Survey of Multiview Representation Learning Techniques

Corbin Rosset*
Dept. of Computer Science
Johns Hopkins University
Baltimore, MD 21218
crosset2@jhu.edu

Neil Mallinar
Dept. of Computer Science
Dept. of Mathematics
Johns Hopkins University
Baltimore, MD 21218
nmallin1@jhu.edu

Akshay Srivatsan
Dept. of Computer Science
Dept. of Applied Math and Stats
Johns Hopkins University
Baltimore, MD 21218
asrivat1@jhu.edu

Abstract

This study surveys state of the art methods for scalable multi-view representation learning. We compared kernel and deep neural network techniques for canonical correlation analysis (CCA) to learn representations for each view that jointly maximize total correlation, as well as split autoencoders that attempt to learn a shared representation of the views. We compared these various methods' performance on the Wisconson X-Ray Micro Beam dataset in terms of classification accuracy and also quality of clusters in the new representation. All of our deep learning was implemented in Google's newly released TensorFlow library, which obviated the need for explicit gradient computation.

1 Introduction

It is common in modern data sets to have multiple views of data collected of a phenomenon, for instance, a set of images and their captions in text, or audio and video data of the same event. If there exist labels, the views are conditionally uncorrelated on them, and it is typically assumed that noise sources between views are uncorrelated so that the representations are discriminating of the underlying semantic content. To distinguish it from multi-modal learning, multi-view learning trains a model or classifier for each view, the application of which depends on what data is available at test time. Typically it is desirable to find representations for each view that are predictive of - and predicted by - the other views so that if one view is not available at test time, it can serve to denoise the other views, or serve as a soft supervisor providing pseudo-labels. The benefits of training on multiple views include reduced sample complexity for prediction scenarios [1], relaxed separation conditions for clustering [2], among others [3][4][5].

Canonical Correlation Analysis (CCA), developed by Hotelling in 1936, is a widely used procedure motivated by the belief that multiple sources of information might ease learnability of useful, low-dimensional representations of data. Specifically, CCA learns linear projections of vectors from two views that are maximally correlated [6]. While CCA is affine invariant, it can learn only linear projections of the two views, which is not sufficient for data that lies on a manifold. We apply a scalable[1] extension of kernel CCA (KCCA) with Gaussian and polynomial kernels. However, kernel methods scale poorly in both feature dimension (a dot product or norm is required) and number of samples (Gram matrix is polynomial in size). To remedy this, we apply deep methods, which enjoy learning a parametric form of a nonlinear target transformation. Those methods described in Arora

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

[1]Although the dimension of the XRMB data is on the order of 10^2, the number of samples ($\approx$ 50,000) would force the Gram matrix to exceed memory capacity

et al are split autoencoders (SplitAE)[2], deep CCA (DCCA), and deep CCA autoencoders (DCCAE) to the multiview data in XRMB for the purpose of speech recognition [7].

We investigate the enhancement of phonetic recognition by learning a transformation of acoustic feature vectors (the primary view) informed by articular data (a second view only available at training time). This experimental setting is motivated by previous work, where it was found that articulatory data is a qualified view of speech content that is usually only available in training settings [8].

2 Related Work

2.1 Canonical Correlation Analysis

CCA can be interpreted as an extension of principle component analysis to multiple data sets with the added constraint that the principle components learned in each subspace are maximally correlated. Concretely, given n observations (x_i, y_i), $x_i \in \mathbb{R}^{d_x}$ and $y_i \in \mathbb{R}^{d_y}$ comprising two views of data $\mathcal{X}$, $\mathcal{Y}$ described by an unknown joint distribution $\mathcal{D}$, find k pairs of vectors (u_j, v_j) of the same dimensions that

$$\max correlation(u_i^\top x, v_i^\top y) \tag{1}$$

subject to the constraint that (u_j, v_j) is uncorrelated to all other (u_r, v_r), $j \neq r$, $1 \leq j \leq k$. After finding the first pair (u_1, v_1), subsequent pairs are found via deflation of the views subject to the uncorrelation constraint above. Expanding the definition of correlation for the vectors u_i and v_i:

$$\max_{u_i \in \mathbb{R}^{d_x}, v_i \in \mathbb{R}^{d_y}} \frac{\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[u_i^\top x y^\top v_i\right]}{\sqrt{\mathbb{E}_x[u_i^\top x x^\top u_i]\mathbb{E}_y[v_i^\top y y^\top v_i]}} \tag{2}$$

Note that scaling of u_i or v_i does not change the objective, therefore the variance of u and v can be normalized to one. Expressed equivalently for all u and v to be learned,

$$\begin{aligned} \max_{U \in \mathbb{R}^{d_x \times k}, V \in \mathbb{R}^{d_y \times k}} \quad & \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\text{trace}\left(U^\top x y^\top V\right)\right] \\ \text{subject to} \quad & \mathbb{E}[U^\top x x^\top U] = I, \mathbb{E}[V^\top y y^\top V] = I \end{aligned} \tag{3}$$

Solving for u and v using the Lagrange multiplier method, linear CCA takes the form of a generalized eigenvalue problem, for which the solution is a product of (regularized) covariance[3] and cross-covariance matrices [9]: U is the top k left eigenvectors (sorted by decreasing eigenvalue) of $C_{xx}^{-1} C_{xy} C_{yy}^{-1} C_{yx}$, which when regularized becomes $(C_{xx} + r_x I)^{-1/2} C_{xy} (C_{yy} + r_y I)^{-1/2} C_{yx} (C_{xx} + r_x I)^{-1/2}$. The i'th column of V is then chosen as $\frac{C_{yy}^{-1} C_{yx} u_i}{\sqrt{\lambda_i}}$, which is regularized similarly.

A second interpretation of CCA is that optimal U and V projection matrices minimize the squared error of reconstructing x (respectively, y) given y (x). Hence, the optimization problems given in Equations 1, 2, 3, and 4 are all equivalent (for i = 1...k, where applicable).

$$\begin{aligned} \min_{U \in \mathbb{R}^{d_x \times k}, V \in \mathbb{R}^{d_y \times k}} \quad & \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\|U^\top x - V^\top y\|_2^2\right] \\ \text{subject to} \quad & \mathbb{E}[U^\top C_{xx} U] = I_k, \mathbb{E}[V^\top C_{yy} V] = I_k \end{aligned} \tag{4}$$

There are a number of settings to which CCA, and all correlation analysis variants, can be applied. The first is when all views are available at test and train time. The second is when only a non-empty subset of the views is available at test time. The third is the presence of labels combined with either of the first two settings.

[2] As described later, SplitAE reconstructs each view from a shared representation, which is not correlation analysis

[3] for numerical stability, a scaled identity matrix is added to a covariance matrix before it is inverted

2.2 Kernel CCA and Scalable KCCA

KCCA by Akaho et al generalizes the transformations learned by CCA to nonlinear functions living in a reproducing kernel hilbert space $\mathcal{H}$ (RKHS), making it suitable for manifold learning. Given $\mathcal{F}$ and $\mathcal{G}$ are function classes living in a reproducing kernel hilbert space reserved for $\mathcal{X}$ and $\mathcal{Y}$ respectively, the goal is to find two sets of k functions each, $\{f_1, ..., f_k\} : \mathbb{R}^{d_x} \to \mathbb{R}$, $\{g_1, ..., g_k\} : \mathbb{R}^{d_y} \to \mathbb{R}$ that for any i, f_i and g_i minimize the squared error of reconstructing each other in the RKHS. That is, given $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, f and g are maximally predictive and predictable of each other,

$$\begin{aligned} \min_{U \in \mathbb{R}^{d_x \times k}, V \in \mathbb{R}^{d_y \times k}} \quad & \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\|f_i(x) - g_i(y)\|_2^2 \right] \\ \text{subject to} \quad & \mathbb{E}[f_i(x) f_j(x)] = \delta_{ij}, \\ & \mathbb{E}[g_i(y) g_j(y)] = \delta ij \end{aligned} \tag{5}$$

for δ_{ij} an indicator random variable that assumes one if $i = j$ and zero otherwise. Equivalent interpretations similar to the above: maximize the correlation of $f(x)$ and $g(x)$ or maximize the covariance $\mathbb{E}_{(x,y)\sim\mathcal{D}} [f_i(x) g_i(y)]$ with the same constraints as Equation 5. As before, every pair of functions (f_j, g_j) is uncorrelated with all other pairs of functions for all k.

For n data vectors from each of $\mathcal{X}$ and $\mathcal{Y}$, the Gram matrices K_x and K_y from $\mathbb{R}^{n \times n}$ can be constructed[4] given a positive definite kernel function $\kappa(\cdot, \cdot) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ which is by construction equal to the dot product of its arguments in the feature space, $k(u, v) = \Phi(u)^\top \Phi(v)$ for a nonlinear feature map $\Phi : \mathbb{R}^d \to \mathcal{H}$. It is straightforward to show that each of the k functions f and g of each view can be represented as a linear combination

Takes $O(n^3)$ time and $O(n^2)$ space, rank-m approximations of the eigendecomposition are available to allow better scaling with n (arora and Livescu 2012; Savosyanov 2014

2.3 Split AutoEncoders

The work of Ngiam et al on multimodal autoencoders introduced a deep network architecture for learning a single representation from one or more views of data [10]. This was later followed up by Wang et al with applications to the Wisconsin X-Ray dataset, as well as a constructed multi-view version of the MNIST data set [7]. The goal of deep autoencoder architectures in multiview learning is to find some shared representation that minimizes the reconstruction error for all views. These deep autoencoders can take as input one or many views at the same time, depending on which are available for testing. For the sake of this paper, we will restrict ourselves to two views.

There are two architectures given by Ngiam et al that find a shared representation of two views. [10] The dataset used to train these have two views available at train time and one view available at test time, as is the case with the XRMB dataset that we are using in this paper.

$$\min_{\mathbf{W}_f, \mathbf{W}_g, \mathbf{W}_p, \mathbf{W}_q} \frac{1}{2} \sum_{i=1}^{N} (\|\mathbf{x}_i - \mathbf{p}(\mathbf{f}(\mathbf{x}_i))\|^2 + \|\mathbf{y}_i - \mathbf{q}(\mathbf{g}(\mathbf{y}_i))\|^2) \tag{6}$$

The first architecture trains using information from both views, minimizing Equation 6 which is the sum of the L_2 Norm of the reconstruction for both views. Both inputs are fed into the autoencoder separately and go through multiple hidden layers before being combined into a shared hidden layer representation of the two views. The decoders are then symmetrical to the encoders. At test time, all of the weights from the decoder of the view not available at test time are ignored, and the shared hidden layer representation calculated from the single view available is used.

$$\min_{\mathbf{W}_f, \mathbf{W}_p, \mathbf{W}_q} \frac{1}{2} \sum_{i=1}^{N} (\|\mathbf{x}_i - \mathbf{p}(\mathbf{f}(\mathbf{x}_i))\|^2 + \|\mathbf{y}_i - \mathbf{q}(\mathbf{f}(\mathbf{x}_i))\|^2) \tag{7}$$

[4] e.g. $[K_x]_{ij} = \kappa(x_i, x_j)$

Method	**% Accuracy**
CCA	
KCCA	
Split AE	
Deep CCA	

Table 1: Accuracy achieved with best parameters for each correlation analysis method implemented.

The second architecture, used by Wang et al, has a single encoder that takes as input the view available at train time. It then attempts to learn a shared representation and sets of weights that can reconstruct both views. The decoder for the view available at test time is symmetric to the encoder. The decoder for the view that's only available at train time can be a multilayer decoder or a single layer decoder that is experimentally tuned for number of layers and nodes.

2.4 Deep CCA

Deep CCA is a parametric technique to simultaneously learn nonlinear mappings for each view that are maximally correlated.

3 Experimental Methods

3.1 Kernel CCA and Scalable KCCA

3.2 Split AutoEncoders

We implemented the Split AutoEncoder using the second architecture described in Section 2.3. This is the design in which one view is used as input into the AutoEncoder, and a shared representation that can best reconstruct both views is found.

The encoder of the first view consists of three hidden layers of rectified linear units and a linear output layer (the shared representation hidden layer). The rectified linear hidden layers each have 500 nodes, while the output dimensionality is 50. In general the larger the width of the hidden layers, the better the AutoEncoders performed. The dimensionality was tuned over $\{30, 50, 70\}$ while the hidden layer width was tuned over values between 1 and 500. The decoder of the first view is symmetric to the encoder.

The decoder for the second view consists of two linear layers with dimensionality 50 and 75, respectively. The number of hidden layers was tuned between 1 and 3 and the width of each layer over values between 1 and 112, the latter being the dimensionality of the second view. We used the RMSProp Algorithm to train the AutoEncoders with a learning rate of 0.0001 and a decay of 0.99.

After training the AutoEncoders in an unsupervised fashion, we fine-tuned the encoder weights using a single-layer linear classifier at the bottleneck (being the shared representation hidden layer). The labels were one-hot encoded and the linear classifier was trained using the Adam Method of Stochastic Approximation, as proposed by Kingma & Ba, with a learning rate of 0.0001 [12].

All training was done in mini-batches of 100 instances each, while tuning accuracy was computed over the entire tuning dataset (full-batch).

3.3 Deep CCA

4 Results

4.1 Citations within the text

Citations within the text should be numbered consecutively. The corresponding number is to appear enclosed in square brackets, such as [1] or [2]-[5]. The corresponding references are to be listed in the same order at the end of the paper, in the **References** section. (Note: the standard BIBTEX style

Table 2: Sample table title

PART	DESCRIPTION
Dendrite	Input terminal
Axon	Output terminal
Soma	Cell body (contains cell nucleus)

`unsrt` produces this.) As to the format of the references themselves, any style is acceptable as long as it is used consistently.

4.2 Footnotes

Indicate footnotes with a number[5] in the text.

4.3 Figures

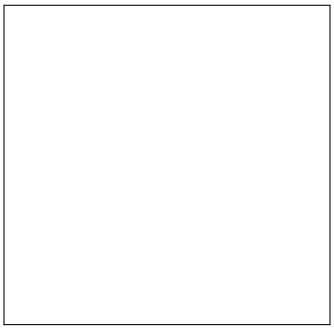

Figure 1: Sample figure caption.

4.4 Tables

All tables must be centered, neat, clean and legible. Do not use hand-drawn tables. The table number and title always appear before the table. See Table 2.

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

4.5 Margins in LaTeX

Most of the margin problems come from figures positioned by hand using `\special` or other commands. We suggest using the command `\includegraphics` from the graphicx package. Always specify the figure width as a multiple of the line width as in the example below using .eps graphics

```
\usepackage[dvips]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.eps}
```

or

```
\usepackage[pdftex]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.pdf}
```

[5] Sample of the first footnote

for .pdf graphics. See section 4.4 in the graphics bundle documentation (`http://www.ctan.org/tex-archive/macros/latex/required/graphics/grfguide.ps`)

A number of width problems arise when LaTeX cannot properly hyphenate a line. Please give LaTeX hyphenation hints using the `\-` command.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

References follow the acknowledgments. Use unnumbered third level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to 'small' (9-point) when listing the references. **Remember that this year you can use a ninth page as long as it contains *only* cited references.**

[1] Kakade, Sham M., and Dean P. Foster. "Multi-view regression via canonical correlation analysis." Learning Theory. Springer Berlin Heidelberg, 2007. 82-96.

[2] Chaudhuri, Kamalika, et al. "Multi-view clustering via canonical correlation analysis." Proceedings of the 26th annual international conference on machine learning. ACM, 2009.

[3] Hardoon, David R., et al. "Unsupervised analysis of fMRI data using kernel canonical correlation." NeuroImage 37.4 (2007): 1250-1259.

[4] Vinokourov, Alexei, Nello Cristianini, and John S. Shawe-Taylor. "Inferring a semantic representation of text via cross-language correlation analysis." Advances in neural information processing systems. 2002.

[5] Dhillon, Paramveer, Dean P. Foster, and Lyle H. Ungar. "Multi-view learning of word embeddings via cca." Advances in Neural Information Processing Systems. 2011.

[6] Hotelling, Harold. "Relations between two sets of variates." Biometrika (1936): 321-377.

[7] Weiran Wang, Raman Arora, Karen Livescu and Jeff Bilmes. On Deep Multi-View Representation Learning. In Proceedings of the 32nd International Conference on Machine Learning (ICML), 2015

[8] Raman Arora and Karen Livescu. Kernel CCA for multi-view acoustic feature learning using articulatory measurements. In Proceedings of the Machine Learning Symposium on Language and Speech Processing (MLSLP), 2012

[9] Rastogi, Pushpendre, Benjamin Van Durme, and Raman Arora. "Multiview LSA: Representation Learning via Generalized CCA."

[10] Ngiam, Jiquan, et al. "Multimodal deep learning." Proceedings of the 28th international conference on machine learning (ICML-11). 2011.

[11] Galen Andrew, Raman Arora, Jeff Bilmes and Karen Livescu. Deep Canonical Correlation Analysis. In Proceedings of the 30th International Conference on Machine Learning (ICML), 2013

[12] Diederik Kingma, Jimmy Ba. Adam: A Method for Stochastic Optimization. In 3rd International Conference for Learning Representations, San Diego, 2015

[13] Akaho, Shotaro. "A kernel method for canonical correlation analysis." arXiv preprint cs/0609071 (2001).