear activation, i.e., of the form $\mathcal{W}_L \circ \sigma \circ \mathcal{W}_{L-1} \ldots \sigma \circ \mathcal{W}_2 \circ \sigma \circ \mathcal{W}_1$, where σ denotes the nonlinear activation applied element-wise, and $\mathcal{W}_i$' s for $i = 1, \ldots, L$ denote the affine mappings. We prefer to state the affine layers in abstract form, as we can have different types of layer. In this paper, we use G_d consisting of fully-connected and convolutional layers to reconstruct numerical data and images, respectively.

For simplicity, assume that each layer of the dth view network, except the first layer, has c_d units. Let the size of the input layer (first layer) be K, where K is the number of latent components. The output of the first layer for the dth view is a function of the shared low-dimensional representation, $\mathbf{Z}$, and is given by $h_1^{(d)} = \sigma(\mathbf{Z}\mathbf{W}_1^{(d)} + \mathbf{b}_1^{(d)}) \in \mathbb{R}^{n \times c_d}$ where $\mathbf{W}_1^{(d)} \in \mathbb{R}^{K \times c_d}$ is a matrix of weights for view d, $\mathbf{b}_1^{(d)} \in \mathbb{R}^{n \times c_d}$ is a matrix of biases, and $\sigma : \mathbb{R} \longrightarrow \mathbb{R}$ is a nonlinear mapping. The output of the second layer for the dth view is the $h_2^{(d)} = \sigma(h_1^{(d)}\mathbf{W}_2^{(d)} + \mathbf{b}_2^{(d)}) \in \mathbb{R}^{n \times c_d}$, $\mathbf{W}_2^{(d)} \in \mathbb{R}^{K \times c_d}$ matrix of weights, $\mathbf{b}_2^{(d)} \in \mathbb{R}^{n \times c_d}$ matrix of biases. The final output layer for the dth view is given by $G_d(\mathbf{Z}) = \sigma(h_{(K_d-1)}^{(d)}\mathbf{W}_{K_d}^{(d)} + \mathbf{b}_{K_d}^{(d)}) \in \mathbb{R}^{n \times p^{(d)}}$, $h_{(K_d-1)}^{(d)} \in \mathbb{R}^{n \times c_d}$, $\mathbf{W}_{K_d}^{(d)} \in \mathbb{R}^{c_d \times p^{(d)}}$, $\mathbf{b}_{K_d}^{(d)} \in \mathbb{R}^{n \times p^{(d)}}$, and the subscript K_d denotes the Kth hidden layer for the view d. $G_d(\mathbf{Z})$ is a function of the weights and biases of the network.

Our first goal is to approximate each view with a nonlinear embedding of the joint low-dimensional representation in an interpretable manner, i.e., $\mathbf{X}^{(d)} \approx G_d(\mathbf{Z})$. To achieve interpretability, MOMA used the attention mechanism to choose important features. In the statistical learning literature, regularization techniques (e.g., lasso [12], elastic net [13], SCAD [14]) are oftentimes used for variable selection to promote interpretability. We also propose a regularization approach for interpretability. Specifically, we assume that some variables in $\mathbf{X}^{(d)}$ are irrelevant and are not needed in the approximation of $\mathbf{X}^{(d)}$. Thus, the columns of $G_d(\mathbf{Z})$ corresponding to the unimportant variables in $\mathbf{X}^{(d)}$ should be made zero or nearly zero in the nonlinear approximation of $\mathbf{X}^{(d)}$. To achieve this, we adopt the $\ell_{2,1}$ norm from [15] to promote column-wise sparsity for features, where $\ell_{2,1}$ is denoted as follows: $\|\mathbf{X}\|_{2,1} = \sum_{j=1}^{p} \sqrt{\sum_{i=1}^{n} \mathrm{X}_{ij}}$, $\mathbf{X} \in \mathbb{R}^{n \times p}$, where X_{ij} is the *ij*th element in $\mathbf{X}$. Given these assumptions, we propose to solve the following optimization problem: find the parameters of the neural network (weight matrices, biases) defining the neural network G_d, and the shared low-dimensional representation $\mathbf{Z}$, for $d = 1, \ldots, D$ that

$$\min_{\mathbf{W}^{(1)},\ldots,\mathbf{W}^{(D)},\mathbf{b}^{(1)},\ldots,\mathbf{b}^{(D)},\mathbf{Z}} \sum_{d=1}^{D} \left(\|\mathbf{X}^{(d)} - G_d(\mathbf{Z})\|_{2,1} + \lambda^d \|G_d(\mathbf{Z})\|_{2,1} \right). \tag{1}$$

The two terms $\|\mathbf{X}^{(d)} - G_d(\mathbf{Z})\|_{2,1} + \lambda^d \|G_d(\mathbf{Z})\|_{2,1}$ together ensure that we select a subset of columns from $\mathbf{X}^{(d)}$ to approximate $\mathbf{X}^{(d)}$. λ^d's are regularization parameters that could be selected by k-fold cross-validation, where $k = 5$ throughout this paper.

Although $\|\cdot\|_{2,1}$ helps promote column-wise sparsity, we did observe that the columns of $G_d(\mathbf{Z})$ were not exactly zero across all samples but were shrunk towards zero for noise variables, perhaps as a result of our use of an automatic differentiation function. Thus, we proceed as follows to select/rank features. Once we have learned the latent code $\mathbf{Z}$ and neural networks G_1, G_2,..., G_D, we use this information to obtain reconstructed data for the different views, that is, to obtain $G_d(\mathbf{Z})$ for the view d. We then calculate the column-wise l_2 norm of $G_d(\mathbf{Z})$, and choose the top r% columns with the largest column norms as important features for the corresponding view. It is imperative that the variables in each view be on the same scale in order to use this ranking procedure. Thus, we standardize each variable to have mean zero and variance one. We save the indices of important features as $\mathbf{I}_1$, $\mathbf{I}_2$,...,$\mathbf{I}_D$, and we denote the new datasets with the selected indices as $\mathbf{X}'^{(1)}, \mathbf{X}'^{(2)}, \ldots, \mathbf{X}'^{(D)}$.

Compared to existing deep learning methods for associating multiple views (e.g., deep generalized CCA), our formulation (1) is unique because we learn the shared low-dimensional representation, $\mathbf{Z}$, while also selecting important variables in each view that drive the association among views. Similarly to Deep GCCA, and unlike CCA and Deep CCA that only learn linear transformations of each view, we learn $\mathbf{Z}$, which is independent of the views and allows one to reconstruct all the view-specific representations simultaneously. Figure 1 is a schematic representation to train the neural network and select features. After that, downstream analyses can use the learned $\mathbf{Z}$ in classification, regression, and clustering algorithms, as shown in Fig. 2.

Network-based feature selection

We consider a knowledge-based approach to identify potentially relevant variables that drive the dependency structure among views (see Fig. 1). In particular, we use prior knowledge about variable–variable interactions (e.g., protein–protein interactions) in

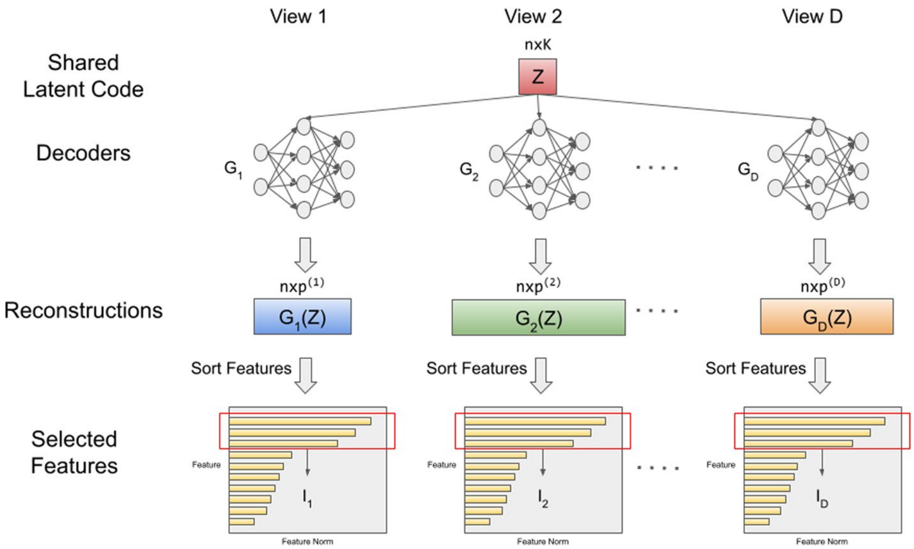

Fig. 1 Feature selection. We train a deep learning model that takes all the views, estimates a shared low-dimensional representation **Z** that drives the variation across the views, and obtains nonlinear reconstructions (G_1 (**Z**),...,G_D (**Z**)) of the original views. We impose sparsity constraints on the reconstructions allowing us to identify a subset of variables for each view (**I**$_1$, ...,**I**$_D$) that approximate the original data

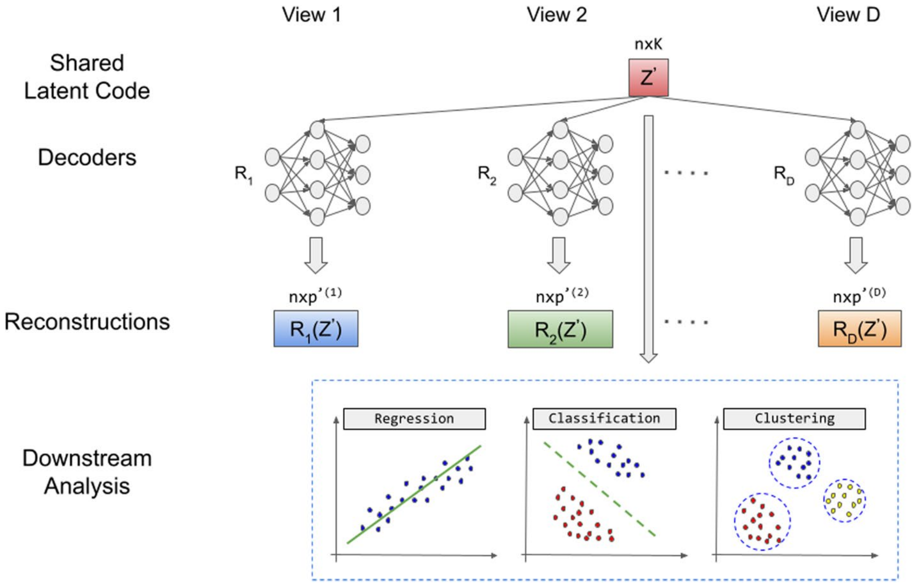

Fig. 2 Reconstruction and downstream analysis. We train a deep learning model to obtain a common low-dimensional representation **Z**′ that is based on the features selected in Algorithm 1, we obtain nonlinear approximations (**R**$_1$ (**Z**),...,**R**$_D$ (**Z**)), and we perform downstream analyses using estimated **Z**′

the estimation of $G_d(\mathbf{Z})$. Incorporating prior knowledge about variable–variable interactions can capture complex bilateral relationships between variables. It has the potential to identify functionally meaningful variables (or networks of variables) within each view for improved prediction performance, as well as aid in interpretation of variables.

There are many databases for obtaining information on variable–variable relationships. One of such database for protein–protein interactions is the Human Protein

Reference Database (HPRD) [16]. We capture the variable–variable connectivity within each view in our deep learning model using the normalized Laplacian [17] obtained from the graph underlying the observed data. Let $\mathcal{G}^{(d)} = (V^{(d)}, E^{(d)}, W^{(d)}), d = 1, 2, \ldots, D$ be a graph network given by a weighted undirected graph. $V^{(d)}$ is the set of vertices corresponding to the $p^{(d)}$ variables (or nodes) for the d-th view. Let $E^{(d)} = \{u \sim v\}$ if there is an edge of variable u to v in the dth view. $W^{(d)}$ is the weight of an edge for the d-th view that satisfies $w(u, v) = w(v, u) \geq 0$. Denote r_v as the degree of vertex v within each view; $r_v = \sum_u w(u, v)$. The normalized Laplacian of $\mathcal{G}^{(d)}$ for the d-th view is $\mathcal{L}^{(d)} = T^{-1/2} L T^{-1/2}$ where L is the Laplacian of a graph defined as

$$L(u, v) = \begin{cases} r_v - w(u, v) & \mathit{if}\ u = v \\ -w(u, v) & \text{if } u \text{ and } v \text{ are adjacent} \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

and T is a diagonal matrix with r_v as the (u,v)-th entry. Given $\mathcal{L}^{(d)}$, we solve the problem:

$$\min_{\mathbf{W}^{(1)},\ldots,\mathbf{W}^{(D)},\mathbf{b}^{(1)},\ldots,\mathbf{b}^{(D)},\mathbf{Z}} \sum_{d=1}^{D} \left(\|\mathbf{X}^{(d)} - G_d(\mathbf{Z})\|_{2,1} + \lambda^d \|G_d(\mathbf{Z})\mathcal{L}^{(d)}\|_{2,1} \right). \tag{3}$$

The normalized Laplacian $\mathcal{L}^{(d)}$ is used as a smoothing operator to smooth the columns in $G_d(\mathbf{Z})$ so that the variables connected in the d-th view are encouraged to be selected together.

Prediction of shared low-dimensional representation and downstream analyses

In this section, we would like to predict the low-dimensional representation shared from the test data $\mathbf{X}^{(d)}_{test}, d = 1, \ldots, D$, (i.e., $\mathbf{Z}_{test}$) and use this information to predict an outcome, $\mathbf{y}$, if available. The schematic graph is shown in Fig. 2. Note that $\mathbf{y}$ can be continuous, binary, or multiclass. We discuss our approach to predict the shared low-dimensional representation, $\mathbf{Z}_{test}$. After getting important features of $\mathbf{X}^{(d)}$ using equation (1) or (3), we extract these features from the original training dataset and form a new training dataset $\mathbf{X}^{'(d)}$. We also form a new testing dataset $\mathbf{X}^{'(d)}_{test} d = 1, \ldots, D$ that consists of the important features. Let $p^{'(d)}$ denote the cardinality of the columns in view d. Since the $\mathbf{Z}$ learned in Equation (1) or (3) is estimated using important and unimportant features, when used in downstream analyses, it can lead to poor results. Therefore, we construct a new shared low-dimensional representation, $\mathbf{Z}'$, which is based only on important features, that is, $\mathbf{X}^{'(d)}, d = 1, \ldots, D$. Because we have already selected a subset of relevant columns of $\mathbf{X}^d$, we are willing to have non-sparse reconstruction results. Therefore, we find $\mathbf{Z}'$ by solving the optimization problem:

$$\min_{\mathbf{W}^{'(1)},\ldots,\mathbf{W}^{'(D)},\mathbf{b}^{'(1)},\ldots,\mathbf{b}^{'(D)},\mathbf{Z}'} \sum_{d=1}^{D} \|\mathbf{X}^{'(d)} - R_d(\mathbf{Z}')\|_F^2, \tag{4}$$

where R_d depends on the weights of the network parameters $\mathbf{W}^{'(d)}$ and the biases $\mathbf{b}^{'(d)}$, and $\|\cdot\|_F$ is the Frobenius norm. Specifically, the final output $R_d = \sigma(h^{(d)}_{(K_d-1)} \mathbf{W}^{'(d)}_{K_d} + \mathbf{b}^{'(d)}_{K_d}) \in \mathbb{R}^{n \times p^{'(d)}}$, the subscript K_d denotes the Kth hidden layer for view d, and the first hidden layer is given as $h_1 = \sigma(\mathbf{Z}'\mathbf{W}^{'(d)}_1 + \mathbf{b}^{'(d)}_1)$.

Suppose that $\widetilde{R_d}\left(\widetilde{\mathbf{Z}'}\right)$ can approximate $\mathbf{X}^{'(d)}$ well for each view, that is, $\mathbf{X}^{'(d)} \approx \widetilde{R_d}\left(\widetilde{\mathbf{Z}'}\right), d = 1, \ldots, D$. Then it is easy to find $\mathbf{X}^{'(d)} \approx \left(\tau\widetilde{R_d}\right)\left(\frac{\widetilde{\mathbf{Z}'}}{\tau}\right)$ for any $\tau \in \mathbb{R}_{\neq 0}$ because $\widetilde{R_d}$ and $\widetilde{\mathbf{Z}'}$ are optimized simultaneously. The lack of control of the scaling of the learned representation $\widetilde{\mathbf{Z}'}$ can lead to robustness problems in downstream analysis, so we add additional constraints on $\mathbf{Z}'$ in equation (4). However, since it is likely that the shape of $\mathbf{Z}'$ is not the same as the latent code learned in the testing stage due to the different number of samples, we put constraints on each row of $\mathbf{Z}'$ (we assume that the number of latent components in the training and testing data is the same) as $\|\mathbf{z}'_i\|_2 \leq 1$ where $\mathbf{z}'_i$ means the i-th row vector of $\mathbf{Z}'$, that is, the latent code of the ith sample. Finally, the optimization problem is as follows:

$$\min_{\mathbf{W}'^{(1)},\ldots,\mathbf{W}'^{(D)},\mathbf{b}'^{(1)},\ldots,\mathbf{b}'^{(D)},\mathbf{Z}'} \sum_{d=1}^{D} \|\mathbf{X}^{'(d)} - R_d\left(\mathbf{Z}'\right)\|_F^2, \text{s.t.}\|\mathbf{z}'_i\|_2 = 1, \quad i = 1, \ldots, n \tag{5}$$

To learn $\mathbf{Z}'_{test}$ from the test data $\mathbf{X}^{'(d)}_{test}, d = 1, \ldots, D$, we use the weights of the learned neural network, $\widetilde{\mathbf{W}}^{'(1)}, \ldots, \widetilde{\mathbf{W}}^{'(D)}$ and biases $\widetilde{\mathbf{b}}^{'(1)}, \ldots, \widetilde{\mathbf{b}}^{'(D)}$ and we solve the following optimization problem for $\widetilde{\mathbf{Z}}_{test}$:

$$\min_{\mathbf{Z}'_{test}} \sum_{d=1}^{D} \|\mathbf{X}^{'(d)}_{test} - \widetilde{R}_d\left(\mathbf{Z}'_{test}\right)\|_F^2, \ \text{s.t.}\|\mathbf{z}'_{test_i}\|_2 = 1, \quad i = 1, \ldots, n. \tag{6}$$

Here, $\mathbf{z}'_{test_i}$ is the ith row vector in $\mathbf{Z}_{test}$ and $\mathbf{X}^{'(d)}_{test}$ refers to the testing dataset with column indices $\mathbf{I}_d$, i.e.,, only the columns that are selected as important are used to estimate $\widetilde{\mathbf{Z}'}_{test}$. The output layer $\widetilde{R}_d = \sigma(h^{(d)}_{(K_d-1)}\widetilde{\mathbf{W}}^{'(d)}_{K_d} + \widetilde{\mathbf{b}}^{'(d)}_{K_d}) \in \mathbb{R}^{n \times p^{'(d)}}$, the subscript K_d denotes the Kth hidden layer for view d, $h^{(d)}_{(K_d-1)} \in \mathbb{R}^{n \times c_d}$, $\widetilde{\mathbf{W}}^{'(d)}_{K_d} \in \mathbb{R}^{c_d \times p^{'(d)}}$, and $\mathbf{b}^{'(d)}_{K_d} \in \mathbb{R}^{n \times p^{(d)}}$, and $h_1 = \sigma(\mathbf{Z}'_{test}\widetilde{\mathbf{W}}^{'(d)}_1 + \widetilde{\mathbf{b}}^{'(d)}_1)$.

Now, when predicting an outcome, the low-dimensional representations $\widetilde{\mathbf{Z}'}$ and $\widetilde{\mathbf{Z}'}_{test}$ become training and testing data, respectively. For example, to predict a binary or multiclass outcome, we train a support vector machine (SVM) [18] classifier with the training data $\widetilde{\mathbf{Z}'}$ and the outcome data $\mathbf{y}$, and we use the learned SVM model and the testing data $\widetilde{\mathbf{Z}'}_{test}$ to obtain the predicted class membership, $\widehat{\mathbf{y}}_{test}$. We compare $\widehat{\mathbf{y}}_{test}$ with $\mathbf{y}_{test}$ and we estimate the classification accuracy. For continuous outcome, one can implement a nonlinear regression model and then compare the predicted and true outcomes using a metric such as the mean squared error (MSE). For unsupervised analyses, such as clustering, an existing clustering algorithm, such as K-means clustering, can be trained on $\widetilde{\mathbf{Z}'}$. Figure 2 is a schematic representation of the prediction algorithm and the downstream analyses proposed.

We provide our optimization approach in the Additional file 1. Our algorithm is divided into three stages. The first stage is the Feature Selection stage. In this stage, we solve the optimization problem (1) or (3) to obtain features that are highly ranked. The second stage is the Reconstruction and Training stage using selected features. Here, we solve the optimization problem (4). Our input data are the observed data with the selected features (that is, the top r or r% features in each view), $\mathbf{X}^{'(1)} \ldots \mathbf{X}^{'(D)}$. At convergence, we obtain the reconstructed data $R_d(\mathbf{Z}')$, and the learned shared low-dimensional representations $\widetilde{\mathbf{Z}'}$ based only on the top r or r% variables in each view. Downstream

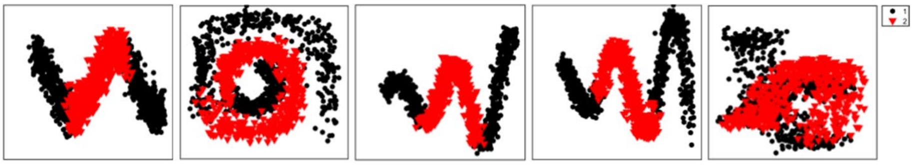

Fig. 3 Structure of nonlinear relationships between (First left panel) signal variables in View 1; (Second left panel) signal variables in View 2; (Middle panel)-(Fifth panel) signal variables between Views 1 and 2. Black circle: Class 1; Red triangle: Class 2

analyses such as classification, regression, or clustering could be carried out on these shared low-dimensional representations learned. The third stage is the prediction stage, if an outcome is available. Here, we solve the optimization problem (6) for the learned shared low-dimensional representation ($\widetilde{\mathbf{Z}}'_{test}$) corresponding to the test views ($\mathbf{X}'^{(1)}_{test} \ldots \mathbf{X}'^{(D)}_{test}$). This can be used to obtain prediction estimates (e.g. testing classification via an SVM model).

Simulation experiments

We conducted simulation studies to assess the performance of iDeepViewLearn for varying data dimensions, as the relationship between views becomes more complex and when prior information on variable–variable relationships is available or not. Please refer to the Additional file 1 for more simulation setup and results.

Set-up when there is no prior information on variable–variable interactions

We consider two different simulation scenarios to demonstrate both the variable selection and classification performance of the proposed method. In the first scenario, we simulate data with linear relationships among the views and within a view (see Additional file 1). In the second scenario, we simulate the data to show nonlinear relationships. In each scenario, there are $D = 2$ views and within each view there are two distinct classes. In all scenarios, we generate 20 Monte Carlo training, tuning, and testing sets. We train the models on the training set, choose optimal hyper parameters using the tuning set, and obtain classification performance using the testing set. We evaluate the proposed and existing methods using the following criteria: i) test accuracy, and ii) feature selection. For feature selection, we evaluate the methods ability to select the true signals and ignore noise variables. We use true positive rates (TPR), false positive rates (FPR), and F-measure as metrics to evaluate the variable selection performance. In Scenario 1, the first 20 variables are important, and in Scenario Two, the top 10% of the variables in both views are signals.

Nonlinear simulations

We consider three different settings for this scenario. Each setting has $K = 2$ classes, but they vary in dimension. In each setting, 10% of the variables in each view are signals and the first five signal variables in each view are related to the remaining signal variables in a nonlinear way (see Fig. 3). We generate data for View 1 as follows: $\mathbf{X}^{(1)} = \widetilde{\mathbf{X}}_1 \cdot \mathbf{W} + 0.2\mathbf{E}_1$ where $(\cdot)$ is element-wise multiplication, $\mathbf{W} \in \mathbb{R}^{n\times p^{(1)}} = [\mathbf{1}_{0.1\times p^{(1)}}, \mathbf{0}_{0.9\times p^{(1)}}]$ is a matrix of ones and zeros, $\mathbf{1}$ is a matrix of ones, $\mathbf{0}$ is a matrix of zeros, and $\mathbf{E}_1 \sim N(0, 1)$. Each of the first five signal variables in $\widetilde{\mathbf{X}}_1 \in \mathbb{R}^{n \times p^{(1)}}$ is obtained from $\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}} + 0.5U(0,1)$, where $\tilde{\boldsymbol{\theta}}$ is a vector of n evenly spaced points between 0 and 3π. The next 45 signal variables (or columns) in $\widetilde{\mathbf{X}}_1 \in \mathbb{R}^{n \times p^{(1)}}$ are simulated from $\cos(\boldsymbol{\theta})$ plus random noise from a standard normal distribution. The remaining $0.9p^{(1)}$ variables (or columns) in $\widetilde{\mathbf{X}}_1$ are generated from the standard normal distribution. We generate data for View 2 as: $\mathbf{X}^{(2)} = \widetilde{\mathbf{X}}_2 \cdot \mathbf{W} + 0.2\mathbf{E}_2$ where $\mathbf{E}_2 \sim N(0,1)$. The first five columns (or variables) of $\widetilde{\mathbf{X}}_2 \in \mathbb{R}^{n \times p^{(2)}}$ are simulated from $\exp(0.15\boldsymbol{\theta}) \cdot \sin(1.5\boldsymbol{\theta})$. The next $0.1p^{(2)} - 5$ variables are simulated from $\exp(0.15\boldsymbol{\theta}) \cdot \cos(1.5\boldsymbol{\theta})$. The remaining $0.9p^{(2)}$ variables (or columns) in $\widetilde{\mathbf{X}}_2$ are generated from the standard normal distribution. The class labels $\mathbf{y} = [\mathbf{1}_{n_1/2}\ 2 \cdot \mathbf{1}_{n_2}\ \mathbf{1}_{n_1/2}]$ where $(n_1, n_2) = (200, 150)$ or $(6000, 4500)$. Figure 3 shows the structure of the nonlinear relationships between signal variables in View 1(First left panel), signal variables in View 2(Second left panel), and signal varibles between Views 1 and View 2 (Middle to Last panel), with black circles denoting data from Class 1 and red triangles data from Class 2.

Competing Methods and Results

We compare the proposed method, iDeepViewLearn, with linear and nonlinear methods for associating data from multiple views. For linear methods, we consider the sparse canonical correlation analysis [Sparse CCA] proposed in [2]. For the nonlinear methods, we compare with deep canonical correlation analysis (Deep CCA) [5] and MOMA [8]. We note that MOMA is a joint integration and classification method and as such does not require further training a classification method such as SVM, after training MOMA. However, per reviewer comment, we add a comparison where we use the important features chosen by MOMA to train and test an SVM classifier; we call this MOMA + SVM. For Sparse CCA and Deep CCA, we use the estimated canonical variates in SVM for classification performance since these two methods are unsupervised. We also compare the proposed method that integrates the two views with our method on stacked data, and SVM and random forest [19] on stacked data as well, to explore the benefits of multiview learning. Of note, by stacking the data, we do not appropriately model the dependency structure among views as one assumes that the views are not correlated, contrary to the assumption for data integration. We perform Sparse CCA with the *SelpCCA* R package provided by the authors on GitHub. We performed Deep CCA and MOMA using PyTorch codes provided by the authors. We pair Deep CCA with the teacher-student framework (TS) [20] to rank variables, and compare the TS feature selection approach with the proposed method. We follow the variable-ranking approach in MOMA to rank variables. We report the classification and variable selection results in Table 1 for nonlinear simulations (see results of linear settings and the network structures in Additional file 1). We implemented the proposed method in the training data, selected the top 10% variables for each view, learned a new model with these selected variables, and obtained test errors with the test data. The misclassification rates for the proposed method were lower or competitive compared to all the competitors. We observed a decreasing misclassification rate with increasing sample sizes for all the methods; nevertheless, the proposed method produced lower or competitive test errors even when the sample size was smaller than the dimension of the variables. In terms of variable selection, the TS framework applied to Deep CCA yielded suboptimal results; MOMA and random forest rank the important features based on their influence

Table 1 Nonlinear settings: randomly select combinations of hyper-parameters to search over

Method	Error (%)	TPR-1	TPR-2	FPR-1	FPR-2	F-1	F-2
Setting 1							
($p_1 = 500, p_2 = 500, n_1 = 200, n_2 = 150$)							
iDeepViewLearn	1.89 (0.47)	100.00	100.00	0.00	0.00	100.00	100.00
iDeepViewLearn on stacked data	4.00 (0.47)	100.00	100.00	0.00	0.00	100.00	100.00
Sparse CCA + SVM	6.10 (0.73)	100.00	90.00	0.11	0.01	99.51	94.69
Deep CCA + TS + SVM	35.61 (2.22)	11.10	11.30	9.88	9.86	11.10	11.30
MOMA	44.96 (1.70)	22.00	29.90	8.67	7.89	22.00	29.90
MOMA + SVM	30.47 (6.05)	22.00	29.90	8.67	7.89	22.00	29.90
Random Forest on stacked data	1.94 (0.60)	70.10	98.00	3.32	0.22	70.10	98.00
SVM on stacked data	28.07 (0.65)	–	–	–	–	–	–
Setting 2							
($p_1 = 500, p_2 = 500, n_1 = 6000, n_2 = 4500$)							
iDeepViewLearn	1.26 (0.11)	100.00	100.00	0.00	0.00	100.00	100.00
iDeepViewLearn on stacked data	1.38 (0.08)	100.00	100.00	0.00	0.00	100.00	100.00
Sparse CCA + SVM	4.25 (0.15)	100.00	90.00	0.00	0.00	100.00	94.74
Deep CCA + TS + SVM	0.66 (0.13)	30.40	21.60	7.73	8.71	30.40	21.60
MOMA	12.77 (8.63)	76.30	89.90	2.63	1.12	76.30	89.90
MOMA + SVM	0.63 (0.08)	76.30	89.90	2.63	1.12	76.30	89.90
Random Forest on stacked data	0.66 (0.05)	100.00	100.00	0.00	0.00	100.00	100.00
SVM on stacked data	2.31 (0.15)	–	–	–	–	–	–
Setting 3							
($p_1 = 2,000, p_2 = 2,000, n_1 = 200, n_2 = 150$)							
iDeepViewLearn	2.56 (0.78)	99.98	99.88	0.00	0.00	99.98	99.88
iDeepViewLearn on stacked data	2.86 (0.73)	99.98	99.65	0.01	0.04	99.98	99.65
Sparse CCA + SVM	4.86 (0.88)	100.00	97.50	0.08	0.02	99.63	98.66
Deep CCA + TS + SVM	29.91 (1.27)	10.30	11.20	9.97	9.87	10.30	11.20
MOMA	46.14 (2.44)	16.40	13.68	9.29	9.59	16.40	13.68
MOMA + SVM	35.46 (5.91)	16.40	13.68	9.29	9.59	16.40	13.68
Random Forest on stacked data	5.40 (1.02)	58.67	89.88	4.59	1.13	58.67	89.88
SVM on stacked data	28.57 (0.53)	–	–	–	–	–	–

TPR-1; true positive rate for $\mathbf{X}^{(1)}$. Similar for TPR-2. FPR-1; false positive rate for $\mathbf{X}^{(1)}$. Similar for FPR-2; F-1 is the F measure for $\mathbf{X}^{(1)}$. Similar for F-2. The highest F-1/2 is in. (The mean error of two views is reported for MOMA; MOMA + SVM means selecting features using MOMA and training an SVM on the selected features)

on the classification performance, and the two methods usually select unimportant features when the sample size is small; iDeepViewLearn and Sparse CCA can always achieve nearly perfect performance for feature selection in the nonlinear simulations. The performance of iDeepViewLearn on the stacked data was similar, although it had slightly higher classification errors, when compared to iDeepViewLearn that holistically integrates the views; thus we recommended against stacking data and implementing the proposed method, but rather using the method that integrates the two views as we have proposed. The results of the linear simulations mimic those of the nonlinear simulations.

Set-up when there is prior information on variable–variable interactions

Here, $\mathbf{X}^{(1)} = \widetilde{\mathbf{X}}_1 \cdot \mathbf{W} + \mathbf{E}_1$ and $\mathbf{X}^{(2)} = \widetilde{\mathbf{X}}_2 \cdot \mathbf{W} + 0.2\mathbf{E}_2$. $\widetilde{\mathbf{X}}_i$, $i = 1, 2$ is defined as before. However, $\mathbf{E}_i \sim N(0, \mathbf{\Sigma}_i)$, $i = 1, 2$, $\mathbf{\Sigma}_i$ is a diagonal block matrix with two blocks that represent signal and noise variables. The first block is a 50×50 covariance matrix that

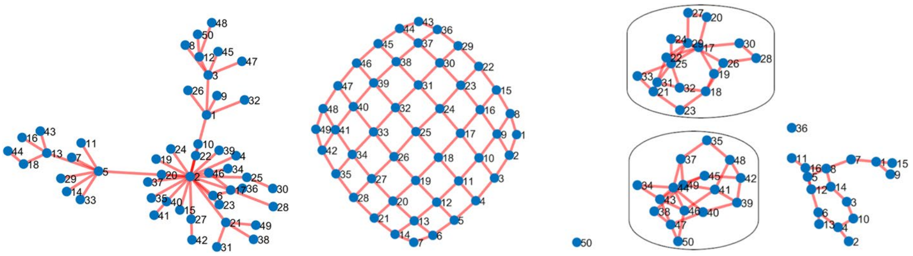

Fig. 4 Network structure for the first 50 variables in $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$. Left: scale-free network; Middle: Lattice; Right: Cluster. For the Scale-free network, we consider variable 2 has a hub variable. Variable 2 and the variables directly connected to it are considered as signal variables. For the Lattice network, all variables except variable 50 are considered as signals. For the Cluster network, the circled clusters are considered as signals

captures the relationship among these 50 variables. Let G be the true graph structure for these variables. The second block is the identity matrix. We use *bdgraph.sim* in the *BDGraph* R package [21] to simulate three different types of networks for the first 50 variables: Scale-free, Lattice, and Cluster, and to obtain the adjacency matrix corresponding to the graph structure G. We then use *rgwish* from the same R package to generate a precision matrix distributed according to the G—Wishart distribution, $W_G(b, D)$, with parameters $b = 3$ and $D = \mathbf{I}$ with respect to the graph structure G. We obtain the covariance matrix from the precision matrix. Figure 4 shows the variable–variable relationships among these 50 variables for the different network structure. Figure 4 [left panel], variable two is connected to more variables, so we consider variable 2 as a hub variable. We set $\mathbf{W} = [\mathbf{1}_{\mathcal{H}}, \mathbf{0}_{p^{(1)}-\mathcal{H}}]$, $\mathcal{H}$ to denote the variables directly connected to variable 2, and $p^{(1)} - \mathcal{H}$ (similarly $p^{(2)} - \mathcal{H}$) denote the remaining variables. By defining $\mathbf{W}$ this way, we assume that only the variables directly connected to the hub variable are signals and contribute to the nonlinear relationship between the two views. For the Lattice network (Middle panel), all variables in the network except variable 50 contribute to the nonlinear relationships among the views. For the cluster network, only two clusters (circled) are signals.

Competing methods and results

We explore the proposed method with and without the use of network information. In addition to competitors in the nonlinear simulations, we further compare the proposed method with Fused CCA [11]. Fused CCA is a sparse canonical correlation analysis method that uses variable–variable information to guide the estimation of the canonical variates and the selection of variables that contribute most to the association between two views. We implemented Fused CCA using the R code accompanying the manuscript. We followed Fused CCA with SVM for classification. We implemented the proposed method on the training data with and without the network information, selected the top ranked variables (21 variables for Scale-free network, 49 variables for Lattice network, and 33 for Cluster network) for each view, learned a new model with these selected variables, and obtained test errors with the testing data. We also compared the top-ranked variables with the true signal variables and the estimated true positive rates (TPR), false positive rates (FPR), and F-score. From Table 2, the classification performance of our method that does not incorporate prior knowledge is comparable to our method that

Table 2 Simulation with variable–variable connections: randomly select combinations of hyper-parameters to search over

Method	Error (%)	TPR-1	TPR-2	FPR-1	FPR-2	F-1	F-2
Scale-free							
Setting 1							
($p_1 = 500, p_2 = 500, n_1 = 200, n_2 = 150$)							
iDeepViewLearn	6.84 (1.96)	95.71	96.19	0.19	0.17	95.71	96.19
iDeepViewLearn-Laplacian	7.10 (1.65)	97.86	98.57	0.09	0.06	97.86	98.57
Sparse CCA + SVM	21.80 (10.14)	100.00	100.00	14.12	15.13	42.97	43.26
Fused CCA + SVM	41.33 (7.52)	19.29	23.10	17.90	33.29	6.56	4.08
Deep CCA + TS + SVM	40.43 (1.61)	5.00	3.33	4.16	4.24	5.00	3.33
MOMA	45.55 (1.84)	17.14	25.95	3.63	3.25	17.14	25.95
MOMA + SVM	36.76 (5.41)	17.14	25.95	3.63	3.25	17.14	25.95
Random Forest on stacked data	11.99 (2.22)	52.38	85.23	2.09	0.65	52.38	85.23
SVM on stacked data	35.46 (1.23)	–	–	–	–	–	–
Setting 2							
($p_1 = 500, p_2 = 500, n_1 = 6000, n_2 = 4500$)							
iDeepViewLearn	2.77 (0.62)	99.76	100.00	0.01	0.00	99.76	100.00
iDeepViewLearn-Laplacian	2.71 (0.14)	100.00	100.00	0.00	0.00	100.00	100.00
Sparse CCA + SVM	9.25 (2.39)	100.00	100.00	9.35	9.22	56.74	52.59
Fused CCA + SVM	33.24 (4.83)	99.75	100.00	13.69	48.83	48.74	19.90
Deep CCA + TS + SVM	2.44 (0.31)	17.62	17.38	3.61	3.62	17.62	17.38
MOMA	41.88 (4.11)	63.81	72.62	1.59	1.20	63.81	72.62
MOMA + SVM	3.56 (4.37)	63.81	72.62	1.59	1.20	63.81	72.62
Random Forest on stacked data	1.86 (0.10)	100.00	100.00	0.00	0.00	100.00	100.00
SVM on stacked data	27.61 (0.23)	–	–	–	–	–	–
Lattice							
Setting 1							
($p_1 = 500, p_2 = 500, n_1 = 200, n_2 = 150$)							
iDeepViewLearn	4.90 (1.76)	100.00	98.88	0.00	0.12	100.00	98.88
iDeepViewLearn-Laplacian	3.90 (0.82)	99.80	99.59	0.02	0.04	99.80	99.59
Sparse CCA + SVM	16.03 (0.86)	100.00	100.00	1.29	1.45	94.96	94.77
Fused CCA + SVM	38.26 (11.58)	22.04	24.69	24.59	30.99	11.13	9.30
Deep CCA + TS + SVM	36.53 (2.05)	10.61	10.71	9.71	9.70	10.61	10.71
MOMA	44.76 (2.12)	23.98	26.53	8.26	7.98	23.98	26.53
MOMA + SVM	32.20 (4.27)	23.98	26.53	8.26	7.98	23.98	26.53
Random Forest on stacked data	3.41 (0.71)	66.84	92.86	3.60	0.78	66.84	92.86
SVM on stacked data	28.51 (0.56)	–	–	–	–	–	–
Setting 2							
($p_1 = 500, p_2 = 500, n_1 = 6000, n_2 = 4500$)							
iDeepViewLearn	1.64 (0.17)	100.00	100.00	0.00	0.00	100.00	100.00
iDeepViewLearn-Laplacian	1.56 (0.12)	100.00	100.00	0.00	0.00	100.00	100.00
Sparse CCA + SVM	7.14 (2.92)	100.00	100.00	1.14	1.72	95.52	93.52
Fused CCA + SVM	5.26 (2.27)	100.00	100.00	3.44	6.84	88.89	78.95
Deep CCA + TS + SVM	0.98 (0.19)	39.49	32.04	6.57	7.38	39.49	32.04
MOMA	21.22 (13.01)	73.98	79.08	2.83	2.27	73.98	79.08
MOMA + SVM	1.27 (1.53)	73.98	79.08	2.83	2.27	73.98	79.08
Random Forest on stacked data	1.02 (0.07)	100.00	100.00	0.00	0.00	100.00	100.00
SVM on stacked data	8.57 (0.24)	–	–	–	–	–	–
Cluster							

Table 2 (continued)

Method	Error (%)	TPR-1	TPR-2	FPR-1	FPR-2	F-1	F-2
Setting 1							
($p_1 = 500, p_2 = 500, n_1 = 200, n_2 = 150$)							
iDeepViewLearn	22.50 (1.73)	96.21	100.00	0.27	0.00	96.21	100.00
iDeepViewLearn-Laplacian	22.40 (2.14)	95.15	100.00	0.34	0.00	95.15	100.00
Sparse CCA + SVM	16.70 (1.22)	100.00	100.00	4.24	3.91	77.50	78.54
Fused CCA + SVM	43.27 (1.65)	16.97	16.06	18.96	22.76	5.52	5.56
Deep CCA + TS + SVM	37.96 (1.81)	7.12	6.97	6.56	6.57	7.12	6.97
MOMA	45.28 (2.02)	21.52	23.18	5.55	5.43	21.52	23.18
MOMA + SVM	36.61 (3.77)	21.52	23.18	5.55	5.43	21.52	23.18
Random Forest on stacked data	29.23 (1.19)	27.42	65.76	5.13	2.42	27.42	65.76
SVM on stacked data	31.60 (1.02)	–	–	–	–	–	–
Setting 2							
($p_1 = 500, p_2 = 500, n_1 = 6000, n_2 = 4500$)							
iDeepViewLearn	15.78 (0.65)	100.00	99.39	0.00	0.04	100.00	99.39
iDeepViewLearn-Laplacian	15.70 (0.35)	96.21	100.00	0.27	0.00	96.21	100.00
Sparse CCA + SVM	14.59 (0.53)	100.00	100.00	12.07	7.54	57.31	66.46
Fused CCA + SVM	29.17 (9.55)	72.73	92.42	26.12	30.40	31.17	41.73
Deep CCA + TS + SVM	28.48 (1.52)	10.45	8.64	6.33	6.46	10.45	8.64
MOMA	39.22 (4.95)	73.18	91.82	1.90	0.58	73.18	91.82
MOMA + SVM	12.77 (0.72)	73.18	91.82	1.90	0.58	73.18	91.82
Random Forest on stacked data	13.83 (0.21)	100.00	100.00	0.00	0.00	100.00	100.00
SVM on stacked data	29.68 (0.22)	–	–	–	–	–	–

TPR-1; true positive rate for $\mathbf{X}^{(1)}$. Similar for TPR-2. FPR; false positive rate for $\mathbf{X}^{(2)}$. Similar for FPR-2; F-1 is the F measure for $\mathbf{X}^{(1)}$. Similar for F-2. The highest F-1/2 is in red. (The mean error of two views is reported for MOMA; MOMA + SVM means combining the feature selection part of MOMA and SVM)

does, in all settings. The fused CCA result for Scale-free Setting 2 was based on 19 out of the 20 simulation replicates due to a computational error. The fused CCA result for Lattice Setting 2 was not available due to time constraints. Like the scenario with no prior information, the misclassification rates for the proposed method (with or without prior information) were lower or competitive, especially for the Scale-free and Lattice networks, when compared to the association-based methods. Furthermore, the proposed method was superior to MOMA and SVM on stacked views across all settings and network type, for both classification and variable selection accuracy. For random forest, it can achieve very comparable prediction and feature selection performance with our methods when there are sufficient training data points; however, our iDeepViewLearn's performance outperforms random forest when the sample size is limited when considering both classification and variable selection performance.

In summary, the classification and variable selection accuracy from both the linear and nonlinear simulations, and when we use or do not use prior information, suggest that the proposed methods are capable of ranking signal variables as high and ignoring noise variables. The proposed methods are also capable of producing competitive or better classification performance among all settings. In particular, we notice that random forest can achieve comparable classification and variable selection accuracies with our iDeepViewLearn when the number of training samples is relatively large, but the feature selection performance of random forest is usually suboptimal in situations where the sample

Table 3 Summary of datasets for each analysis

Dataset	Categories	Number of features in each view	Sample Size	Task
Holm Breast Cancer	Died: 65	View 1, gene expression, 469	Training $n = 112$	Classification and
Study	Survived: 103	View 2, methylation, 334	Testing $n = 56$	Clustering
LGG Dataset	Grade 2: 246	View 1, methylation, 9691	Training $n = 410$	Classification
	Grade 3: 264	View 2, miRNA, 235	Testing $n = 100$	
		View 3, mRNAseq, 7603		
Shear Transformed	Hand-written digits 0 to 9	View 1, digits, 784	Training $n = 60000$	Classification and
MNIST Dataset	count ranging from 5400 to 6800	View 2, digits, 784	Testing $n = 10000$	Reconstruction

size is less than the number of variables, as shown in Tables 1 and 2. These findings are encouraging to us since in a typical setting of high-dimensional and biomedical problems, the sample size is smaller than the number of variables.

Real-world experiments

In this section, we consider three real-world applications to show the effectiveness of the proposed method across different tasks and settings. We first applied the proposed method to integrate gene expression and methylation data from the Holm breast cancer study [22] for classification and clustering tasks with two views. We next applied the proposed method to data pertaining to brain lower grade glioma (LGG) to demonstrate the use of our method for classification tasks with three views. Finally, we applied our method on a MNIST handwriting data, for a reconstruction task, demonstrating that handwriting digits can be reconstructed with few pixels while maintaining competitive classification accuracy. The details of all the datasets used in this section are shown in Table 3.

Evaluation of data from holm breast cancer study

Breast cancer is the most common cancer among women worldwide, accounting for 12.5% of new cases and is one of the leading causes of death in women [23]. Research shows that breast cancer is a multi-step process that involves both genetic and epigenetic changes. Epigenetic factors such as DNA methylation and histone modification lead to breast tumorigenesis by silencing critical tumor suppressor and growth regulator genes [24]. Identifying methylated sites correlated with gene expression data could shed light on the genomic architecture of breast cancer. Our work is motivated by a molecular subtyping study conducted in [22], which used gene expression and DNA methylation data to identify methylation patterns in breast cancer. For completeness, we describe the data here. Raw methylation profiles from 189 breast cancer samples were extracted using the Beadstudio Methylation Module (Illumina). There were 1452 CpG sites (corresponding to 803 cancer-related genes). β-values were stratified into three groups: 0, 0.5, and 1. The value of 1 corresponded to hypermethylation. Relative methylation levels were obtained from raw values by centering the stratified β values in all samples. Furthermore, relative gene expression levels of 179 of 189 breast cancer tumors were obtained using oligonucleotide arrays for 511 probes. The number of samples with data on gene expression and methylation for our analysis is $n = 179$. The first view, corresponding to gene expression data, had 468 variables (genes), and the second view, corresponding to methylation data, had 1452 variables (CpG sites). The methylation data were filtered to include the most variable methylated sites by removing CpG sites with a standard deviation less than 0.3 between samples; this resulted in 334 CpG sites (corresponding to 249 cancer-related genes). In addition to molecular data, data on whether an individual died from breast cancer or not is also available.

The goal of our analysis is to perform an integrative analysis of the methylation and gene expression data to model nonlinear associations between CpG sites and genes through a joint non-linear embedding that drives the overall dependency structure in the data. Importantly, we wish to identify a subset of CpG sites and genes that contribute to the dependency structure and could be used to discriminate between those who survived and those who did not survive breast cancer. Further, we wish to explore the use of the joint nonlinear embedding in molecular clustering.

Goal 1: Model nonlinear relationships between methylation and gene expression data and identify CpG sites and genes that can discriminate between those who died and those who did not die from breast cancer

For the first goal, we split the data into three sets of approximately equal size. We used 2/3rd of the data to train the model and we used the remaining 1/3rd to test our models. We implemented the proposed method on the training set, selected the top 10% and 20% highly ranked variables in each view, learned new models with these selected features, used the test data and the models learned to predict the test outcomes, and obtained test errors. We repeated the process 20 times, obtained the highly ranked variables for each run, and estimated the average test errors. We compared the proposed method with SVM, random forest, Deep CCA, Sparse CCA, MOMA and MOMA + SVM based on average test errors.

Average misclassification rates and genes and CpG sites selected Table 4 gives the average test errors for the methods. On the basis of the high classification errors across the methods, it seems that separating those who died from breast cancer from those who did not die using methylation and gene expression data is a difficult problem. We investigate the use of iDeepViewLearn on the stacked data, and we notice that, similar to the nonlinear simulations, there are small gaps present compared to the results when we integrate the data. Hence, we still recommend using our method as proposed when there are two or more views, and not apply on stacked data. The average test error for the proposed method based on the top 10% or 20% CpG sites and genes is comparable to that of the other methods. Our proposed method allows us to obtain insight into the genes and CpG sites that drive the classification accuracy.

For this purpose, we explored the "stable" genes and CpG sites that potentially discriminate between people who died and those who did not die from breast cancer. We consider a variable to be "stable" if it is ranked in the top 20% at least 16 times ($\geq$ 80%) of the 20 resampled datasets. Table 5 shows the genes selected and how often they were selected. Genes DAB2, DCN, HLAF, MFAP4, MMP2, PDGFRB, TCF4 and TMEFF1

Table 4 Breast cancer data: SVM is based on stacked raw data with two views

Method	Average Error (Std.Dev) (%)
SVM	39.02 (4.77)
Deep CCA + SVM	38.57 (5.40)
Sparse CCA + SVM	40.94 (4.24)
MOMA	44.51 (3.90)
MOMA + SVM	39.46 (5.67)
Random Forest	40.36 (5.28)
iDeepViewLearn with selected top 10% features	39.02 (5.03)
iDeepViewLearn with selected top 20% features	39.02 (5.03)
iDeepViewLearn with selected top 10% stacked features	39.11 (4.82)
iDeepViewLearn with selected top 20% stacked features	39.38 (5.55)

Deep CCA + SVM is a training SVM based on the last layer of Deep CCA. iDeepViewLearn with selected top 10% features reconstructs the original views with only 10% of the features and obtains a test classification error based on a shared low-dimensional representation trained on data with only 10% of the features. Similar to iDeepViewLearn with selected top 20%. (The mean error of two views is reported for MOMA; MOMA + SVM means combining the feature selection part of MOMA and SVM)

Table 5 Frequency of Genes selected at least 16 times in the top 20% across 20 resampled datasets

Gene	Gene Name	Frequency
DAB2	DAB adaptor protein 2	20
DCN	decorin	20
HLAF	major histocompatibility complex, class I, F	20
MFAP4	microfibril associated protein 4	20
MMP2	matrix metallopeptidase 2	20
PDGFRB	platelet derived growth factor receptor beta	20
TCF4	transcription factor 4	20
TMEFF1	transmembrane protein with EGF like and two follistatin like domains 1	20
AFF3	AF4/FMR2 family member 3	19
BIRC5	baculoviral IAP repeat containing 5	19
CDH11	cadherin 11	19
COL1A2	collagen type I alpha 2 chain	19
LYN	LYN proto-oncogene, Src family tyrosine kinase	19
SPARC	secreted protein acidic and cysteine rich	19
THBS2	thrombospondin 2	19
BGN	biglycan	18
COL6A1	collagen type VI alpha 1 chain	18
CSPG2	versican	18
LOX	lysyl oxidase	18
SLIT2	slit guidance ligand 2	18
TIMP2	TIMP metallopeptidase inhibitor 2	18
EPHB3	EPH receptor B3	17
HLADPA1	Major Histocompatibility Complex, Class II, DP Alpha 1	17
IGFBP7	insulin like growth factor binding protein 7	17
SPDEF	SAM pointed domain containing ETS transcription factor	17
THY1	Thy-1 cell surface antigen	17
TNFRSF1B	TNF receptor superfamily member 1B	17
IL16	interleukin 16	16

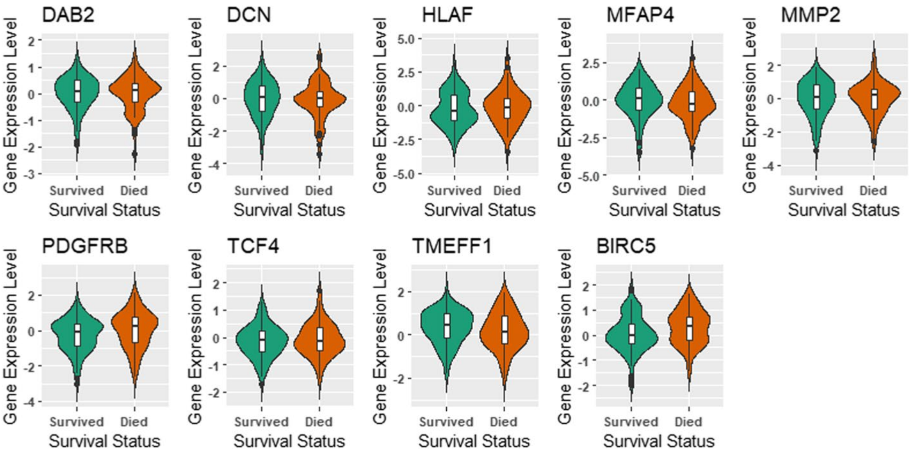

Fig. 5 All genes except BIRC5 were consistently selected in the top 20% of highly-ranked genes across the twenty resampled datasets. BIRC5 was selected 19 times (out of 20) in the top 20% highly-ranked genes. Genes PDGFRB and BIRC5 have mean expression levels that are statistically significantly different between individuals that died from breast cancer and those that survived

were consistently selected in the top 20% in all resampled data sets. There is support in the literature for the potential role of some of these genes in a variety of human cancers. Disabled homolog 2 [or DAB adaptor protein 2] (DAB2) is a protein-coding gene that is often deleted or silenced in several human cancer cells. The decorin gene (DCN) is a protein coding gene that encodes the protein decorin. Research on different human cancers (e.g. breast, prostate, bladder) has shown that DCN expression levels in cancerous cells are significantly reduced from expression levels in normal tissues or are often completely silenced in tumor tissues [25]. Research suggests that individuals expressing lower levels of DCN in cancer tend to have poorer outcomes compared to individuals expressing higher levels of DCN. In our data, the mean expression levels of DCN for those who survived were not statistically different (based on the Anova test) from those who did not.

We observed statistically significant differences in mean expression levels of PDGFR and BIRC5 for the two groups (p-value $<$ 0.05 from ANOVA test), as shown in Fig. 5. The median expression values of these genes were higher in individuals who died of breast cancer. The platelet-derived growth factor receptor alpha (PDGFRA) gene is a protein encoding gene that encodes the PDGFRA protein. The PDGFRA protein is involved in important biological processes such as cell growth, division, and survival. Mutated forms of the PDGFRA gene and protein have been found in some types of cancer. The BIRC5 gene is a protein encoder gene that encodes the baculoviral IAP repeat containing protein 5 in humans. This protein is believed to play an important role in the promotion of cell division (proliferation) and in the prevention of cell apoptosis (death) [26, 27].

Table 6 shows the CpG sites selected at least 16 times in the top 20% of the highly ranked CpG sites. The CpG sites FGF2_P229_F, IL1RN_E42_F, RARA_P1076_R, TFF1_P180_R, TGFB3_E58_R, WNT2_P217_F were consistently selected in the top 20% of highly ranked CpG sites across all resampled datasets. Figure 6 shows the relative methylation levels of the CpG sites that were consistently selected or were

Table 6 Frequency of Genes selected at least 16 times in the top 20% across 20 resampled datasets

CpG Site	Corresponding Gene	Gene Name	Frequency
FGF2_P229_F	FGF2	Fibroblast growth factor 2	20
IL1RN_E42_F	IL1RN	Interleukin 1 receptor antagonist	20
RARA_P1076_R	RARA	Retinoic acid receptor alpha	20
TFF1_P180_R	TFF1	Trefoil factor 1)	20
TGFB3_E58_R	TGFB3	Transforming growth factor beta 3	20
WNT2_P217_F	WNT2	Wnt family member 2	20
ADAMTS12_E52_R	ADAMTS12	ADAM metallopeptidase with thrombospondin type 1 motif 12	19
RASSF1_P244_F	RASSF1	Ras association domain family member 1	18
FABP3_E113_F	FABP3	Fatty acid binding protein 3	16
IGFBP7_P297_F	IGFBP7	Insulin like growth factor binding protein 7	16
IL1RN_P93_R	IL1RN	Interleukin 1 receptor antagonist	16
RASSF1_E116_F	RASSF1	Ras association domain family member 1	16
SLC22A3_E122_R	SLC22A3	Solute carrier family 22 member 3	16
TPEF_seq_44_S88_R	TPEF	Transmembrane Protein With EGF Like And Two Follistatin Like Domains 2	16

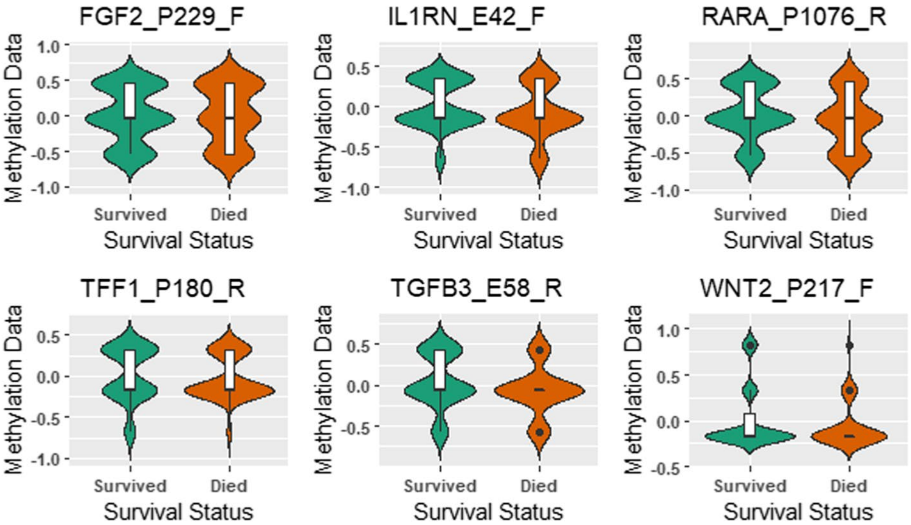

Fig. 6 All CpG sites were consistently selected in the top 20% of highly-ranked CpG sites across the twenty resampled datasets. The mean methylation levels of IL1RN_E42_F and TGFB3_E58_R are statistically different between individuals that died from breast cancer and those that survived

significantly different between those who survived and those who died. The mean methylation levels for the CpG sites IL1RN_E42_F and TGFB3_E58_R were statistically different between those who died and those who survived breast cancer (p-value < 0.05 from Anova test). In particular, the mean relative methylation levels for IL1RN_E42_F and TGFB3_E58_R were lower in those who died from breast cancer compared to those who did not. The interleukin 1 receptor antagonist (IL1RN) gene is a protein-coding gene that encodes the interleukin-1 receptor antagonist protein, a member of the interleukin 1 cytokine family. IL1RN is an anti-inflammatory molecule

Table 7 Top 10 Gene Ontology (GO) Biological Processes enriched with ToppFun in ToppGene Suite

GO ID	GO Biological Process	Bonferroni *P*-value	Genes
GO:0001944	Vasculature development	2.71E-10	DAB2,EPHB3,SLIT2,BIRC5,TCF4,THBS2,SPARC,MMP2 TNFRSF1B,THY1,DCN,IGFBP7,PDGFRB,LOX,HLA-F,COL1A2
GO:0001568	Blood vessel development	2.10E-09	DAB2,EPHB3,SLIT2,BIRC5,TCF4,THBS2,SPARC,MMP2 THY1,DCN,IGFBP7,PDGFRB,LOX,HLA-F,COL1A2
GO:0048514	Blood vessel morphogenesis	9.21E-09	DAB2,EPHB3,SLIT2,BIRC5,TCF4,THBS2,SPARC,MMP2 THY1,DCN,IGFBP7,PDGFRB,LOX,HLA-F
GO:0030198	Extracellular matrix organization	1.40E-08	COL6A1,MFAP4,SPARC,MMP2,TNFRSF1B DCN,TIMP2,LOX,VCAN,BGN,COL1A2
GO:0043062	Extracellular structure organization	1.43E-08	COL6A1,MFAP4,SPARC,MMP2,TNFRSF1B DCN,TIMP2,LOX,VCAN,BGN,COL1A2
GO:0045229	External encapsulating structure organization	1.53E-08	COL6A1,MFAP4,SPARC,MMP2,TNFRSF1B DCN,TIMP2,LOX,VCAN,BGN,COL1A2
GO:0072359	Circulatory system development	1.83E-08	DAB2,EPHB3,SLIT2,BIRC5,TCF4,THBS2,SPARC,MMP2,TNFRSF1B THY1,DCN,IGFBP7,PDGFRB,LOX,VCAN,HLA-F,COL1A2
GO:0001525	Angiogenesis	2.46E-08	DAB2,EPHB3,SLIT2,BIRC5,TCF4,THBS2,SPARC MMP2,THY1,DCN,IGFBP7,PDGFRB,HLA-F
GO:0035295	Tube development	1.39E-07	DAB2,EPHB3,SLIT2,SPDEF,BIRC5,TCF4,THBS2,SPARC,MMP2 THY1,DCN,IGFBP7,PDGFRB,LOX,VCAN,HLA-F
GO:0035239	Tube morphogenesis	1.02E-06	DAB2,EPHB3,SLIT2,BIRC5,TCF4,THBS2,SPARC,MMP2 THY1,DCN,IGFBP7,PDGFRB,LOX,HLA-F

that modulates the biological activity of the pro-inflammatory cytokine, interleukin-1 [28]. IL1RN has been implicated in several cancers.

Gene Ontology and Pathway Enrichment Analyses: We use an online enrichment tool, ToppGene Suite [29], to explore the biological relationships of these "stable" genes and CpG sites. We took these genes from the gene expression data and genes corresponding to the CpG sites as input for ToppFun in ToppGene Suite. Some of the biological processes enriched with gene ontology (GO) included vasculature development, tissue development, angiogenesis, and tube morphogenesis (see Tables 7 and 8). Some of the biological processes significantly enriched in our list of methylation include tissue morphogenesis, epithelial tube morphogenesis, and tube development. All these biological processes are essential in cell development, and aberrations or disruptions in these processes could result in cancer. Tables 9 and 10 show the top 10 pathways that are enriched in our list of methylated genes and genes, respectively. Some of these pathways included cancer and pathways related to extracellular matrix orgnaization (ECM). ECM is a complex collection of proteins and plays a key role in cell survival, cell proliferation, and

Table 8 Genes corresponding to CpG sites. Top 10 Gene Ontology (GO) Biological Processes enriched with ToppFun in ToppGene Suite

GO ID	GO Biological Process	Bonferroni *P*-value	Genes
GO:0048729	Tissue morphogenesis	0.00003361	ADAMTS12,IGFBP7,TGFB3,IL1RN,FGF2,WNT2,RARA,FABP3
GO:0060562	Epithelial tube morphogenesis	0.0002242	ADAMTS12,IGFBP7,FGF2,WNT2,RARA,FABP3
GO:0010092	Specification of animal organ identity	0.003616	FGF2,WNT2,RARA
GO:0060591	Chondroblast differentiation	0.004815	FGF2,RARA
GO:0008285	Negative regulation of cell population proliferation	0.004981	IGFBP7,TGFB3,FGF2,TFF1,RARA,FABP3
GO:0002009	Morphogenesis of an epithelium	0.005639	ADAMTS12,IGFBP7,FGF2,WNT2,RARA,FABP3
GO:0035295	Tube development	0.01252	ADAMTS12,IGFBP7,TGFB3,FGF2,WNT2,RARA,FABP3
GO:0048598	Embryonic morphogenesis	0.0138	TGFB3,IL1RN,FGF2,WNT2,RARA,FABP3
GO:0061035	Regulation of cartilage development	0.01498	ADAMTS12,TGFB3,RARA
GO:1905330	Regulation of morphogenesis of an epithelium	0.01603	ADAMTS12,FGF2,WNT2

Table 9 Genes corresponding to CpG sites. Top 10 Pathways enriched with ToppFun in ToppGene Suite

ID	Pathway	Source	Bonferroni *P*-value	Genes
M12868	Pathways in cancer	MSigDB C2 BIOCARTA	0.001107	TGFB3,FGF2,WNT2,RASSF1,RARA
M39427	Pluripotent stem cell differentiation pathway	MSigDB C2 BIOCARTA	0.002385	TGFB3,FGF2,WNT2
83105	Pathways in cancer	BioSystems: KEGG	0.002876	TGFB3,FGF2,WNT2,RASSF1,RARA
M5889	Ensemble of genes encoding extracellular matrix and extracellular matrix-associated proteins	MSigDB C2 BIOCARTA	0.02151	ADAMTS12,IGFBP7,TGFB3 IL1RN,FGF2,WNT2
M5883	Genes encoding secreted soluble factors	MSigDB C2 BIOCARTA	0.04072	TGFB3,IL1RN,FGF2,WNT2
M5885	Ensemble of genes encoding ECM-associated proteins including ECM-affilaited proteins, ECM regulators and secreted factors	MSigDB C2 BIOCARTA	0.06319	ADAMTS12,TGFB3,IL1RN, FGF2,WNT2
138010	Glypican 1 network	BioSystems: Pathway Interaction Database	0.06838	TGFB3,FGF2
M33	Glypican 1 network	MSigDB C2 BIOCARTA	0.07382	TGFB3,FGF2
749777	Hippo signaling pathway	BioSystems: KEGG	0.07853	TGFB3,WNT2,RASSF1
M12095	Signal transduction through IL1R	MSigDB C2 BIOCARTA	0.09762	TGFB3,IL1RN

Table 10 Genes selected. Top 10 Pathways enriched with ToppFun in ToppGene Suite

ID	Pathway	Source	Bonferroni *P*-value	Genes
1270244	Extracellular matrix organization	BioSystems: REACTOME	5.117E-08	COL6A1,MFAP4,SPARC,MMP2 DCN,TIMP2,LOX,VCAN,BGN,COL1A2
M5889	Ensemble of genes encoding extracellular matrix and extracellular matrix-associated proteins	MSigDB C2 BIOCARTA	0.000000478	SLIT2,COL6A1,MFAP4,THBS2 IL16,SPARC,MMP2,DCN,IGFBP7 ITIMP2,LOX,VCAN,BGN,COL1A2
1269016	Defective CHSY1 causes TPBS	BioSystems: REACTOME	0.0001055	DCN,VCAN,BGN
1269017	Defective CHST3 causes SEDCJD	BioSystems: REACTOME	0.0001055	DCN,VCAN,BGN
1269018	Defective CHST14 causes EDS, musculocontractural type	BioSystems: REACTOME	0.0001055	DCN,VCAN,BGN
1269986	Dermatan sulfate biosynthesis	BioSystems: REACTOME	0.0004947	DCN,VCAN,BGN
1269987	CS/DS degradation	BioSystems: REACTOME	0.001087	DCN,VCAN,BGN
1270256	ECM proteoglycans	BioSystems: REACTOME	0.001966	SPARC,DCN,VCAN,BGN
1309217	Defective B3GALT6 causes EDSP2 and SEMDJL1	BioSystems: REACTOME	0.002875	DCN,VCAN,BGN

differentiation [30]. ECM is involved in tumor progression, dissemination, and response to therapy [30, 31].

Goal 2: Model nonlinear relationships between methylation and gene expression data and derive molecular clusters

We demonstrate the use of the estimated shared low-dimensional representation and the reconstructed methylation and gene expression data in molecular clustering. For this purpose, we applied the proposed method (without Laplacian) to all data to identify the top 20% genes and CpG sites that could be used to nonlinearly approximate the original views. Then we obtained the shared low-dimensional representation ($\widetilde{\mathbf{Z}}'$), and the reconstructed views ($R_1(\mathbf{Z}')$, and $R_2(\mathbf{Z}')$) based only on the top 20% genes and CpG sites. We perform K-means clustering on $\widetilde{\mathbf{Z}}'$, $R_1(\mathbf{Z})$ and $R_2(\mathbf{Z})$. We set the number of clusters to 4, which is within the number of clusters investigated in the original article [22]. We compared the number of clusters detected with several variables related to breast cancer, including estrogen receptor (ER) status, progesterone receptor (PgR) status, survival time, and survival status for ten years. We obtained Kaplan-Meier (KM) curves to compare the survival curves for the identified clusters. We also fitted a Cox regression model to compare the estimated hazard ratios for 10-year survival. Finally, we performed an enrichment analysis of the top 20% genes and CpG sites.

Figure 8A shows the KM curves for the clusters detected using the low-dimensional shared representation (first panel) and the reconstructed gene expression (middle panel) and methylation data (right panel). From the KM plots, the 10-year survival curves for the clusters detected using the shared low-dimensional representation or the reconstructed methylation data are significantly different (p-value $= 0.041$ and 0.032, respectively, based on a log-rank test to compare survival curves). As reported in Table 11,

Table 11 Characteristics of the patients. Continuous variables are tested based on regular ANOVA with equal variance assumption, and categorical variables are tested based on the Chi-square test

n	Cluster 0	Cluster 1	Cluster 2	Cluster 3	p test
	32	51	35	50	
ER = er_pos (%)	7 (22.6)	37 (75.5)	18 (51.4)	43 (87.8)	<0.001
PgR = pgr_pos (%)	7 (22.6)	38 (77.6)	16 (45.7)	39 (79.6)	<0.001
Overall Survival					
Time (yr) (mean (SD))	9.14 (5.30)	11.13 (4.32)	11.87 (5.05)	8.68 (5.31)	0.010
Event = 1 (%)	11 (34.4)	18 (35.3)	10 (28.6)	26 (52.0)	0.125
Ten Year Survival					
Survived = 1 (%)	17 (53.1)	18 (35.3)	9 (25.7)	31 (63.3)	0.002
HuSubtype (%)					<0.001
Basal	21 (65.6)	6 (11.8)	12 (34.3)	0 (0.0)	
Her2	1 (3.1)	5 (9.8)	6 (17.1)	2 (4.0)	
LumA	4 (12.5)	15 (29.4)	6 (17.1)	19 (38.0)	
LumB	4 (12.5)	8 (15.7)	6 (17.1)	14 (28.0)	
non-classified	0 (0.0)	12 (23.5)	4 (11.4)	6 (12.0)	
Normal	2 (6.2)	5 (9.8)	1 (2.9)	9 (18.0)	
ageYear (mean (SD))	48.84 (9.64)	52.16 (11.75)	47.74 (10.46)	49.80 (12.14)	0.308

the clusters (from shared low-dimensional representations) are significantly associated with ER, PgR, overall survival time, and 10-year survival event. Individuals in Cluster 3 seemed to have worse survival outcomes compared to individuals in Cluster 0. In particular, the proportion of individuals in Cluster 3 with ER/PgR negative tumors was higher, the 10-year survival rate was lower (only 40% of participants from Cluster 0 survived while 69% of participants from Cluster 1 survived, Fig. 8B), and the average survival time was shorter compared to those in Cluster 1 (Fig. 8C). Furthermore, the estimated unadjusted risk ratio for 10-year survival for those in Cluster 3 compared to those in Cluster 0 was 1.409 (Fig. 8D 95%CI: 1.686–2.894, p-value = 0.04), suggesting that being in Cluster 3 reduces your survival rate by a factor of 1.41 at each point during 10-year follow-up compared to Cluster 0. This effect persisted even after adjusting for age or age and ER status. Significantly enriched pathways, as shown in Fig. 7, from our gene list and CpG sites (genes corresponding to the top 20% CpG sites) include ECM, inflammatory response pathway, and pathways in cancer.

Evaluation of brain lower grade glioma data

We applied our method to data pertaining to brain lower grade glioma (LGG) to identify molecules that discriminate between levels of LGG grade (grade 2 vs 3 gliomas). We obtained data from the Board GDAC Firehose of the Cancer Genome Atlas Program (TCGA).[1] We used three types of omics data: methylation, miRNA, and mRNAseq, following the analysis in [32]. Only patients with all available omics and classifications of grade were included in our analyses, giving a total sample size of 510, with 246 patients classified as grade 2 and 264 patients as grade 3. We used the LGG dataset to

[1] https://gdac.broadinstitute.org.

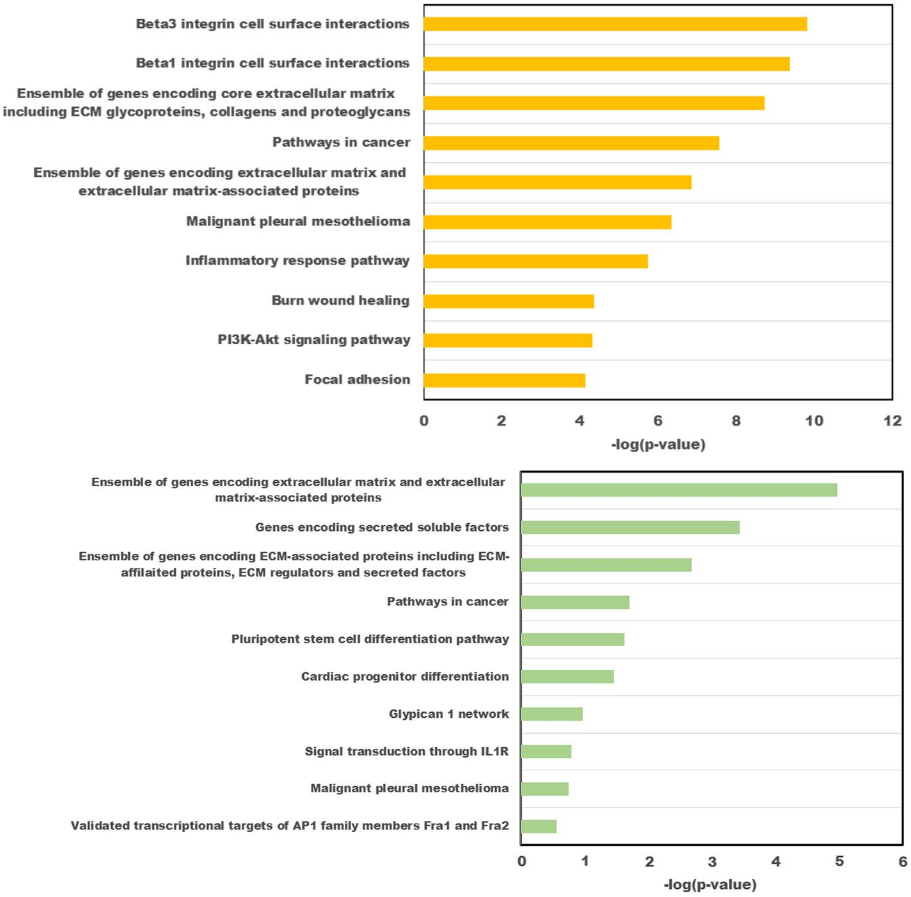

Fig. 7 Top 10 significant pathways using highly-ranked genes (Top Panel) and genes corresponding to highly-ranked CpG sites (Bottom Panel)

demonstrate that the proposed method can be used to associate three views, select important biomarkers, and predict patient grade category.

Data cleaning and data preprocessing were carried out on each view of data to remove features with low potential for discrimination. For all views, we first removed features with missing measures. Due to the limited number of features left in the miRNA view after removing missing values, future preprocessing was conducted only on the DNA methylation view and the mRNAseq view. Unsupervised preprocessing was applied to remove features whose variance was less than 0.001 for DNA methylation measures and 0.1 for mRNAseq measures, following the thresholds used in [32]. The data were then divided into training $(n = 410)$ and testing $(n = 100)$ sets and supervised preprocessing was conducted on the training set. Logistic regression was fitted for each feature in the DNA methylation view and the mRNAseq view. The p-values were adjusted by the Benjamini-Hochberg procedure, and the features with adjusted p-values < 0.05 were kept in the dataset. After data cleaning and preprocessing, the number of features for DNA methylation, miRNA, and mRNAseq was 9691, 235, and 7603 respectively.

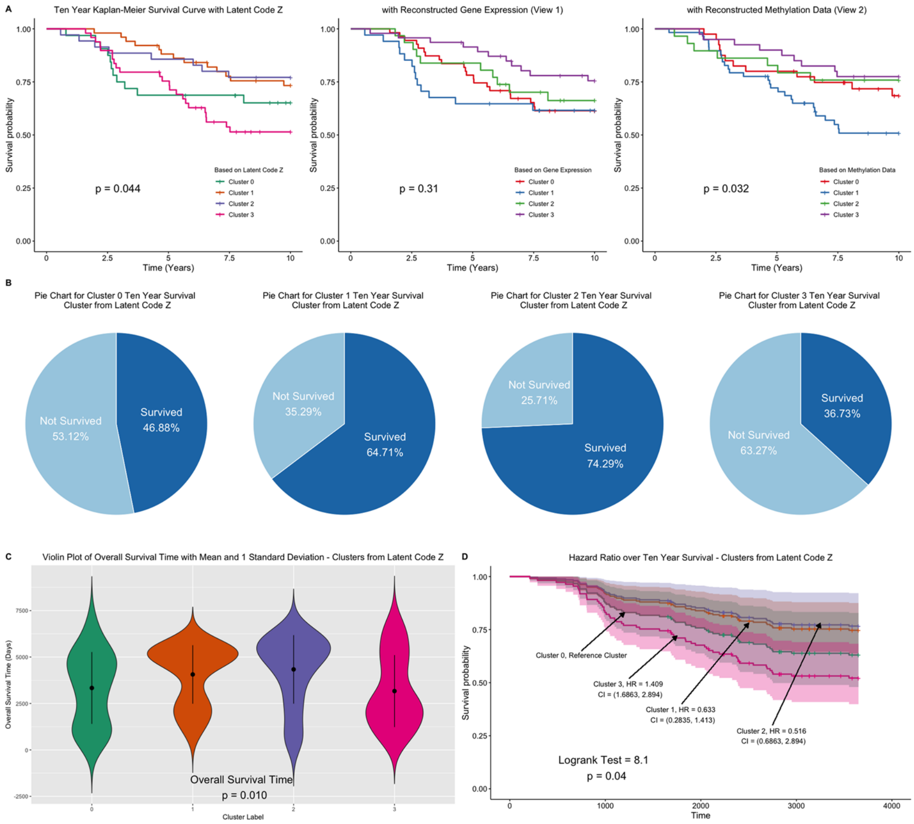

Fig. 8 Top 20% genes and CpG sites that approximate the original data are used to obtain shared low-dimensional representations, and reconstructed gene expression and methylation data. **A** Kaplan–Meier plots comparing survival curves for clusters obtained from the shared low-dimensional representations, and the reconstructed data. Survival curves for the clusters based on the joint and low-dimensional representations and reconstructed methylation data are significantly different. **B**–**D** Clusters are derived from the shared low-dimensional representation. **B** Comparison of 10-year survival rates across clusters. Chi-square test of independence shows that the clusters detected are significantly associated with 10-year survival event (*p*-value = 0.011). **C** Violin plot of overall survival time by clusters. The average survival times are significantly different across clusters. **D** Comparison of hazard ratios and survival curves across clusters

Table 12 LGG dataset: SVM and random forest are based on stacked views. Deep IDA + SVM means selecting features from Deep IDA and training an SVM classifier on these features. iDeepViewLearn with selected top 50 features obtains a classification error based on a shared low-dimensional representation trained on data with the selected top 50 features. Similar for iDeepViewLearn with selected top 100 features

Method	AverageError (%)
SVM on stacked data	30.00
Random Forest on stacked data	26.00
iDeepViewLearn with selected top 50 features	28.00
iDeepViewLearn with selected top 100 features	26.00
SIDA	29.00
Deep GCCA + SVM	29.00
Deep IDA	28.00
Deep IDA + SVM	26.00

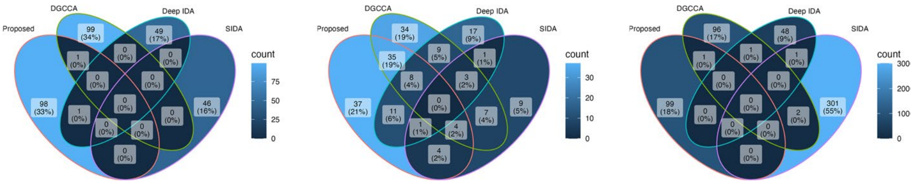

Fig. 9 Venn diagrams of features selected by the proposed method, and the three comparison methods that are capable of feature selection. The left, middle, and right panels correspond to DNA methylation, miRNA, and mRNAseq, respectively. The percentages represent the proportion of the total selected features from the four methods

We applied the proposed approach to the training dataset, where we selected the important features from each type of omics data. Subsequently, we used these selected features to make predictions for the patient's grade category in the testing dataset, as shown in Table 12. We used cross-validation to tune hyper-parameters based on the training set. Our proposed method was compared with Deep Generalized Canonical Correlation Analysis (Deep GCCA) [33] with PyTorch implementation.[2] We added the teacher-student network (TS) [20] for feature selection, and implemented SVM for classification; Deep IDA [9]; Features selected from Deep IDA with SVM for classification; SIDA [10], and SVM and Random Forest on stacked data. The classification performance of our method is comparable to other methods (Table 12).

In Fig. 9, we show the overlaps of features selected by the methods. We used the top 100 features of each view selected by the proposed method. We compare the top 100 features selected by the TS network with Deep GCCA and the top 50 features selected by the TS network with Deep IDA. SIDA selected 46, 29, and 304 features for each omics, respectively. We presented the overlaps between the selected genes across the four methods matched from NCBI.[3] The overlaps between 2 or more methods of DNA methylation were COL11A2 and FBLN2. The overlaps between 3 or more methods for miRNA view were MIR379, MIR409, MIR29C, MIR129-1, MIR20B, MIR30E, MIR92A2, MIR222, MIR24-2, MIR767, MIR128-2, MIR105-2, and MIR17. The overlaps between 2 or more methods for mRNAseq view were NCAPH, LY86-AS1, HSFX2, and SLC25A41.

Evaluation of shear transformed MNIST data

We apply our method to the MNIST dataset [34]. The MNIST handwritten image dataset consists of 70,000 images of handwritten digits divided into training and testing sets of 60,000, and 10,000 images, respectively. The digits have been size-normalized and centered in a fixed-size image. Each image is 28 × 28 pixels and has an associated label that denotes which digit the image represents (0–9). We make good use of a shear mapping to generate a second view of these handwritten digits. A shear mapping is a linear map that displaces each point in a fixed direction by an amount proportional to its signed distance from the line that is parallel to that direction and goes through the origin. Figure 10 shows two image plots of a digit for views 1 and 2.

[2] https://github.com/arminarj/DeepGCCA-pytorch.

[3] https://www.ncbi.nlm.nih.gov/.

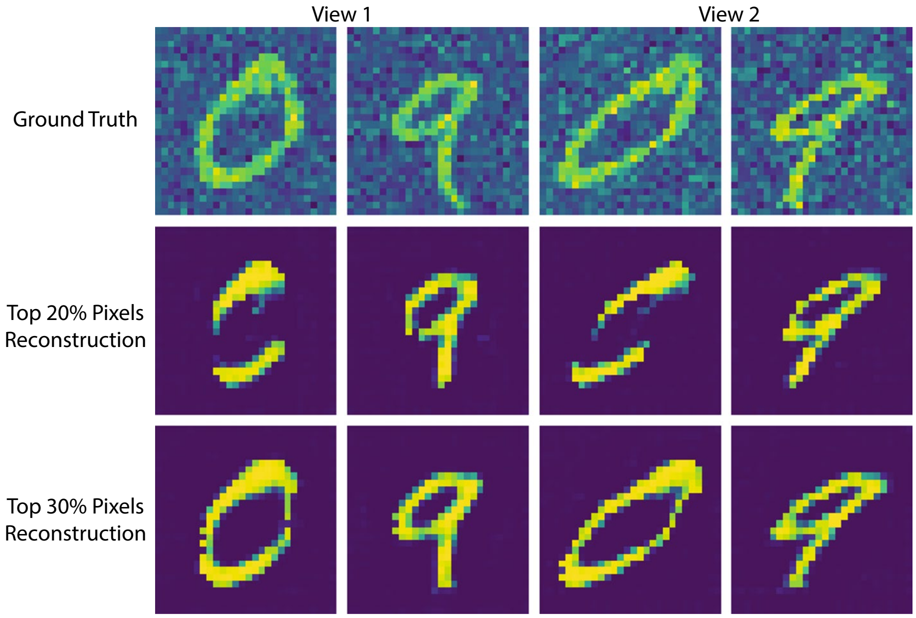

Fig. 10 An example of shear transformed MNIST dataset. For the subject with label "0" and "9", view 1 observation is on the left and view 2 observation is on the right. Notably, we show the grayscale images with color only for better visualization

Table 13 MNIST dataset: SVM is based on stacked views. Deep CCA + SVM is a training SVM based on the last layer of Deep CCA. iDeepViewLearn with selected top 20% pixels obtains a classification error based on a shared low-dimensional representation trained on data with the selected 20% of the pixels. Similar for iDeepViewLearn with selected top 30%

Method	**AverageError (%)**
Deep CCA + SVM	2.97
SVM on stacked data	2.81
iDeepViewLearn with selected top 20% pixels	3.91
iDeepViewLearn with selected top 30% pixels	**2.56**

We used the MNIST dataset to demonstrate the ability of the proposed method to reconstruct handwritten images using a few pixels. In particular, neural networks G_d consist of convolutional layers instead of fully connected layers, since they reconstruct images. We apply the proposed method to the training dataset, select 20% and 30% of the pixels based on our variable ranking criteria and reconstruct the images using only the selected pixels. We also learn a new model with these pixels, we use the learned model and the testing data to classify the test digits, and we obtain the test errors. Figure 10 shows the reconstructed images based on the top 20% and 30% pixels. The digits are apparent even with only 30% of the pixels. From Table 13, the classification performance using the top 30% of the pixels is comparable to Deep CCA and SVM, which use all pixels. Even when only 20% of the pixels were selected and used to reconstruct the images, the classification performance of our method was competitive.

Discussion

We have presented iDeepViewLearn, short for Interpretable Deep Learning Method for Multiview Learning, to learn nonlinear relationships in data from multiple sources. iDeepViewLearn combines the flexibility of deep learning with the statistical advantages of data- and knowledge-driven feature selection to yield interpretable results. In particular, iDeepViewLearn learns low-dimensional representations of the views that are common to all the views and assumes that each view can be approximated by a nonlinear function of the shared representations. Deep neural networks are used to model the nonlinear function and an optimization problem that minimizes the difference between the observed data and the nonlinearly transformed data are used to reconstruct the original data. A regularization penalty is imposed on the reconstructed data in the optimization problem, permitting us to reconstruct each view only with relevant variables. Beyond the data-driven approach for feature selection, we also consider a knowledge-based approach to identify relevant features. We use the normalized Laplacian of a graph to model bilateral relationships between variables in each view and to encourage the selection of connected variables.

We have developed a user-friendly algorithm in Python 3, specifically PyTorch, and interfaced it with R to increase the reach of our method. Extensive simulations with varying data dimensions and complexity revealed that iDeepViewLearn outperforms several other linear and nonlinear methods for integrating data from multiple views, even in high-dimensional scenarios where the sample size is typically smaller than the number of variables.

When iDeepViewLearn was applied to methylation and gene expression data related to breast cancer, we observed that iDeepViewLearn is capable of achieving meaningful biological insights. We identified several CpG sites and genes that better discriminated people who died from breast cancer and those who did not. The biological processes of the gene ontology enriched in the top-ranked genes and methylated CpG sites included processes essential to cell proliferation and death. The enriched pathways included cancer and others that have been implicated in tumor progression and response to therapy. Using the shared low-dimensional representations of gene expression and methylation data from our method, we detected four molecular clusters that differed in their 10-year survival rates. The enrichment analysis of highly ranked genes and genes corresponding to the CpG sites selected by our method showed a strong enrichment of pathways and biological processes, some related to breast cancer and others that could be further explored for their potential role in breast cancer. We also applied iDeepViewLearn to DNA methylation, miRNA, and mRNASeq data pertaining to Brain Lower Grade Glioma (LGG) and found our method to be competitive in discriminating between LGG categories, demonstrating the ability of our methods to be used for more than two views. We further applied iDeepViewLearn to handwritten image data and we were able to reconstruct the digits with about 30% pixels while also achieving competitive classification accuracy. For more applications, e.g., drug repositioning [35–37], we leave them for future work. A limitation of our work is that the number (or proportion) of top-ranked features needs to be specified in advance.

Conclusion

In conclusion, we have developed deep learning methods to learn nonlinear relationships in multiview data that are able to identify features likely driving the overall association in the views. The simulations and real data applications are encouraging, even for scenarios with small to moderate sample sizes, thus we believe the methods will motivate other applications.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-024-05679-9.

Additional file 1. contains information on our optimization and algorithm, hyper-parameter selection and linear simulations results.

Author contributions

SES and JS conceived of the idea and developed the methods. HW developed algorithms to implement the methods. HW and HL conducted simulations and real data analyses. SES and HW wrote a first draft of the paper. All authors edited and read the final manuscript.

Funding

The project described was supported by the Award Number 1R35GM142695 of the National Institute of General Medical Sciences of the National Institutes of Health. The content is solely the responsibility of the authors and does not represent the official views of the National Institutes of Health.

Availibility of data and materials

The data used were obtained from [22]. We provide Python codes and an R package, ***iDeepViewLearn***, to facilitate the use of our method. Its source codes, along with a README file, are available at: https://github.com/lasandrall/iDeepViewLearn.

Declarations

Ethics approval and consent to participate

Not Applicable. This research uses publicly available data.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 22 May 2023 Accepted: 29 January 2024
Published online: 14 February 2024

References

1. Hotelling H. Relations between two sets of variables. Biometrika. 1936;28:312–77.
2. Safo SE, Ahn J, Jeon Y, Jung S. Sparse generalized eigenvalue problem with application to canonical correlation analysis for integrative analysis of methylation and gene expression data. Biometrics. 2018;74(4):1362–71.
3. Akaho, S. A kernel method for canonical correlation analysis. Int'l Meeting on Psychometric Society. 2001.
4. Lopez-Paz D, Sra S, Smola A, Ghahramani Z, Schölkopf B. Randomized nonlinear component analysis. In: International Conference on Machine Learning, 2014;pp. 1359–1367 . PMLR
5. Andrew G, Arora R, Bilmes J, Livescu K. Deep canonical correlation analysis. In: Journal of Machine Learning Research: Workshop and Conference Proceedings. 2013.
6. Benton A, Khayrallah H, Gujral B, Reisinger DA, Zhang S, Arora R. Deep generalized canonical correlation analysis. In: Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019), 2019;1–6
7. Lee C, Schaar M. A variational information bottleneck approach to multi-omics data integration. In: International Conference on Artificial Intelligence and Statistics, 2021;pp. 1513–1521. PMLR
8. Moon S, Lee H. MOMA: a multi-task attention learning algorithm for multi-omics data interpretation and classification. Bioinformatics. 2022;38(8):2287–96. https://doi.org/10.1093/bioinformatics/btac080.
9. Wang J, Safo SE. Deep ida: a deep learning method for integrative discriminant analysis of multi-view data with feature ranking—an application to covid-19 severity. ArXiv, 2021;2111
10. Safo SE, Min EJ, Haine L. Sparse linear discriminant analysis for multiview structured data. Biometrics. 2021. https://doi.org/10.1111/biom.13458.
11. Safo SE, Li S, Long Q. Integrative analysis of transcriptomic and metabolomic data via sparse canonical correlation analysis with incorporation of biological information. Biometrics. 2018;74(1):300–12.
12. Tibshirani R. Regression shrinkage and selection via the lasso. J R Stat Soc Ser B (Methodol). 1996;58(1):267–88.

13. Zou H, Hastie T. Regularization and variable selection via the elastic net. J R Stat Soc Ser B Stat Methodol. 2005;67(2):301–20.
14. Fan J, Li R. Variable selection via nonconcave penalized likelihood and its oracle properties. J Am Stat Assoc. 2001;96:1348–60.
15. Nie F, Huang H, Cai X, Ding CHQ. Efficient and robust feature selection via joint $\ell 2$, 1-norms minimization. In: Lafferty, J.D., Williams, C.K.I., Shawe-Taylor, J., Zemel, R.S., Culotta, A. (eds.) Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a Meeting Held 6–9 December 2010, Vancouver, British Columbia, Canada, pp. 1813–1821. Curran Associates, Inc., 2010. https://proceedings.neurips.cc/paper/2010/hash/09c6c3783b4a70054da74f2538ed47c6-Abstract.html
16. Peri S, Navarro JD, Amanchy R, Kristiansen TZ, Jonnalagadda CK, Surendranath V, Niranjan V, Muthusamy B, Gandhi T, Gronborg M, et al. Development of human protein reference database as an initial platform for approaching systems biology in humans. Genome Res. 2003;13(10):2363–71.
17. Chung FR, Graham FC. Spectral graph theory. London: American Mathematical Soc; 1997.
18. Ben-Hur A, Ong CS, Sonnenburg S, Schölkopf B, Rätsch G. Support vector machines and kernels for computational biology. PLoS Comput Biol. 2008;4(10):1000173.
19. Breiman L. Random forests. Mach Learn. 2001;45(1):5–32. https://doi.org/10.1023/A:1010933404324.
20. Mirzaei A, Pourahmadi V, Soltani M, Sheikhzadeh H. Deep feature selection using a teacher–student network. Neurocomputing. 2020;383:396–408.
21. Mohammadi R, Wit EC. Bdgraph: An R package for Bayesian structure learning in graphical models. arXiv preprint arXiv:1501.05108.
22. Holm K, Hegardt C, Staaf J, Vallon-Christersson J, Jönsson G, Olsson H, Borg Å, Ringnér M. Molecular subtypes of breast cancer are associated with characteristic dna methylation patterns. Breast Cancer Res. 2010;12(3):1–16.
23. Giaquinto AN, Sung H, Miller KD, Kramer JL, Newman LA, Minihan A, Jemal A, Siegel RL. Breast cancer statistics, 2022. CA Cancer J Clin. 2022;72(6):524–41.
24. Lustberg MB, Ramaswamy B. Epigenetic therapy in breast cancer. Curr Breast Cancer Rep. 2011;3:34–43.
25. Järvinen TA, Prince S. Decorin: a growth factor antagonist for tumor growth inhibition. BioMed Res Int 2015, 2015.
26. Oparina N, Erlandsson MC, Beding AF, Parris T, Helou K, Karlsson P, Einbeigi Z, Bokarewa MI. Prognostic significance of birc5/survivin in breast cancer: results from three independent cohorts. Cancers. 2021;13(9):2209.
27. Li F, Ambrosini G, Chu EY, Plescia J, Tognin S, Marchisio PC, Altieri DC. Control of apoptosis and mitotic spindle checkpoint by survivin. Nature. 1998;396(6711):580–4.
28. Shiiba M, Saito K, Yamagami H, Nakashima D, Higo M, Kasamatsu A, Sakamoto Y, Ogawara K, Uzawa K, Takiguchi Y, et al. Interleukin-1 receptor antagonist (il1rn) is associated with suppression of early carcinogenic events in human oral malignancies. Int J Oncol. 2015;46(5):1978–84.
29. Chen J, Bardes EE, Aronow BJ, Jegga AG. Toppgene suite for gene list enrichment analysis and candidate gene prioritization. Nucleic Acids Res. 2009;37(suppl–2):305–11.
30. Naba A, Clauser KR, Ding H, Whittaker CA, Carr SA, Hynes RO. The extracellular matrix: Tools and insights for the "omics" era. Matrix Biol. 2016;49:10–24.
31. Henke E, Nandigama R, Ergün S. Extracellular matrix in the tumor microenvironment and its impact on cancer therapy. Front Mol Biosci. 2020;6:160.
32. Wang T, Shao W, Huang Z, Tang H, Zhang J, Ding Z, Huang K. MOGONET integrates multi-omics data using graph convolutional networks allowing patient classification and biomarker identification. Nat Commun. 2021;12(1):3445. https://doi.org/10.1038/s41467-021-23774-w.
33. Benton A, Khayrallah H, Gujral B, Reisinger DA, Zhang S, Arora R. Deep generalized canonical correlation analysis. arXiv:1702.02519, 2017. Accessed 2023-10-24
34. Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86(11):2278–324. https://doi.org/10.1109/5.726791.
35. Zhao B-W, Su X-R, Hu P-W, Ma Y-P, Zhou X, Hu L. A geometric deep learning framework for drug repositioning over heterogeneous information networks. Brief Bioinform. 2022;23(6):384. https://doi.org/10.1093/bib/bbac384.
36. Zhao B-W, Wang L, Hu P-W, Wong L, Su X-R, Wang B-Q, You Z-H, Hu L. Fusing higher and lower-order biological information for drug repositioning via graph representation learning. IEEE Trans Emerg Top Comput. 2023. https://doi.org/10.1109/TETC.2023.3239949.
37. Zhao B-W, Su X-R, Hu P-W, Huang Y-A, You Z-H, Hu L. iGRLDTI: an improved graph representation learning method for predicting drug-target interactions over heterogeneous biological information network. Bioinformatics. 2023;39(8):451. https://doi.org/10.1093/bioinformatics/btad451.

Publisher's Note