Deep Generalized Canonical Correlation Analysis

Adrian Benton[‡*] **Huda Khayrallah**[◇] **Biman Gujral**[◇†]
Dee Ann Reisinger[◇] **Sheng Zhang**[◇] and **Raman Arora**[◇]
[‡]Bloomberg LP
[◇]Johns Hopkins University
abenton10@bloomberg.net

Abstract

We present Deep Generalized Canonical Correlation Analysis (DGCCA) – a method for learning nonlinear transformations of arbitrarily many views of data, such that the resulting transformations are maximally informative of each other. While methods for nonlinear two-view representation learning (Deep CCA, (Andrew et al., 2013)) and linear many-view representation learning (Generalized CCA (Horst, 1961)) exist, DGCCA combines the flexibility of nonlinear (deep) representation learning with the statistical power of incorporating information from many sources, or views. We present the DGCCA formulation as well as an efficient stochastic optimization algorithm for solving it. We learn and evaluate DGCCA representations for three downstream tasks: phonetic transcription from acoustic & articulatory measurements, recommending hashtags, and recommending friends on a dataset of Twitter users.

1 Introduction

Multiview representation learning refers to settings where one has access to many "views" of data at train time. Views often correspond to different modalities about examples: a scene represented as a series of audio and image frames, a social media user characterized by the messages they post and who they friend, or a speech utterance and the configuration of the speaker's tongue. Multiview techniques learn a representation of data that captures the sources of variation common to all views.

Multiview representation techniques are attractive since a representation that is able to explain many views of the data is more likely to capture meaningful variation than a representation that is a good fit for only one of the views. These methods are often based on canonical correlation analysis (CCA), a classical statistical technique proposed by Hotelling (1936). CCA-based techniques cannot currently model nonlinear relationships between arbitrarily many views. Either they are able to model variation across many views, but can only learn linear mappings to the shared space (Horst, 1961), or they can learn nonlinear mappings, but they cannot be applied to data with more than two views using existing techniques based on Kernel CCA (Hardoon et al., 2004) and Deep CCA (Andrew et al., 2013).

We present Deep Generalized Canonical Correlation Analysis (DGCCA). DGCCA learns a shared representation from data with arbitrarily many views and simultaneously learns nonlinear mappings from each view to this shared space.Our main methodological contribution is the derivation of the gradient update for the Generalized Canonical Correlation Analysis (GCCA) objective (Horst, 1961).[1] We evaluate DGCCA-learned representations on three downstream tasks: (1) phonetic transcription from aligned speech & articulatory data, (2) Twitter hashtag and (3) friend recommendation from six text and network feature views. We find that features learned by DGCCA outperform linear multiview techniques on these tasks.

2 Prior Work

Some of the most successful techniques for multiview representation learning are based on canonical correlation analysis and its extension to the nonlinear and many view settings (Wang et al., 2015b,a).

[*] Work done while at Johns Hopkins University.

[†] Now at Google.

[1] See `https://bitbucket.org/adrianbenton/dgcca-py3` for an implementation of DGCCA along with data from the synthetic experiments.

Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019), pages 1–6
Florence, Italy, August 2, 2019.

Canonical correlation analysis (CCA) Canonical correlation analysis (CCA) (Hotelling, 1936) is a statistical method that finds maximally correlated linear projections of two random vectors. It is a fundamental multiview learning technique. Given two input views, $X_1 \in \mathbb{R}^{d_1}$ and $X_2 \in \mathbb{R}^{d_2}$, with covariance matrices, Σ_{11} and Σ_{22}, respectively, and cross-covariance matrix Σ_{12}, CCA finds directions that maximize the correlation between these two views:

$$(u_1^*, u_2^*) = \underset{u_1 \in \mathbb{R}^{d_1}, u_2 \in \mathbb{R}^{d_2}}{\operatorname{argmax}} \quad corr(u_1^\top X_1, u_2^\top X_2)$$

Since this formulation is invariant to affine transformations of u_1 and u_2, we can write it as the following constrained optimization formulation:

$$(u_1^*, u_2^*) = \underset{u_1 \in \mathbb{R}^{d_1}, u_2 \in \mathbb{R}^{d_2}}{\operatorname{argmax}} \quad u_1^\top \Sigma_{12} u_2 \qquad (1)$$

such that $u_1^\top \Sigma_{11} u_1 = u_2^\top \Sigma_{22} u_2 = 1$. This technique has two limitations that have led to significant extensions: (1) it is limited to learning representations that are *linear* transformations of the data in each view, and (2) it can only leverage two input views.

Deep Canonical Correlation Analysis (DCCA) Deep CCA (DCCA) (Andrew et al., 2013) addresses the first limitation by finding maximally correlated *non-linear* transformations of two vectors. It passes each of the input views through neural networks and performs CCA on the outputs.

Let us use $f_1(X_1) = Z_1$ and $f_2(X_2) = Z_2$ to represent the network outputs. The weights, W_1 and W_2, of these networks are trained through standard backpropagation to maximize the CCA objective:

$$(u_1^*, u_2^*, W_1^*, W_2^*) = \underset{u_1, u_2}{\operatorname{argmax}}\, corr(u_1^\top Z_1, u_2^\top Z_2)$$

Generalized Canonical Correlation Analysis (GCCA) Generalized CCA (GCCA) (Horst, 1961) addresses the limitation on the number of views. It solves the optimization problem in Equation 2, finding a shared representation G of J different views, where N is the number of data points, d_j is the dimensionality of the jth view, r is the dimensionality of the learned representation, and $X_j \in \mathbb{R}^{d_j \times N}$ is the data matrix for the jth view.

$$\begin{aligned} &\underset{U_j \in \mathbb{R}^{d_j \times r}, G \in \mathbb{R}^{r \times N}}{\text{minimize}} \quad \sum_{j=1}^{J} \|G - U_j^\top X_j\|_F^2 \qquad (2) \\ &\text{subject to} \qquad GG^\top = I_r \end{aligned}$$

Solving GCCA requires finding an eigendecomposition of an $N \times N$ matrix, which scales quadratically with sample size and leads to memory constraints.

Unlike CCA and DCCA, which only learn projections or transformations on each of the views, GCCA also learns a view-independent representation G that best reconstructs all of the view-specific representations simultaneously. The key limitation of GCCA is that it can only learn *linear* transformations of each view.

3 Deep Generalized Canonical Correlation Analysis (DGCCA)

We present Deep GCCA (DGCCA): a multi-view representation learning technique that benefits from the expressive power of deep neural networks and can leverage statistical strength from more than two views. More fundamentally, Deep CCA and Deep GCCA have very different objectives and optimization problems, and it is not immediately clear how to extend deep CCA to more than two views.

DGCCA learns a nonlinear map for each view in order to maximize the correlation between the learned representations across views. In training, DGCCA passes the input vectors in each view through multiple layers of nonlinear transformations and backpropagates the gradient of the GCCA objective with respect to network parameters to tune each view's network. The objective is to train networks that reduce the GCCA reconstruction error among their outputs. New data can be projected by feeding each view through its respective network.

Problem Consider a dataset of J views and let $X_j \in \mathbb{R}^{d_j \times N}$ denote the j^{th} input matrix.The network for the j^{th} view consists of K_j layers. Assume, for simplicity, that each layer in the j^{th} view network has c_j units with a final (output) layer of size o_j. The output of the k^{th} layer for the j^{th} view is $h_k^j = s(W_k^j h_{k-1}^j)$, where $s : \mathbb{R} \to \mathbb{R}$ is a nonlinear activation function and $W_k^j \in \mathbb{R}^{c_k \times c_{k-1}}$ is the weight matrix for the k^{th} layer of the j^{th} view network. We denote the output of the final layer as $f_j(X_j)$.

DGCCA can be expressed as the following optimization problem: find weight matrices $W^j = \{W_1^j, \ldots, W_{K_j}^j\}$ defining the functions f_j, and linear transformations U_j (of the output of the j^{th}

network), for $j = 1, \ldots, J$, that

$$\underset{U_j \in \mathbb{R}^{o_j \times r}, G \in \mathbb{R}^{r \times N}}{\text{minimize}} \sum_{j=1}^{J} \|G - U_j^\top f_j(X_j)\|_F^2 \quad (3)$$

where $G \in \mathbb{R}^{r \times N}$ is the shared representation we are interested in learning, subject to $GG^\top = I_r$.

Optimization We solve the DGCCA optimization problem using stochastic gradient descent (SGD) with mini-batches. In particular, we estimate the gradient of the DGCCA objective in Equation 3 on a mini-batch of samples that is mapped through the network and use backpropagation to update the weight matrices, W^j's. Because DGCCA optimization is a constrained optimization problem, it is not immediately clear how to perform projected gradient descent with backpropagation. Instead, we characterize the objective function of the GCCA problem at an optimum and compute its gradient with respect to the inputs to GCCA (i.e. with respect to the network outputs), which are subsequently backpropagated through the network to update W^js.

Gradient Derivation The solution to the GCCA problem is given by solving an eigenvalue problem. In particular, define $C_{jj} = f(X_j)f(X_j)^\top \in \mathbb{R}^{o_j \times o_j}$, to be the scaled empirical covariance matrix of the j^{th} network output, and $P_j = f(X_j)^\top C_{jj}^{-1} f(X_j) \in \mathbb{R}^{N \times N}$ to be the corresponding projection matrix that whitens the data; note that P_j is symmetric and idempotent. We define $M = \sum_{j=1}^{J} P_j$. Since each P_j is positive semi-definite, so is M. One can then check that the rows of G are the top r (orthonormal) eigenvectors of M, and $U_j = C_{jj}^{-1} f(X_j) G^\top$. Thus, at the minimum of the objective, we can rewrite the reconstruction error as follows:

$$\sum_{j=1}^{J} \|G - U_j^\top f_j(X_j)\|_F^2 = rJ - \text{Tr}(GMG^\top)$$

Minimizing the GCCA objective (with respect to the weights of the neural networks) means maximizing $\text{Tr}(GMG^\top)$, which is the sum of eigenvalues $L = \sum_{i=1}^{r} \lambda_i(M)$. Taking the derivative of L with respect to each output layer $f_j(X_j)$ gives:

$$\frac{\partial L}{\partial f_j(X_j)} = 2U_jG - 2U_jU_j^\top f_j(X_j)$$

Thus, the gradient is the difference between the r-dimensional auxiliary representation G embedded into the subspace spanned by the columns of U_j (the first term) and the projection of the actual data in $f_j(X_j)$ onto said subspace (the second term).

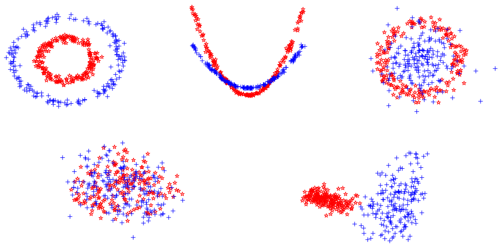

Figure 1: Three synthetic views are displayed in the top row, and the bottom rows displays the matrix G learned by GCCA (left) and DGCCA (right).

4 Experiments

4.1 Synthetic Multiview Mixture Model

We apply DGCCA to a small synthetic data set to show how it preserves the generative structure of data sampled from a multiview mixture model. The three views of the data we use for this experiment are plotted in the top row of Figure 1. Points that share the same color across different views are sampled from the same mixture component.

Importantly, in each view, there is no linear transformation of the data that separates the two mixture components. This point is reinforced by Figure 1 (bottom left), which shows the two-dimensional representation G learned by applying linear GCCA to the data in plotted in the top row. The learned representation completely loses the structure of the data. In contrast, the representation G learned by DGCCA (bottom right) largely preserves the structure of the data, even after projection onto the first coordinate. In this case, the input neural networks for DGCCA had three hidden layers with ten units each, with randomly-initialized weights.

4.2 Phoneme Classification

We perform experiments on the University of Wisconsin X-ray Microbeam Database (XRMB) (Westbury, 1994), a collection of acoustic & articulatory recordings along with phonemic labels. We present phoneme classification results on the acoustic vectors projected using DCCA, GCCA, and DGCCA. Acoustic and articulatory data are set as the first two views and phoneme labels are set as the third view for GCCA and DGCCA, and

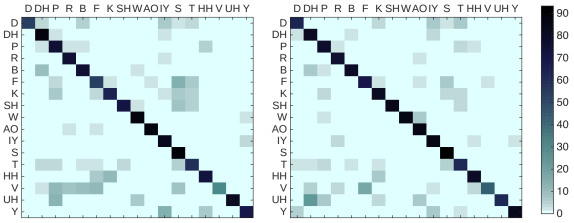

Figure 2: Confusion matrices over phonemes for speaker-dependent GCCA (left) and DGCCA (right).

Table 1: KNN phoneme classification performance.

	Cross-Speaker		Speaker-dependent	
Method	Dev/Test Acc	Rec Err	Dev/Test Acc	Rec Err
MFCC	48.9/49.3		66.3/66.2	
DCCA	45.4/46.1		65.9/65.8	
GCCA	49.6/50.2	40.7	69.5/69.8	40.4
DGCCA	**53.8/54.2**	**35.9**	**72.6/72.3**	**20.5**

K-nearest neighbor classification (Cover and Hart, 1967) is then run on the projected result.

Data We use the same split of the data as Arora and Livescu (2014). To limit experiment runtime, we use a subset of speakers for our experiments. We run a set of *cross-speaker* experiments using the male speaker JW11 for training and two splits of JW24 for tuning and testing. We also perform parameter tuning for the third view with 5-fold cross validation using a *single speaker*, JW11. For both experiments, we use acoustic and articulatory measurements as the two views in DCCA. Following the pre-processing in Andrew et al. (2013), we get 273 and 112 dimensional feature vectors for the first and second view respectively. Each speaker has ~50,000 frames. For the third view in GCCA and DGCCA, we use 39-dimensional one-hot vectors corresponding to the labels for each frame, following Arora and Livescu (2014).

Parameters We use a fixed network size and regularization for the first two views, each containing three hidden layers. Hidden layers for the acoustic view were all width 1024, and layers in the articulatory view all had width 512 units. L2 penalty constants of 0.0001 and 0.01 were placed on the acoustic and articulatory view networks, with 0.0005 on the label view. The output layer dimension of each network is set to 30 for DCCA and DGCCA. For the 5-fold speaker-dependent experiments, we performed a grid search for the network sizes in $\{128, 256, 512, 1024\}$ and covariance matrix regularization in $\{10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}\}$ for the third view in each fold. We fix the hyperparameters for these experiments optimizing the networks with minibatch stochastic gradient descent with a step size of 0.005, and a batch size of 2,000.

Results DGCCA improves upon both the linear multiview GCCA and the non-linear 2-view DCCA for both the cross-speaker and speaker-dependent cross-validated tasks (Table 1). In addition to accuracy, we examine the reconstruction error (i.e. the objective in Equation 3) obtained from the objective in GCCA and DGCCA. The sharp improvement in reconstruction error shows that a non-linear algorithm can better model the data.

In this experimental setup, DCCA underperforms the baseline of simply running KNN on the original acoustic view. Prior work considered the output of DCCA stacked on to the central frame of the original acoustic view (39 dimensions). This poor performance, in the absence of original features, indicates that it was not able to find a more informative projection than original acoustic features based on correlation with the articulatory view within the first 30 dimensions.

To highlight the improvements of DGCCA over GCCA, Figure 2 presents a subset of the the confusion matrices on speaker-dependent test data. We observe large improvements in the classification of D, F, K, SH, V and Y. For instance, DGCCA rectifies the frequent misclassification of V as P, R and B by GCCA. In addition, commonly incorrect classification of phonemes such as S and T is corrected by DGCCA, which enables better performance on other voiceless consonants such as like F, K and SH. Vowels are classified with almost equal accuracy by both the methods.

4.3 Hashtag & Friend Recommendation

Linear multiview techniques are effective at recommending hashtag and friends for Twitter users (Benton et al., 2016). In this experiment, six views of a Twitter user were constructed by applying principal component analysis (PCA) to the bag-of-words representations of (1) tweets posted by the ego user, (2) other mentioned users, (3) their friends, and (4) their followers, as well as one-hot encodings of the local (5) friend and (6) follower networks. We learn and evaluate DGCCA models on identical training, development, and test sets as Benton et al. (2016), and evaluate

Table 2: Dev/test performance at Twitter friend and hashtag recommendation tasks.

	Friend		Hashtag	
Algorithm	P@1K	R@1K	P@1K	R@1K
PCA[T+N]	**.445/.439**	**.149/.147**	.011/.008	.312/.290
GCCA[T]	.244/.249	.080/.081	.012/.009	.351/.326
GCCA[T+N]	.271/.276	.088/.089	**.012/.010**	.359/.334
DGCCA[T+N]	.297/.268	.099/.090	**.013/.010**	**.385/.373**
WGCCA[T]	.269/.279	.089/.091	.012/.009	.357/.325
WGCCA[T+N]	.376/.364	.123/.120	.013/.009	.360/.346

the DGCCA representations on macro precision at 1,000 (P@1K) and recall at 1,000 (R@1K) for the hashtag and friend recommendation tasks described there.

We trained 40 different DGCCA model architectures, each with identical network architectures across views, where the width of the hidden and output layers, c_1 and c_2, for each view are drawn uniformly from $[10, 1000]$, and the auxiliary representation width r is drawn uniformly from $[10, c_2]$.[2] All networks used rectified linear units in the hidden layer, and were optimized with Adam (Kingma and Ba, 2014) for 200 epochs. Networks were trained on 90% of 102,328 Twitter users, with 10% of users used as a tuning set to estimate held-out reconstruction error for model selection. We report development and test results for the best performing model on the downstream task development set. The learning rate was set to 10^{-4} with regularization of $\ell_1 = 10^{-2}$, $\ell_2 = 10^{-4}$.

Table 2 displays the precision and recall at 1000 recommendations of DGCCA compared to PCA[T+N] (PCA applied to concatenation of text and network view feature vectors), linear GCCA applied to the four text views *[T]*, and all text and network views *[T+N]*, along with a GCCA variant with discriminative view weighting (WGCCA). We learned PCA, GCCA, and WGCCA representations of width $r \in [10, 1000]$, and select embeddings based on development set R@1K.

There are several points to note: First is that DGCCA outperforms linear methods at hashtag recommendation by a wide margin in terms of recall. This is exciting because this task was shown to benefit from incorporating more than just two views from Twitter users. These results suggest that a nonlinear transformation of the input views can yield additional gains in performance. In addition, WGCCA models sweep over every possible weighting of views with weights in $\{0, 0.25, 1.0\}$. The fact that DGCCA is able to outperform WGCCA at hashtag recommendation is encouraging, since WGCCA has much more freedom to discard uninformative views. As noted in Benton et al. (2016), only the friend network view was useful for learning representations for friend recommendation (corroborated by performance of PCA applied to friend network view), so it is unsurprising that DGCCA, when applied to all views, cannot compete with WGCCA representations learned on the single useful friend network view.

5 Discussion

There has also been strong work outside of CCA-related methods to combine nonlinear representation and learning from multiple views. Kumar et al. (2011) outlines two main approaches to learn a joint representation from many views: either by (1) explicitly maximizing similarity/correlation between view pairs (Masci et al., 2014; Rajendran et al., 2015) or by (2) alternately optimizing a shared, "consensus" representation and view-specific transformations (Kumar et al., 2011; Xiaowen, 2014; Sharma et al., 2012). Unlike the first class of methods, the complexity of solving DGCCA does not scale quadratically with number of views, nor does it require a privileged pivot view (G is learned). Unlike methods that estimate a "consensus" representation, DGCCA admits a globally optimal solution for both the view-specific projections $U_1 \ldots U_J$, and the shared representation G. Local optima arise in the DGCCA objective only because we are also learning nonlinear transformations of the input views.

We present DGCCA, a method for non-linear multiview representation learning from an arbitrary number of views. We show that DGCCA clearly outperforms prior work in phoneme recognition when using labels as a third view, and can successfully exploit multiple views to learn Twitter user representations useful for downstream tasks, such as hashtag recommendation. To date, CCA-style multiview learning techniques were either restricted to learning representations from no more than two views, or strictly linear transformations of the input views. This work overcomes these limitations.

[2] We only consider architectures with single-hidden-layer networks identical across views so as to avoid a fishing expedition. If DGCCA is an appropriate method for learning Twitter user embeddings, then it should require little architecture exploration.

References

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. 2013. Deep canonical correlation analysis. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1247–1255.

Raman Arora and Karen Livescu. 2014. Multi-view learning with supervision for transformed bottleneck features. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 2499–2503. IEEE.

Adrian Benton, Raman Arora, and Mark Dredze. 2016. Learning multiview embeddings of twitter users. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 14.

Thomas M Cover and Peter E Hart. 1967. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27.

David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664.

Paul Horst. 1961. Generalized canonical correlations and their applications to experimental data. *Journal of Clinical Psychology*, 17(4).

Harold Hotelling. 1936. Relations between two sets of variates. *Biometrika*, pages 321–377.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Abhishek Kumar, Piyush Rai, and Hal Daume. 2011. Co-regularized multi-view spectral clustering. In *Advances in neural information processing systems*, pages 1413–1421.

Jonathan Masci, Michael M Bronstein, Alexander M Bronstein, and Jürgen Schmidhuber. 2014. Multimodal similarity-preserving hashing. *IEEE transactions on pattern analysis and machine intelligence*, 36(4):824–830.

Janarthanan Rajendran, Mitesh M Khapra, Sarath Chandar, and Balaraman Ravindran. 2015. Bridge correlational neural networks for multilingual multimodal representation learning. *arXiv preprint arXiv:1510.03519*.

Abhishek Sharma, Abhishek Kumar, Hal Daume, and David W Jacobs. 2012. Generalized multiview analysis: A discriminative latent space. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2160–2167. IEEE.

Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. 2015a. On deep multi-view representation learning. In *Proc. of the 32nd Int. Conf. Machine Learning (ICML 2015)*.

Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. 2015b. Unsupervised learning of acoustic features via deep canonical correlation analysis. In *Proc. of the IEEE Int. Conf. Acoustics, Speech and Sig. Proc. (ICASSP'15)*.

John R. Westbury. 1994. X-ray microbeam speech production database user's handbook. In *Waisman Center on Mental Retardation & Human Development University of Wisconsin Madison, WI 53705-2280*.

Dong Xiaowen. 2014. *Multi-View Signal Processing and Learning on Graphs*. Ph.D. thesis, École Polytechnique Fédérale de Lausanne.