Learning Multiple Views with Orthogonal Denoising Autoencoders

TengQi Ye[1(✉)], Tianchun Wang[2], Kevin McGuinness[1], Yu Guo[3], and Cathal Gurrin[1]

[1] Insight Centre for Data Analytics, Dublin City University, Dublin, Ireland
{yetengqi,kevin.mcguinness}@gmail.com
[2] School of Software, TNList, Tsinghua University, Beijing, China
wtc13@mails.tsinghua.edu.cn
[3] Department of Computer Science, City University of Hong Kong, Hong Kong, China

Abstract. Multi-view learning techniques are necessary when data is described by multiple distinct feature sets because single-view learning algorithms tend to overfit on these high-dimensional data. Prior successful approaches followed either consensus or complementary principles. Recent work has focused on learning both the shared and private latent spaces of views in order to take advantage of both principles. However, these methods can not ensure that the latent spaces are strictly independent through encouraging the orthogonality in their objective functions. Also little work has explored representation learning techniques for multi-view learning. In this paper, we use the denoising autoencoder to learn shared and private latent spaces, with orthogonal constraints — disconnecting every private latent space from the remaining views. Instead of computationally expensive optimization, we adapt the backpropagation algorithm to train our model.

Keywords: Denoising autoencoder · Autoencoder · Representation learning · Multi-view learning · Multimedia fusion

1 Introduction

In many machine learning problems, data samples are collected from diverse sensors or described by various features and inherently have multiple disjoint feature sets (conventionally referred as views) [1,2]. For example, in video classification, the videos can be characterized with respect to vision, audio and even attached comments; most article search engines take title, keywords, author, publisher, date and content into consideration; images have different forms of descriptors: color descriptors, local binary patterns, local shape descriptors, etc. The last example reveals the noteworthy case of views obtained from manual descriptors instead of natural splits. With multiple descriptors, important information concerning the task, which may be discarded by single descriptor, is hopefully retained by others [3].

Q. Tian et al. (Eds.): MMM 2016, Part I, LNCS 9516, pp. 313–324, 2016.
DOI: 10.1007/978-3-319-27671-7_26

Traditional machine learning methods may fail when concatenating all views into one for learning because the concatenation can cause over-fitting due to the high-dimensionality of the features (the curse of dimensionality [4]) and also ignores the specific properties of each view [2]. Compared with traditional single-view data, the multi-view data contains significant redundancy shared by its views. Previous successful multi-view learning algorithms follow two principles: consensus and complementary principles [2]. The consensus principle aims to maximize the agreement or shared knowledge between views. The complementary principle asserts each view may contain useful knowledge that the rest do not have, and errors can be corrected by this private knowledge.

To generalize over previous multi-view learning approaches, recent works focused on explicitly accounting for the dependencies and independencies of views, i.e., decompose the latent space into a shared common one (of all views) and several private spaces (of each view) [5,6]. The intuition is that each view is generated from the combination of the shared latent space and a corresponding private latent space. Although these methods benefit from the idea, they embed the orthogonality requirement into the objective function with a weight to set its relative influence, i.e., they encourage rather than restrict the orthogonality. The main reason is that it is hard to optimize an objective function with complex constraints. However, in this case, orthogonality may not be satisfied and extra costly computation effort is needed.

Representation learning, which seeks good representations or features that facilitate learning functions of interest, has promising performance in single-view learning [7]. From the perspective of representation learning, multi-view learning can be viewed as learning several latent spaces or features with orthogonality constraints. Only very few works have discussed the similarity between representation learning, in the form of sparse learning, and multi-view learning [5,8].

In this paper, we propose using the denoising autoencoder to learn the shared and private latent spaces with orthogonality constraints. In our approach, the constraints are satisfied by disconnecting every private latent space from the remaining views. The advantages of our proposed method are: (i) By disconnecting the irrelevant latent spaces and views, the orthogonality constraints are enforced. (ii) Such constraints keep the chain rule almost the same, thus simplify training the model, i.e., no extra effort is needed for tuning weights or complex optimization. (iii) No preprocessing is required for denoising because the denoising autoencoder learns robust features.

2 Related Work

Existing multi-view learning algorithms can be classified into three groups: co-training, multiple kernel learning, and subspace learning [2]. Co-training [9] was the first formalized learning method in the multi-view framework. It trains repeatedly on labeled and unlabeled data until the mutual agreement on two distinct views is maximized. Further extensions include: an improved version with the Bayesian undirected graphical model [10], and application in active multi-view learning [11]. Multiple Kernel Learning naturally corresponds to different

modalities (views) and combining kernels either linearly or non-linearly improves learning performance [12]. Multiple Kernel Learning always comes with diverse linear or nonlinear constraints, which makes the objective function complex [13–15]. Subspace learning-based approaches aim to obtain latent subspaces that have lower dimensions than input views, thus effective information is learned and redundancy is discarded from views. As those latent spaces can be regarded as effective features, subspace learning-based algorithms allow single-view learning algorithms to be capable for learning on multi-view data.

Subspace learning-based algorithms initially targeted on conducting meaningful dimensional reduction for multi-view data [2]. Canonical correlation analysis based approaches, following the consensus principle, linearly or non-linearly project two different views into the same space where the correlation between views is maximized [16,17]. Because the dimension of the space to be projected on equals to the smaller one of the two views, the dimension is reduced by at least half. Other similar approaches include Multi-view Fisher Discriminant Analysis, Multi-view Embedding and Multi-view Metric Learning [2].

Unlike other types of subspace learning-based algorithms, latent subspace learning models resort to explicitly building a shared latent space and several private latent spaces (a private latent space corresponds to a view). The Factorized Orthogonal Latent Space model proposes to factorize the latent space into shared and private latent spaces by encouraging these spaces to be orthogonal [6]. In addition, it penalized the dimensions of latent spaces to reduce redundancy. Similarly, a more advanced version is employed with sparse coding of structured sparsity for the same purpose [5]. Nevertheless, in their approaches, orthogonality is encouraged by penalizing inner products of latent spaces or encouraging the structured sparsity in the objective function. These approaches to not guarantee orthogonality.

Representation learning focuses on learning good representations or extracting useful information from data that simplifies further learning tasks [18]. A well-known successful example of representation learning is deep learning. By stacking autoencoders to learn better representations for each layer, deep learning drastically improves the performance of neural networks in tasks such as image classification [19], speech recognition [20], and natural language processing [21]. An autoencoder is simply a neural network that tries to copy its input to its output [7].

Multi-view feature learning was first analyzed from the perspective of representation learning through sparse coding in [8], but little work has employed representation learning for studying multi-view data so far, thus far only sparse coding has been used [5,22]. Since the autoencoder is the most prevalent representation learning algorithm, our model enables various existing work on autoencoder to inherently extend it to multi-view settings.

3 Approach

3.1 Problem Formulation

Let $X = \{X^{(1)}, X^{(2)}, \cdots, X^{(V)}\}$ be a data set of N observations from V views and $X^{(v)}$ be the v^{th} view of data, where $X^{(v)} \in \mathbb{R}^{N \times P(X^{(v)})}$. $P(\cdot)$ is the number of columns of the matrix; and $D(\cdot)$ is the number of features that the space or matrix has. Additionally, $Y \in \mathbb{R}^{N \times P(Y)}$ is the shared latent space across all views; $Z = \{Z^{(1)}, Z^{(2)}, \cdots, Z^{(V)}\}$ is the set of private latent spaces of each individual view, where $Z^{(v)} \in \mathbb{R}^{N \times P(Z^{(V)})}$. Because the latent spaces are required to have no redundancy, then $P(Y) = D(Y)$ and $P(Z^{(v)}) = D(Z^{(v)})$. Moreover, $X^{(v)}$ is expected to be linearly represented by Y and $Z^{(v)}$, $D(X^{(v)}) = D(Y) + D(Z^{(v)})$. Our goal is to learn Y and the Z from X where Y is independent of Z and the arbitrary two private latent spaces $Z^{(v_i)}, Z^{(v_j)}$ are orthogonal.

3.2 Basic Autoencoder

A standard autoencoder takes an input vector x and initially transforms it to a hidden representation vector $y = s(Wx + b)$ through an activation function s and weights W. Note the activation function can be linear or nonlinear. The latent representation y is subsequently mapped back to a reconstructed vector $z = s(W'y + b')$. The objective is that the output z is as close as possible to x, i.e., the parameters are optimized to minimize the average reconstruction error:

$$W^\star, b^\star, W'^\star, b'^\star = \underset{W,b,W',b'}{\arg\min}\, L(x, z) \tag{1}$$

where L is a loss function to measure how good the reconstruction is, and often least squares $L = \|(x - z)\|^2$ is used. The expectation is that the hidden representation y could capture the main factors of data x [23].

3.3 Orthogonal Autoencoder for Multi-view Learning

In the scenarios of multi-view learning, the hidden or latent representation (neuron) is expected to be consist of shared and private spaces. To this end, we modify the autoencoder such that every private latent space only connects to its own view. Figure 1 depicts the graphical model of such improved autoencoder given that the first view has two original features while the second one has three.

Because the first view is disconnected from the second private latent space (the third hidden neuron), the second private latent space is strictly independent of the first view. Similarly, the first private latent space is independent of the second view. In order to maintain orthogonality of private latent spaces, the bias is disconnected from private latent spaces (proof is given below).

In addition, in order to retain that views are linearly dependent of latent spaces, the hidden representation before the nonlinear mapping $(Wx + b)$ is regarded as latent spaces. If the activation function is nonlinear, then y and

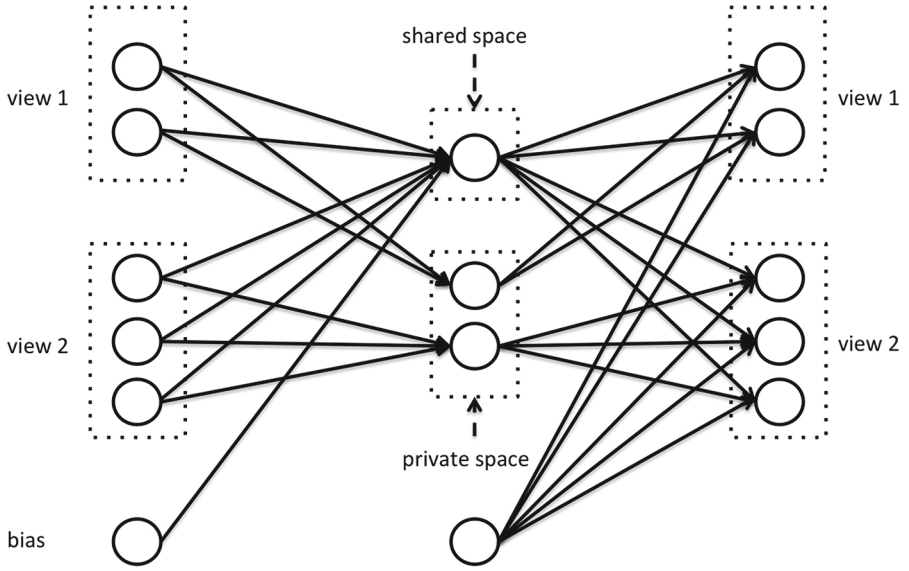


Fig. 1. Graphical model of an orthogonal autoencoder for multi-view learning with two views.

$W'y + b'$ can be considered as mappings of latent representation and reconstruction of input in a nonlinear space. And the last activation function maps the $W'y + b'$ back to original space.

Also note that the number of neurons representing the shared latent space equals the dimensions of its features $D(Y)$ and it is the same for private latent space, i.e., a neuron represents a feature in the space. Our model is inherently able to fit in any number of views and arbitrary numbers of features for each view and latent space.

Following the aforementioned notations, we further define $I(A|B)$ as the indices of columns of A in terms of B if matrix A is a submatrix of matrix B and they have same row numbers. The orthogonality constraints on weights can be formulated as:

$$W_{I(Z^{(v_2)}|[Y,Z]),I(X^{(v_1)}|X)} = 0 \qquad (v_1 \neq v_2) \tag{2}$$

$$W'_{I(X^{(v_1)}|X),I(Z^{(v_2)}|[Y,Z])} = 0 \qquad (v_1 \neq v_2) \tag{3}$$

$$b_{I(Z|[Y,Z])} = 0 \tag{4}$$

For a matrix A, the symbol $A_{I,\cdot}$ denotes a sub-matrix consisting of row vectors indexing by I (of A); similarly, $A_{\cdot,I}$ is a sub-matrix from such column vectors. We provide the rigorous proof that arbitrary two private latent spaces ($Z^{(v_1)}$ and $Z^{(v_2)}$) are orthogonal:

$$
\begin{aligned}
&(Z^{(v_1)})^T \cdot Z^{(v_2)} = (W_{I(Z^{(v_1)}|[Y,Z]),\cdot} \cdot x + \mathbf{0})^T \cdot (W_{I(Z^{(v_2)}|[Y,Z]),\cdot} \cdot x + \mathbf{0}) \\
&= x^T \cdot ((W_{I(Z^{(v_1)}|[Y,Z]),\cdot})^T \cdot W_{I(Z^{(v_2)}|[Y,Z]),\cdot}) \cdot x = x^T \cdot \mathbf{0} \cdot x \\
&= \mathbf{0}
\end{aligned}
$$

$[(Z^{(v_1)})^T \cdot Z^{(v_2)}]_{ij} = 0$ indicates that the component $[Z^{(v_1)}]_{\cdot,i}$ is orthogonal to $[Z^{(v_2)}]_{\cdot,j}$. Because any two components of any two different private latent spaces are orthogonal, the private latent spaces are orthogonal to each other.

Although we do not explicitly restrict Y to be orthogonal to Z, the objective function (Eq. 1) prefers that the shared latent space is orthogonal to the private latent spaces. Because if Y contains components from any views, i.e. not orthogonal to private latent spaces, then the components of that view will introduce noise into the others during reconstruction.

3.4 Training of Orthogonal Autoencoder

The autoencoder, intrinsically a neural network, is trained by the backpropagation algorithm, which propagates the derivation of the error from the output to the input, layer-by-layer [4]. As the 0 values automatically break gradients passing through that connection, we only need to set Eqs. 5, 6 and 7 after basic backpropagation as follows:

$$
\frac{\partial L}{W_{I(Z^{(v_2)}|[Y,Z]),I(X^{(v_1)}|X)}} = 0 \qquad (v_1 \neq v_2) \tag{5}
$$

$$
\frac{\partial L}{W'_{I(X^{(v_1)}|X),I(Z^{(v_2)}|[Y,Z])}} = 0 \qquad (v_1 \neq v_2) \tag{6}
$$

$$
\frac{\partial L}{b_{I(Z|[Y,Z])}} = 0 \tag{7}
$$

3.5 Orthogonal Denoising Autoencoder for Robust Latent Spaces

The denoising autoencoder was formally proposed to enforce the autoencoder to learn robust features or latent spaces in our case [24]. The idea is based on the assumption that robust features can reconstruct or repair input which is partially destroyed or corrupted. A typical way of producing $\tilde{x}$, the corrupted version of initial input x, is randomly choosing a fixed number of components and forcing their values to be 0; while other components stays the same. As a consequence, the autoencdoer is encouraged to learn robust features which are most likely to recover the wiped information.

Afterwards the $\tilde{x}$ is fed as input to the autoencoder, then mapped to the hidden representation $y = s(W\tilde{x} + b)$ and finally transformed to output $z = s(W'y + b')$. In the process, the aforementioned orthogonality constraints (Eqs. (2) and (3)) remain the same. In the end, the objective function L enforces the output z to be as close as possible to the original, uncorrupted x, instead of input $\tilde{x}$.

4 Experiments

In this section, we introduce two datasets to evaluate our method: the synthetic dataset is straightforward to demonstrate the ability of our method to learn shared and private latent spaces from data with noise; and the real-world dataset is employed to compare our approach with other state-of-the-art algorithms and display optimization on the number of neurons for vieww using random search.

4.1 Synthetic Dataset

We evaluated our approach with a toy dataset similar to [6], which can be generated in 10 lines of MATLAB code (listed in Algorithm 1).

Algorithm 1. Toy data generation in MATLAB notation

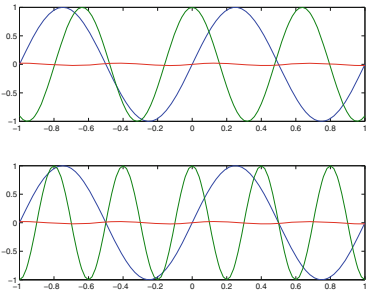
```
t = -1:0.02:1;
x = sin(2*pi*t);
z1 = cos(pi*pi*t);
z2 = cos(5*pi*t);
v1 = (rand(20, size(x, 1)+1)) * [x;z1];
v1 = v1 + randn(size(v1))*0.01;
v2 = (rand(20, size(x, 1)+1)) * [x;z2];
v2 = v2 + randn(size(v2))*0.01;
v1 = [v1; 0.02*sin(3.6*pi*t)];
v2 = [v2; 0.02*sin(3.6*pi*t)];
```

In words, v1 and v2 (first and second views in Fig. 2b respectively) are two views generated by randomly projecting two ground truths, `[x;z1]` and `[x;z2]`, to 20 dimensions spaces. The first component (blue curve in Fig. 2a) of the two ground truths are shared latent space and their second components (green curves in first and second ground truth of Fig. 2a respectively) are individual private latent spaces. The two views are then added to Gaussian noise with standard deviation 0.01 and correlated noise on their last dimension (both views have 21 dimensions now).

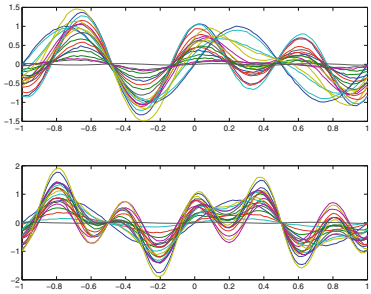
In the experiment, we adopt the Hyperbolic Tangent ($tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$) as the activation function. Three hidden neurons are used as latent spaces, one represents the shared space and the other two represent the private ones. Features of the original data are evenly corrupted for denoising.

The result of our method is depicted in Fig. 2d, where the first graph employs modified autoencoder while the second one is modified denoising autoencoder. The shared latent factor is in blue, the private latent factor of first view is in green and private latent factor of second view is in red. The denoising autoencoder generates more robust latent spaces than those from the autoencoder.

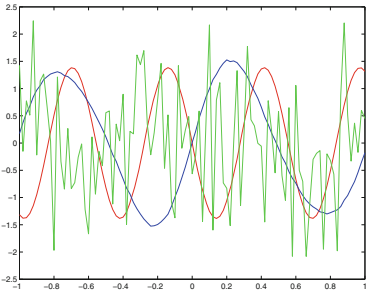
As expected, Canonical Correlation Analysis (Fig. 2c) extracts the true shared signal (in blue) and the correlated noise (in red), while fails to discover the two private signals. Notice that both true recovered signals are scaled and the correlated noise is even inverted. The reason is that we can multiply a number



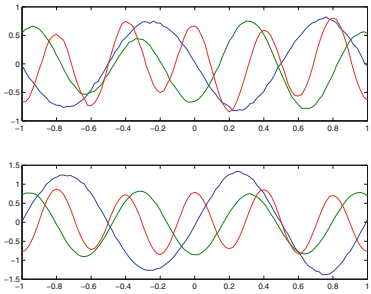
(a) Two ground truths (shared signals in blue, private signals in green, correlated noise in red)



(b) Two 21D views



(c) Result of CCA



(d) Result of our approach (orthogonal non-denoising and denoising autoencoder)

Fig. 2. Latent spaces recovered on synthetic data (Color figure online).

and its multiplicative inverse respectively to the latent spaces and corresponding coefficient to attain same product.

The experiment confirms our approach is able to effectively learn robust shared and private latent spaces, while CCA is sensitive to noise. Moreover, we do not need to use PCA like previous work [5][1]. Because the local minimum differs due to the random initialization of weights, methods of optimization, etc., Fig. 2d may vary slightly in repeat experiments [25].

4.2 Real-World Dataset

We applied our model to the PASCAL VOC'07 data set [26] for multi-label object classification, i.e., an image may contain more than one object label. Through the experiment, we compare the results from using only images, tags, and their combinations to demonstrate multi-view learning algorithms are capable of using

[1] With PCA, CCA can also perform well.

Table 1. Mean and variance of AP of different approaches (best results of classification for each class in bold).

	aeroplane	**bicycle**	**bird**	**boat**	**bottle**	**bus**	**car**
Images	0.596 ± 0.012	0.367 ± 0.028	0.277 ± 0.044	0.568 ± 0.018	0.127 ± 0.021	0.319 ± 0.023	0.613 ± 0.011
Tags	0.734 ± 0.016	0.539 ± 0.012	0.688 ± 0.003	0.495 ± 0.016	$\mathbf{0.237} \pm 0.017$	0.430 ± 0.016	0.591 ± 0.003
MKL	$\mathbf{0.847} \pm 0.031$	0.532 ± 0.030	0.711 ± 0.027	0.596 ± 0.044	0.210 ± 0.032	0.569 ± 0.026	0.694 ± 0.025
SVM2K(0)	0.584 ± 0.157	0.360 ± 0.271	0.576 ± 0.356	0.290 ± 0.290	0.117 ± 0.098	0.116 ± 0.033	0.431 ± 0.115
SVM2K(1)	0.682 ± 0.132	0.385 ± 0.377	0.113 ± 0.356	$\mathbf{0.796} \pm 0.122$	0.080 ± 0.038	0.133 ± 0.098	0.579 ± 0.292
ODAE	0.837 ± 0.009	$\mathbf{0.548} \pm 0.008$	$\mathbf{0.725} \pm 0.005$	0.682 ± 0.020	0.205 ± 0.024	$\mathbf{0.588} \pm 0.029$	$\mathbf{0.714} \pm 0.013$

	cat	**chair**	**cow**	**diningtable**	**dog**	**horse**	**motorbike**
Images	0.369 ± 0.010	$\mathbf{0.364} \pm 0.006$	0.225 ± 0.079	$\mathbf{0.291} \pm 0.102$	0.220 ± 0.019	0.607 ± 0.040	0.394 ± 0.039
Tags	0.715 ± 0.009	0.174 ± 0.004	0.471 ± 0.014	0.106 ± 0.012	0.678 ± 0.006	0.775 ± 0.004	0.607 ± 0.006
MKL	0.722 ± 0.018	0.296 ± 0.018	0.513 ± 0.037	0.193 ± 0.017	$\mathbf{0.689} \pm 0.022$	$\mathbf{0.815} \pm 0.014$	$\mathbf{0.660} \pm 0.032$
SVM2K(0)	0.231 ± 0.307	0.095 ± 0.013	0.089 ± 0.011	0.074 ± 0.034	0.137 ± 0.056	0.068 ± 0.013	0.521 ± 0.305
SVM2K(1)	0.366 ± 0.199	0.311 ± 0.164	0.187 ± 0.170	0.070 ± 0.028	0.486 ± 0.293	0.516 ± 0.353	0.258 ± 0.222
ODAE	$\mathbf{0.725} \pm 0.012$	0.347 ± 0.011	$\mathbf{0.521} \pm 0.015$	0.219 ± 0.014	0.665 ± 0.009	0.760 ± 0.015	0.624 ± 0.021

	person	**pottedplant**	**sheep**	**sofa**	**train**	**tvmonitor**
Images	0.689 ± 0.016	0.115 ± 0.018	0.163 ± 0.026	0.220 ± 0.016	0.589 ± 0.035	0.230 ± 0.014
Tags	0.686 ± 0.002	0.329 ± 0.012	0.592 ± 0.031	0.180 ± 0.006	0.811 ± 0.005	0.378 ± 0.013
MKL	$\mathbf{0.782} \pm 0.028$	$\mathbf{0.388} \pm 0.047$	0.584 ± 0.043	0.225 ± 0.026	$\mathbf{0.845} \pm 0.017$	0.395 ± 0.022
SVM2K(0)	0.643 ± 0.227	0.053 ± 0.003	0.599 ± 0.329	$\mathbf{0.510} \pm 0.404$	0.188 ± 0.156	$\mathbf{0.711} \pm 0.299$
SVM2K(1)	0.739 ± 0.038	0.256 ± 0.308	0.126 ± 0.115	0.151 ± 0.064	0.290 ± 0.214	0.197 ± 0.115
ODAE	0.749 ± 0.015	0.350 ± 0.012	$\mathbf{0.644} \pm 0.008$	0.244 ± 0.026	0.838 ± 0.002	0.435 ± 0.006

information from different views. We also provide a comparison of our method to other methods, and test the sensitivity of our model.

The data set contains around 10,000 images of 20 different categories of object with standard train/test sets provided. The dataset also provides tags for each image: textual descriptions of the images. A total of 804 tags, which appear at least 8 times, form the bag-of-words vectors (bit-based) of a view. 15 image features are chosen from [15] as visual descriptors: local SIFT features [27]; local hue histograms [28] (both were computed on a dense multi-scale grid and on regions found with a Harris interest-point detector); global color histograms over RGB, HSV, and LAB color spaces; the above histogram image representations computed over a 3×1 horizontal decomposition of the image; and GIST descriptor [29], which roughly encodes the image layout. This produces two views: a visual modality with 37,152 features; and a textual modality with 804 features.

The default experiment setup used 6 SVMs (sigmoid kernel) with different C parameters[2] for classification. Table 1 reports the mean and variance of the average precision (AP) values from these SVMs. For the single modality image classification ("Images" in Table 1), we report AP using the single visual feature that performed best for each class.

In Multiple Kernel Learning (MKL), we computed textual kernel $k_t(\cdot,\cdot)$ and visual kernels $k_v(\cdot,\cdot)$ following [15]. Because the final kernel $k_f(\cdot,\cdot) = d_v k_v(\cdot,\cdot) + d_t k_t(\cdot,\cdot)$, where $d_v, d_t > 0$ and $d_v + d_t = 1$, is a convex combination, the d_v and d_t

[2] C from $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$.

were chosen by grid search with step 0.1 and $C = 1$. Instead of original features, kernels $k_f(\cdot, \cdot)$ were then used in place of the original features.

We tested two variants of SVM2K, both containing two views, since SVM2K [17] only supports two-view learning. SVM2K(0) concatenates all visual features into one view, with the other view comprising the textual features; SVM2K(1) uses only the 2 SIFT-based features. SVM2K contains 4 penalty parameters: we set the penalty value for the first SVM to 0.2; for the second SVM to 0.1; tolerance for the synthesis to 0.01; and varied the penalty value for synthesis in the same way as previously described for linear SVM parameter C.

Our orthogonal denoising autoencoder for multi-view learning (ODAE) uses the sigmoid function ($y = \frac{1}{1+e^{-x}}$) as the activation function. We reduplicate the data 10 times and randomly corrupted 10 % of total features each time for denoising. Random search was used to find the optimal numbers of neurons for each hidden views [30].

Generally speaking, multi-view learning approaches for object classification outperform those using only the visual or textual view. SVM2K methods have the worst performance and are highly unstable among all the multi-view learning approaches, since they can only accept 2 views. Specifically, the visual view of SVM2K(0) tends to overfit while that of SVM2K(1) tends to underfit. ODAE performs best for 7 classes and second best for another 10 classes (slightly worse).

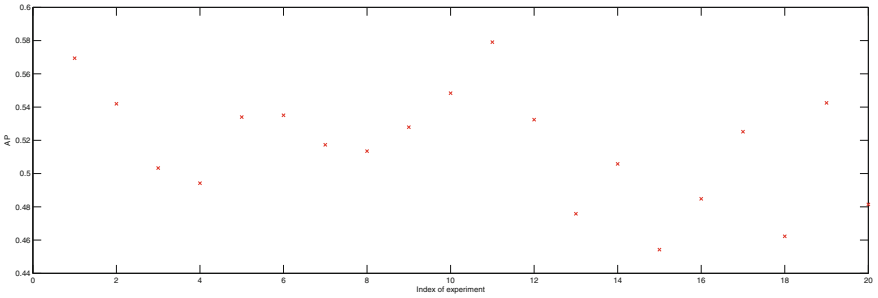


Fig. 3. AP values for 20 experiments with random hyperparameters (max 300 SGD iterations). The number of hidden nodes in each experiment are uniformly drawn from: $[3, 40]$ for the shared view, $[500, 700]$ for the textual view, and $[50, 200]$ for all visual views with the exception of Harris hue ranges, which were drawn from $[50, 100]$.

We used random search [30] to select reasonable values for the number of hidden neurons for each view[3]. Figure 3 plots AP values for different parameter configurations of an orthogonal autoencoder, demonstrating the relative robustness of the method with respect to the hyperparameters. The best configuration found was: 29 nodes of hidden layer for the shared view, 672 for the textual view, and $[99, 148, 149, 143, 74, 51, 113, 121, 96, 164, 113, 91, 148, 68, 56]$ for the visual views.

[3] Because of pages limit, we can not provide more details.

5 Conclusions

In this paper, we modify the basic autoencoder for multi-view learning to enable private latent spaces to be absolutely orthogonal to each other while simultaneously encouraging the shared latent space to be orthogonal to private latent spaces as well. Inheriting from the denoising autoencoder, our model is able to learn robust features, i.e., features that are strongly resistant to noise. We also extend back-propagation algorithm elegantly to train the model, which means our model is exempt from extra complex optimization tricks.

Acknowledgement. The research was supported by the Irish Research Council (IRCSET) under Grant Number GOIPG/2013/330. The authors wish to acknowledge the DJEI/DES/SFI/HEA Irish Centre for High-End Computing (ICHEC) for the provision of computational facilities and support. Amen.

References

1. Sun, S.: A survey of multi-view machine learning. Neural Comput. Appl. **23**(7–8), 2031–2038 (2013)
2. Xu, C., Tao, D., Xu, C.: A survey on multi-view learning. arXiv preprint arXiv:1304.5634 (2013)
3. Dasgupta, S., Littman, M.L., McAllester, D.: Pac generalization bounds for co-training. Adv. Neural Inf. Process. Syst. **1**, 375–382 (2002)
4. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
5. Jia, Y., Salzmann, M., Darrell, T.: Factorized latent spaces with structured sparsity. In: Advances in Neural Information Processing Systems, pp. 982–990 (2010)
6. Salzmann, M., Ek, C.H., Urtasun, R., Darrell, T.: Factorized orthogonal latent spaces. In: International Conference on Artificial Intelligence and Statistics, pp. 701–708 (2010)
7. Bengio, Y., Goodfellow, I.J., Courville, A.: Deep learning. Book in preparation for MIT Press (2015)
8. Memisevic, R.: On multi-view feature learning. arXiv preprint arXiv:1206.4609 (2012)
9. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Eleventh Annual Conference On Computational Learning Theory, pp. 92–100. ACM (1998)
10. Yu, S., Krishnapuram, B., Rosales, R., Rao, R.B.: Bayesian co-training. J. Mach. Learn. Res. **12**, 2649–2680 (2011)
11. Muslea, I., Minton, S., Knoblock, C.A.: Active learning with multiple views. J. Artif. Intell. Res. **27**, 203–233 (2006)
12. Gönen, M., Alpaydın, E.: Multiple kernel learning algorithms. J. Mach. Learn. Res. **12**, 2211–2268 (2011)
13. Rakotomamonjy, A., Bach, F., Canu, S., Grandvalet, Y.: More efficiency in multiple kernel learning. In: Proceedings of the 24th International Conference On Machine Learning, pp. 775–782. ACM (2007)
14. Akaho, S.: A kernel method for canonical correlation analysis. arXiv preprint cs/0609071 (2006)

15. Guillaumin, M., Verbeek, J., Schmid, C.: Multimodal semi-supervised learning for image classification. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 902–909. IEEE (2010)
16. Sun, S., Hardoon, D.R.: Active learning with extremely sparse labeled examples. Neurocomputing **73**(16), 2980–2988 (2010)
17. Farquhar, J., Hardoon, D., Meng, H., Shawe-taylor, J.S., Szedmak, S.: Two view learning: Svm-2k, theory and practice. In: Advances in Neural Information Processing Systems, pp. 355–362 (2005)
18. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013)
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
20. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. **29**(6), 82–97 (2012)
21. Collobert, R., Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th International Conference On Machine Learning, pp. 160–167. ACM (2008)
22. Liu, W., Tao, D., Cheng, J., Tang, Y.: Multiview hessian discriminative sparse coding for image annotation. Comput. Vis. Image Underst. **118**, 50–60 (2014)
23. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H., et al.: Greedy layer-wise training of deep networks. Adv. Neural Inf. Process. Syst. **19**, 153 (2007)
24. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning, pp. 1096–1103. ACM (2008)
25. LeCun, Y.A., Bottou, L., Orr, G.B., Müller, K.-R.: Efficient BackProp. In: Montavon, G., Orr, G.B., Müller, K.-R. (eds.) Neural Networks: Tricks of the Trade, 2nd edn. LNCS, vol. 7700, pp. 9–48. Springer, Heidelberg (2012)
26. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
27. Lowe, D.G.: Object recognition from local scale-invariant features. In: The proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157. IEEE (1999)
28. van de Weijer, J., Schmid, C.: Coloring local feature extraction. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 334–348. Springer, Heidelberg (2006)
29. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. Int. J. Comput. Vis. **42**(3), 145–175 (2001)
30. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**(1), 281–305 (2012)