DEEP VARIATIONAL CANONICAL CORRELATION ANALYSIS

Weiran Wang[1] **Xinchen Yan**[2] **Honglak Lee**[2] **Karen Livescu**[1]
[1]TTI-Chicago, Chicago, IL 60637, USA
[2]University of Michigan, Ann Arbor, MI 48109, USA
[1]{weiranwang,klivescu}@ttic.edu {xcyan,honglak}@umich.edu

ABSTRACT

We present deep variational canonical correlation analysis (VCCA), a deep multi-view learning model that extends the latent variable model interpretation of linear CCA (Bach and Jordan, 2005) to nonlinear observation models parameterized by deep neural networks (DNNs). Computing the marginal data likelihood, as well as inference of the latent variables, are intractable under this model. We derive a variational lower bound of the data likelihood by parameterizing the posterior density of the latent variables with another DNN, and approximate the lower bound via Monte Carlo sampling. Interestingly, the resulting model resembles that of multi-view autoencoders (Ngiam et al., 2011), with the key distinction of an additional sampling procedure at the bottleneck layer. We also propose a variant of VCCA called VCCA-private which can, in addition to the "common variables" underlying both views, extract the "private variables" within each view. We demonstrate that VCCA-private is able to disentangle the shared and private information for multi-view data without hard supervision.

1 INTRODUCTION

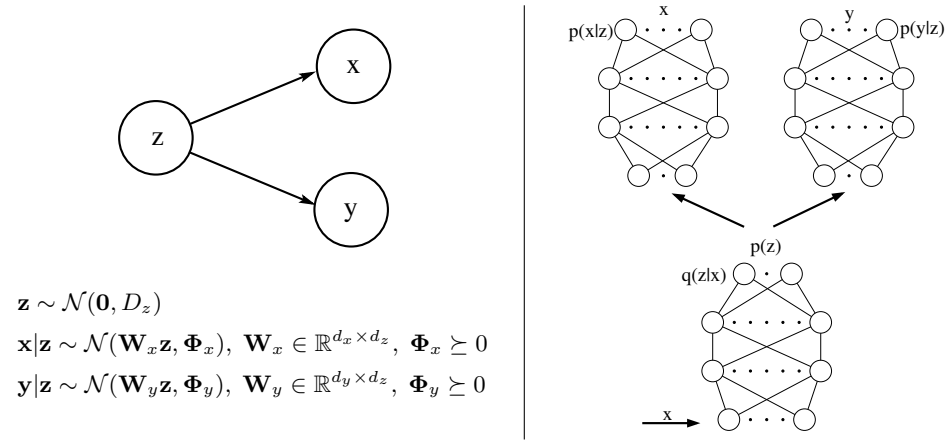
In the multi-view representation learning setting, we have multiple views/measurements of the same underlying signal, and the goal is to learn useful features of each view using complementary information contained in the views. The intuition underlying this setting is that the learned features can help uncover the common sources of variation in the views, which can be helpful for exploratory analysis or for downstream tasks.

A classical approach in this setting is canonical correlation analysis (CCA, Hotelling, 1936) and its nonlinear extensions, including the kernel extension (Lai and Fyfe, 2000; Akaho, 2001; Melzer et al., 2001; Bach and Jordan, 2002) and the deep neural network (DNN) extension (Andrew et al., 2013; Wang et al., 2015b). CCA projects two random vectors $\mathbf{x} \in \mathbb{R}^{d_x}$ and $\mathbf{y} \in \mathbb{R}^{d_y}$ into a lower-dimensional subspace so that the projections are maximally correlated. There is a probabilistic latent variable model interpretation of linear CCA (Bach and Jordan, 2005) as shown in Figure 1 (left). Assume that $\mathbf{x}$ and $\mathbf{y}$ are linear functions of some lower-dimensional random variable $\mathbf{z} \in \mathbb{R}^{d_z}$, where $d_z \leq \min(d_x, d_y)$. When the prior distribution of the latent variable $p(\mathbf{z})$ and the conditional distributions $p(\mathbf{x}|\mathbf{z})$ and $p(\mathbf{y}|\mathbf{z})$ are Gaussian, Bach and Jordan (2005) showed that $\mathbb{E}[\mathbf{z}|\mathbf{x}]$ (resp. $\mathbb{E}[\mathbf{z}|\mathbf{y}]$) lives in the same space as the linear CCA projection for $\mathbf{x}$ (resp. $\mathbf{y}$).

This generative interpretation of CCA is often lost in nonlinear extensions of CCA. For example, in deep CCA (DCCA, (Andrew et al., 2013)), to extend CCA to nonlinear mappings with greater representation power, one extracts nonlinear features from the original inputs of each view using two DNNs, $\mathbf{f}$ for $\mathbf{x}$ and $\mathbf{g}$ for $\mathbf{y}$, so that the canonical correlation of the DNN outputs (measured by a linear CCA with projection matrices $\mathbf{U}$ and $\mathbf{V}$) is maximized. Formally, given a dataset of N pairs of observations $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)$ of the random vectors $(\mathbf{x}, \mathbf{y})$, DCCA optimizes

$$\max_{\substack{\mathbf{W_f}, \mathbf{W_g} \\ \mathbf{U}, \mathbf{V}}} \operatorname{tr}\left(\mathbf{U}^\top \mathbf{f}(\mathbf{X}) \mathbf{g}(\mathbf{Y})^\top \mathbf{V}\right) \quad \text{s.t. } \mathbf{U}^\top \left(\mathbf{f}(\mathbf{X}) \mathbf{f}(\mathbf{X})^\top\right) \mathbf{U} = \mathbf{V}^\top \left(\mathbf{g}(\mathbf{Y}) \mathbf{g}(\mathbf{Y})^\top\right) \mathbf{V} = N\mathbf{I}, \quad (1)$$

where $\mathbf{f}(\mathbf{X}) = [\mathbf{f}(\mathbf{x}_1), \ldots, \mathbf{f}(\mathbf{x}_N)]$ and $\mathbf{g}(\mathbf{Y}) = [\mathbf{g}(\mathbf{y}_1), \ldots, \mathbf{g}(\mathbf{y}_N)]$, and $\mathbf{W_f}$ denotes all weight parameters of the DNN $\mathbf{f}$ (and similarly for $\mathbf{g}$).



$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, D_z)$$
$$\mathbf{x}|\mathbf{z} \sim \mathcal{N}(\mathbf{W}_x\mathbf{z}, \mathbf{\Phi}_x),\ \mathbf{W}_x \in \mathbb{R}^{d_x \times d_z},\ \mathbf{\Phi}_x \succeq 0$$
$$\mathbf{y}|\mathbf{z} \sim \mathcal{N}(\mathbf{W}_y\mathbf{z}, \mathbf{\Phi}_y),\ \mathbf{W}_y \in \mathbb{R}^{d_y \times d_z},\ \mathbf{\Phi}_y \succeq 0$$

Figure 1: Left: Probabilistic interpretation of CCA (Bach and Jordan, 2005). Right: The deep variational CCA (VCCA) model.

DCCA has achieved good performance in the multi-view representation learning setting across different domains (Wang et al., 2015b,a; Lu et al., 2015; Yan and Mikolajczyk, 2015). However, a disadvantage of DCCA is that it directly looks for DNNs that can map inputs into the low-dimensional space, without a model for generating samples from the latent space. Although Wang et al. (2015b)'s deep canonically correlated autoencoders (DCCAE) model optimizes the combination of the autoencoder objective (reconstruction errors) and the canonical correlation objective, the authors found that in practice, the canonical correlation term tends to dominate the reconstruction error terms in the DCCAE objective when tuning performance for a downstream task (especially when the inputs are noisy), and as a result the inputs are not reconstructed well. At the same time, optimization of the DCCA and DCCAE objectives is challenging due to the constraints that couple all training samples.

The main contribution of this paper is the proposal of a new deep multi-view learning model named deep variational CCA (VCCA), which extends the latent variable model interpretation of linear CCA to nonlinear observation models parameterized by DNNs. Computing the marginal data likelihood, as well as inference of the latent variables, are intractable under this model. Inspired by variational autoencoders (VAE, Kingma and Welling, 2014), we parameterize the posterior distribution of the latent variables with another DNN, and derive a variational lower bound of the data likelihood as the objective of VCCA, which is further approximated by Monte Carlo sampling. With the reparameterization trick, sampling for the Monte Carlo approximation is trivial and all DNN weights in VCCA can be optimized jointly via stochastic gradient descent, using unbiased gradient estimates from small minibatches. Interestingly, VCCA is related to multi-view autoencoders (Ngiam et al., 2011), with the key distinctions of additional regularization on the posterior distribution and the sampling procedure at the bottleneck layer.

We also propose a variant of VCCA called VCCA-private that can, in addition to the "common variables" underlying both views, extract the "private variables" within each view. We demonstrate that VCCA-private is able to disentangle the shared and private information for multi-view data without hard supervision. Last but not least, as generative models, VCCA and VCCA-private enable us to obtain high-quality samples for the input of each view.

2 VARIATIONAL CCA

The probabilistic latent variable model of CCA (Bach and Jordan, 2005) defines the following joint distribution over the random variables $(\mathbf{x}, \mathbf{y})$:

$$p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})p(\mathbf{y}|\mathbf{z}), \qquad p(\mathbf{x}, \mathbf{y}) = \int p(\mathbf{x}, \mathbf{y}, \mathbf{z})d\mathbf{z}. \tag{2}$$

The assumption underlying this model is that, conditioned on the latent variables $\mathbf{z} \in \mathbb{R}^{d_z}$, the two views $\mathbf{x}$ and $\mathbf{y}$ are independent. However, linear observation models ($p(\mathbf{x}|\mathbf{z})$ and $p(\mathbf{y}|\mathbf{z})$ as shown in Figure 1 (left)) have limited representation power. In this paper, we consider nonlinear

observation models $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z};\boldsymbol{\theta}_x)$ and $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z};\boldsymbol{\theta}_y)$, parameterized by $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_y$ respectively, which can be the collections of weights of DNNs. In this case, the marginal likelihood $p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y})$ does not have a closed form. In addition, the inference problem $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$—the problem of inferring the latent variables given one of the views—is also intractable.

Inspired by Kingma and Welling (2014)'s work on variational autoencoders (VAE), we approximate $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ with the conditional density $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x};\boldsymbol{\phi}_z)$, where $\boldsymbol{\phi}_z$ is the collection of parameters of another DNN.[1] We can derive a lower bound on the marginal data likelihood using $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$:

$$\begin{aligned}
\log p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y}) &= \log p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y}) \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) d\mathbf{z} = \int \log p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y}) q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) d\mathbf{z} \\
&= \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \left(\log \frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x},\mathbf{y})} + \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \right) d\mathbf{z} \\
&= D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) || p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x},\mathbf{y})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \right] \\
&\geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{y},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \right] =: \mathcal{L}(\mathbf{x},\mathbf{y};\boldsymbol{\theta},\boldsymbol{\phi}) \qquad (3)
\end{aligned}$$

where we used the fact that KL divergence is nonnegative in the last step. As a result, $\mathcal{L}(\mathbf{x},\mathbf{y};\boldsymbol{\theta},\boldsymbol{\phi})$ is a lower bound on the data log-likelihood $\log_{\boldsymbol{\theta}} p(\mathbf{x},\mathbf{y})$. Substituting (2) into (3), we have

$$\begin{aligned}
\mathcal{L}(\mathbf{x},\mathbf{y};\boldsymbol{\theta},\boldsymbol{\phi}) &= \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \left(\log \frac{p(\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} + \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z}) \right) d\mathbf{z} \\
&= -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) || p(\mathbf{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z}) \right]. \qquad (4)
\end{aligned}$$

VCCA maximizes this variational lower bound on the data likelihood on the training set:

$$\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathbf{x}_i,\mathbf{y}_i;\boldsymbol{\theta},\boldsymbol{\phi}). \qquad (5)$$

The first term in (4) measures the KL divergence between the approximate posterior distribution and the prior distribution of the latent variables $\mathbf{z}$. When the parameterization $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is chosen properly, this term can be computed exactly in closed form. As a concrete example, let the variational approximate posterior be a multivariate Gaussian with diagonal covariance. That is, for a sample pair $(\mathbf{x}_i,\mathbf{y}_i)$, we have

$$\log q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x}_i) = \log \mathcal{N}(\mathbf{z}_i;\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i), \qquad \boldsymbol{\Sigma}_i = \operatorname{diag}\left(\sigma_{i1}^2,\ldots,\sigma_{id_z}^2\right), \qquad (6)$$

where the mean $\boldsymbol{\mu}_i$ and covariance $\boldsymbol{\Sigma}_i$ are outputs of an encoding DNN $\mathbf{f}$ (and thus $[\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i] = \mathbf{f}(\mathbf{x}_i;\boldsymbol{\phi}_z)$ are *deterministic* nonlinear functions of $\mathbf{x}_i$). In this case, we have

$$D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x}_i) || p(\mathbf{z}_i)) = -\frac{1}{2} \sum_{j=1}^{d_z} \left(1 + \log \sigma_{ij}^2 - \sigma_{ij}^2 - \mu_{ij}^2\right).$$

The second term of (4) corresponds to the expected complete data likelihood under the approximate posterior distribution. Though still intractable, this term can be approximated by Monte Carlo sampling. In particular, we draw L samples $\mathbf{z}_i^{(l)} \sim q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x}_i)$:

$$\mathbf{z}_i^{(l)} = \boldsymbol{\mu}_i + \boldsymbol{\Sigma}_i \boldsymbol{\epsilon}^{(l)}, \qquad \text{where} \quad \boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(\mathbf{0},\mathbf{I}), \qquad \text{for} \quad l = 1,\ldots,L, \qquad (7)$$

and have

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x}_i)} \left[\log p_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}_i) + \log p_{\boldsymbol{\theta}}(\mathbf{y}_i|\mathbf{z}_i) \right] \approx \frac{1}{L} \sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}\left(\mathbf{x}_i|\mathbf{z}_i^{(l)}\right) + \log p_{\boldsymbol{\theta}}\left(\mathbf{y}_i|\mathbf{z}_i^{(l)}\right). \qquad (8)$$

Notice that we parameterized $q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x}_i)$ above to obtain the VCCA objective; this is useful when the first view is available for downstream tasks, in which case we can directly apply $q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x}_i)$ to obtain its projection (as features). One could also derive likelihood lower bounds by parameterizing the approximate posteriors $q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{y}_i)$ and $q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x}_i,\mathbf{y}_i)$, and optimize their convex combinations for training. We give a sketch of VCCA in Figure 1 (right).

[1] For notational simplicity, we denote by $\boldsymbol{\theta}$ the collection of parameters associated with the model probabilities $p_{\boldsymbol{\theta}}(\cdot)$, and $\boldsymbol{\phi}$ the collection of parameters associated with the variational approximate probabilities $q_{\boldsymbol{\phi}}(\cdot)$, and often omit specific parameters inside the probabilities.

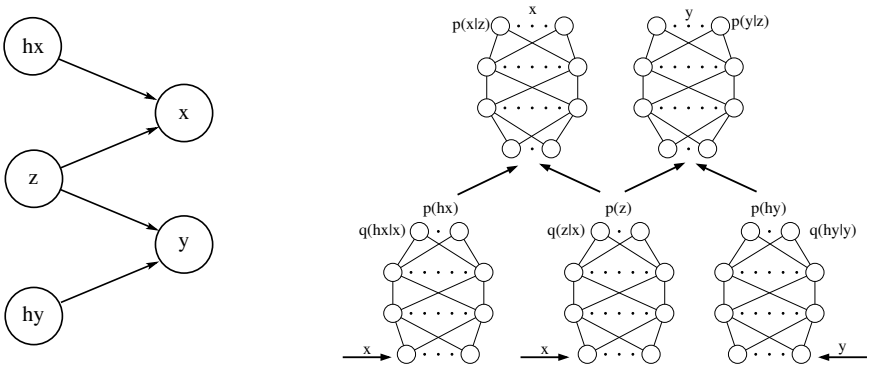


Figure 2: VCCA-private: variational CCA with view-specific private variables.

Connection to multi-view autoencoder (MVAE) If we use the Gaussian observation models

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \log \mathcal{N}(\mathbf{g}_x(\mathbf{z};\boldsymbol{\theta}_x), \mathbf{I}), \qquad \log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z}) = \log \mathcal{N}(\mathbf{g}_y(\mathbf{z};\boldsymbol{\theta}_y), \mathbf{I}),$$

we observe that $\log p_{\boldsymbol{\theta}}\left(\mathbf{x}_i|\mathbf{z}_i^{(l)}\right)$ and $\log p_{\boldsymbol{\theta}}\left(\mathbf{y}_i|\mathbf{z}_i^{(l)}\right)$ measure the reconstruction errors of each view's inputs from samples $\mathbf{z}_i^{(l)}$ using the two DNNs $\mathbf{g}_x$ and $\mathbf{g}_y$ respectively. In this case, maximizing $\mathcal{L}(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}, \boldsymbol{\phi})$ is equivalent to

$$\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \quad \frac{1}{N}\sum_{i=1}^{N} D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x}_i)||p(\mathbf{z}_i)) + \frac{1}{2NL}\sum_{i=1}^{N}\sum_{l=1}^{L}\left\|\mathbf{x}_i - \mathbf{g}_x\left(\mathbf{z}_i^{(l)};\boldsymbol{\theta}_x\right)\right\|^2 + \left\|\mathbf{y}_i - \mathbf{g}_y\left(\mathbf{z}_i^{(l)};\boldsymbol{\theta}_y\right)\right\|^2 \tag{9}$$

$$\text{s.t.} \quad \mathbf{z}_i^{(l)} = \boldsymbol{\mu}_i + \boldsymbol{\Sigma}_i\boldsymbol{\epsilon}^{(l)}, \quad \text{where } \boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad l = 1, \ldots, L.$$

Now, consider the case of $\boldsymbol{\Sigma}_i \to \mathbf{0}$, for $i = 1, \ldots, N$, and we have $\mathbf{z}_i^{(l)} \to \boldsymbol{\mu}_i$ which is a deterministic function of $\mathbf{x}$ (and there is no need for sampling). In the limit, the second term of (9) reduces to

$$\frac{1}{2N}\sum_{i=1}^{N}\left\|\mathbf{x}_i - \mathbf{g}_x(\mathbf{f}(\mathbf{x}_i;\boldsymbol{\phi}_z);\boldsymbol{\theta}_x)\right\|^2 + \left\|\mathbf{y}_i - \mathbf{g}_y(\mathbf{f}(\mathbf{x}_i;\boldsymbol{\phi}_z);\boldsymbol{\theta}_y)\right\|^2, \tag{10}$$

which is the objective of the multi-view autoencoder (MVAE, Ngiam et al., 2011). Note, however, that $\boldsymbol{\Sigma}_i \to \mathbf{0}$ is prevented by the VCCA objective as it results in a large penalty in $D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x}_i)||p(\mathbf{z}_i))$. Compared with the MVAE objective, in the VCCA objective we are creating L different "noisy" versions of the latent representation and enforce that these versions reconstruct the original inputs well. The "noise" distribution (the variances $\boldsymbol{\Sigma}_i$) are also learned and regularized by the KL divergence $D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x}_i)||p(\mathbf{z}_i))$. Using the VCCA objective, we expect to learn different representations from those of MVAE, due to these regularization effects.

2.1 Extracting private variables

So far, VCCA aims at extracting only the latent variables $\mathbf{z}$ that are common to both views. A potential disadvantage of this model is that it assumes the common variables are sufficient by themselves to generate the views, which can be too restrictive in practice. Consider the example of audio and articulatory measurements as two views for speech. Although the transcription is a common variable behind the views, it combines with the physical environment and the vocal tract anatomy to generate the individual views. In other words, there might be large variations in the input space that can not be explained by the common variables, making the objective (4) hard to optimize. It may then be beneficial to explicitly model the private variables within each view.

We therefore propose a new probabilistic graphical model, shown in Figure 2, that we refer to as VCCA-private. We introduce two sets of hidden variables $\mathbf{h}_x \in \mathbb{R}^{d_{h_x}}$ and $\mathbf{h}_y \in \mathbb{R}^{d_{h_y}}$ to explain the aspects of $\mathbf{x}$ and $\mathbf{y}$ not captured by the common variables $\mathbf{z}$. Under this model, the data likelihood

is defined by

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{h}_x, \mathbf{h}_y) = p(\mathbf{z})p(\mathbf{h}_x)p(\mathbf{h}_y)p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}, \mathbf{h}_x; \boldsymbol{\theta}_x)p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z}, \mathbf{h}_y; \boldsymbol{\theta}_y), \tag{11}$$
$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) = \int\int\int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{h}_x, \mathbf{h}_y) d\mathbf{z}\, d\mathbf{h}_x\, d\mathbf{h}_y.$$

To obtain tractable inference, we introduce the following factored variational posterior

$$q_{\boldsymbol{\phi}}(\mathbf{z}, \mathbf{h}_x, \mathbf{h}_y|\mathbf{x}, \mathbf{y}) = q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}; \boldsymbol{\phi}_z)q_{\boldsymbol{\phi}}(\mathbf{h}_x|\mathbf{x}; \boldsymbol{\phi}_x)q_{\boldsymbol{\phi}}(\mathbf{h}_y|\mathbf{y}; \boldsymbol{\phi}_y), \tag{12}$$

where each factor is parameterized by a different DNN. Similarly to VCCA, we can derive a variational lower bound on the data likelihood for VCCA-private as

$$\begin{aligned}
&\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) \\
&\geq \int\int\int q_{\boldsymbol{\phi}}(\mathbf{z}, \mathbf{h}_x, \mathbf{h}_y|\mathbf{x}, \mathbf{y}) \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{h}_x, \mathbf{h}_y)}{q_{\boldsymbol{\phi}}(\mathbf{z}, \mathbf{h}_x, \mathbf{h}_y|\mathbf{x}, \mathbf{y})} d\mathbf{z}\, d\mathbf{h}_x\, d\mathbf{h}_y \\
&= \int\int\int q_{\boldsymbol{\phi}}(\mathbf{z}, \mathbf{h}_x, \mathbf{h}_y|\mathbf{x}, \mathbf{y}) \Bigg[\log \frac{p(\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} + \log \frac{p(\mathbf{h}_x)}{q_{\boldsymbol{\phi}}(\mathbf{h}_x|\mathbf{x})} + \log \frac{p(\mathbf{h}_y)}{q_{\boldsymbol{\phi}}(\mathbf{h}_y|\mathbf{y})} \\
&\qquad\qquad + \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}, \mathbf{h}_x) + \log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z}, \mathbf{h}_y)\Bigg] d\mathbf{z}\, d\mathbf{h}_x\, d\mathbf{h}_y \\
&= -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) - D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{h}_x|\mathbf{x})||p(\mathbf{h}_x)) - D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{h}_y|\mathbf{y})||p(\mathbf{h}_y)) \\
&\quad + \int\int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{h}_x|\mathbf{x}) \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}, \mathbf{h}_x) d\mathbf{z}\, d\mathbf{h}_x + \int\int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{h}_y|\mathbf{y}) \log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z}, \mathbf{h}_y) d\mathbf{z}\, d\mathbf{h}_y \\
&=: \mathcal{L}_{\text{private}}(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}, \boldsymbol{\phi}).
\end{aligned} \tag{13}$$

As in VCCA, the last two terms of (14) can be approximated by Monte Carlo sampling. For example, we draw samples of $\mathbf{z}$ and $\mathbf{h}_x$ from their corresponding approximate posteriors, and concatenate their samples as inputs to the DNN parameterizing $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}, \mathbf{h}_x)$. In this paper, we use simple Gaussian prior distributions for the private variables, i.e., $\mathbf{h}_x \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\mathbf{h}_y \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We leave to future work to examine the effect of more sophisticated prior distributions for the latent variables.

VCCA-private maximizes this lower bound on the training set, i.e.,

$$\max_{\boldsymbol{\theta}, \boldsymbol{\phi}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\text{private}}(\mathbf{x}_i, \mathbf{y}_i; \boldsymbol{\theta}, \boldsymbol{\phi}). \tag{14}$$

Optimization The objectives (5) and (14) decouple over the training samples and can be trained efficiently using stochastic gradient descent. Enabled by the reparameterization trick, unbiased gradient estimates are obtained by Monte Carlo sampling and the standard backpropagation procedure on minibatches of training samples. We apply the ADAM algorithm (Kingma and Ba, 2015) for optimizing our objectives.

3 Related work

Recently, there has been much interest in unsupervised deep generative models (Kingma and Welling, 2014; Rezende et al., 2014; Goodfellow et al., 2014; Gregor et al., 2015; Makhzani et al., 2016; Burda et al., 2016; Alain et al., 2016). A common motivation behind these models is that, with the expressive power of DNNs, the generative models can capture distributions for complex inputs. Additionally, if we are able to generate realistic samples from the learned distribution, we can infer that we have discovered the underlying structure of the data, which may allow us to reduce the sample complexity for learning for downstream tasks. These previous models have mostly focused on single-view data. Here we focus on the multi-view setting where multiple views of the data are present for feature extraction but only one view is available at test time (in downstream tasks).

Some recent work has explored deep generative models for (semi-)supervised learning. Kingma et al. (2014) built a generative model based on variational autoencoders (VAEs) for semi-supervised classification, where the authors model the input distribution with two set of latent variables: the class label (if it is missing) and another set that models the intra-class variabilities (styles). Sohn

et al. (2015) proposed a conditional generative model for structured output prediction, where the authors explicitly model the uncertainty in the input/output using Gaussian latent variables. While there are two set of observations (input and output labels) in these work, their graphical models are different from that of VCCA.

Our work is also related to the deep multi-view probabilistic models based on restricted Boltzmann machines (Srivastava and Salakhutdinov, 2014; Sohn et al., 2014). We note that these are undirected graphical models for which both inference and learning are difficult, and one typically resorts to carefully designed variational approximation and Gibbs sampling procedures for training such models. In contrast, our models only require sampling from simple, standard distributions (such as Gaussians), and all parameters can be learned end-to-end by standard stochastic gradient methods. Therefore, our models are more scalable than the previous multi-view probabilistic models.

On the other hand, there is a rich literature in modeling multi-view data using the same or similar graphical models behind VCCA/VCCA-private (Wang, 2007; Jia et al., 2010; Salzmann et al., 2010; Virtanen et al., 2011; Memisevic et al., 2012; Klami et al., 2013). Our methods differ from previous work in parameterizing the probability distributions using DNNs. This makes the model more powerful, while still having tractable objectives and efficient end-to-end training using the local reparameterization technique. We note that, unlike earlier work on probabilistic models of linear CCA (Bach and Jordan, 2005), VCCA does not optimize the same criterion, nor produce the same solution, as any linear or nonlinear CCA. However, we retain the terminology in order to clarify the connection with earlier work on probabilistic models for CCA, which we are extending with DNN models for the observations and for the variational posterior distribution approximation.

4 Experimental results

In this section, we compare different multi-view representation learning algorithms on three tasks involving several domains: image-image, speech-articulation, and image-text. The algorithms we choose to compare below are closely related to the proposed model or have been shown to have strong empirical performance under similar settings.

- Linear CCA: its probabilistic interpretation motivates this work.
- Deep CCA (DCCA) (Andrew et al., 2013): see its objective in (1).
- Deep canonically correlated autoencoders (DCCAE) (Wang et al., 2015b): combination of the DCCA objective and the reconstruction errors of each view.
- Multi-view autoencoder (MVAE) (Ngiam et al., 2011): see its objective in (10).
- Multi-view contrastive loss (Hermann and Blunsom, 2014): based on the intuition that the distance between embeddings of paired examples $\mathbf{x}^+$ and $\mathbf{y}^+$ should be smaller than the distance between embeddings of $\mathbf{x}^+$ and an unmatched negative example $\mathbf{y}^-$ by a margin:

$$\min_{f,g} \mathcal{L}_{contrast} := \frac{1}{N}\sum_{i}^{N} \max\left(0,\ m + dis\left(f(\mathbf{x}_i^+),\, g(\mathbf{y}_i^+)\right) - dis\left(f(\mathbf{x}_i^+),\, g(\mathbf{y}_i^-)\right)\right),$$

  where $\mathbf{y}_i^-$ is a randomly sampled view 2 example, and m is a margin hyperparameter. We use the cosine distance $dis\left(\mathbf{a}, \mathbf{b}\right) = 1 - \left\langle \frac{\mathbf{a}}{\|\mathbf{a}\|}, \frac{\mathbf{b}}{\|\mathbf{b}\|} \right\rangle$.

4.1 Noisy MNIST dataset

We first demonstrate our algorithms on the noisy MNIST dataset used by Wang et al. (2015b). The dataset is generated using the MNIST dataset (LeCun et al., 1998), which consists of 28×28 grayscale digit images, with $60K/10K$ images for training/testing. We first linearly rescale the pixel values to the range $[0, 1]$. Then, we randomly rotate the images at angles uniformly sampled from $[-\pi/4, \pi/4]$ and the resulting images are used as view 1 inputs. For each view 1 image, we randomly select an image of the same identity (0-9) from the original dataset, add independent random noise uniformly sampled from $[0, 1]$ to each pixel, and truncate the pixel final values to $[0, 1]$ to obtain the corresponding view 2 sample. Selection of input images are given in Figure 3 (left). The original

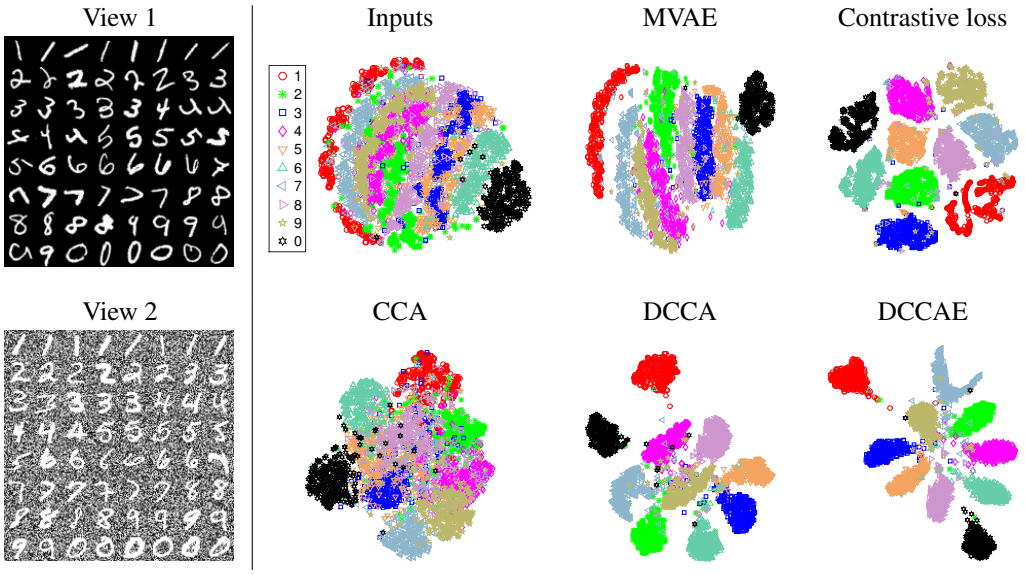


Figure 3: Left: Selection of view 1 images (top) and their corresponding view 2 images (bottom) from noisy MNIST. Right: 2D t-SNE visualization of features learned by previous multi-view models.

training set is further split into training/tuning sets of size $50K$/$10K$. The data generation process ensures that the digit identity is the only common variable underlying both views.

To evaluate the amount of class information extracted by different methods, after unsupervised learning of latent representations, we reveal the labels and train a linear SVM on the projected view 1 training data (using the one-versus-all scheme), and use it to classify the projected test set. This experiment simulates the typical usage of multi-view learning methods, which is to extract useful representations for downstream discriminative tasks.

Note that this synthetic dataset perfectly satisfies the multi-view assumption that the two views are independent given the class label, so the latent representation should contain precisely the class information. This is indeed achieved by CCA-based and contrastive loss-based multi-view approaches. In Figure 3 (right), we show 2D t-SNE (van der Maaten and Hinton, 2008) visualizations of the original view 1 inputs and view 1 projections by various deep multi-view methods.

We use DNNs with 3 hidden layers of 1024 rectified linear units (ReLUs, Nair and Hinton, 2010) each to parameterize the distributions: $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z})$ in VCCA, and additionally $q_{\boldsymbol{\phi}}(\mathbf{h}_x|\mathbf{x})$ and $q_{\boldsymbol{\phi}}(\mathbf{h}_y|\mathbf{y})$ in VCCA-private. The capacities of these networks are the same as those of their counterparts in DCCA and DCCAE from Wang et al. (2015b). The reconstruction networks $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ or $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}, \mathbf{h}_x)$ model each pixel of $\mathbf{x}$ as an independent Bernoulli variable and parameterize its mean (using a sigmoid activation); $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z})$ and $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z}, \mathbf{h}_y)$ model $\mathbf{y}$ with diagonal Gaussians and parameterize the mean (using a sigmoid activation) and standard deviation for each pixel dimension. We tune the dimensionality d_z over $\{10, 20, 30, 40, 50\}$, and fix $d_{h_x} = d_{h_y} = 30$ for VCCA-private. We select the hyperparameter combination that yields the best SVM classification accuracy on the projected tuning set, and report the corresponding accuracy on the projected test set.

The effect of dropout We add dropout (Srivastava et al., 2014) to all intermediate layers and the input layers and find it to be very useful in our models, with most of the gain coming from dropout applied to the samples of $\mathbf{z}$, $\mathbf{h}_x$ and $\mathbf{h}_y$. This is because dropout encourages each latent dimension to reconstruct the inputs well in the absence of other dimensions, and therefore avoids learning co-adapted features. Intuitively, in VCCA-private dropout also helps to prevent the degenerate situation where the pathways $\mathbf{x} \rightarrow \mathbf{h}_x \rightarrow \mathbf{x}$ and $\mathbf{y} \rightarrow \mathbf{h}_y \rightarrow \mathbf{y}$ achieve good reconstruction while ignoring $\mathbf{z}$ (e.g., by setting it to a constant). We use the same dropout rate for all layers and tune it over $\{0, 0.1, 0.2, 0.3, 0.4\}$.

We show the 2D t-SNE embeddings of the common variables $\mathbf{z}$ learned by VCCA and VCCA-private on test set in Figure 4. We observe that in general, VCCA/VCCA-private tend to separate the classes

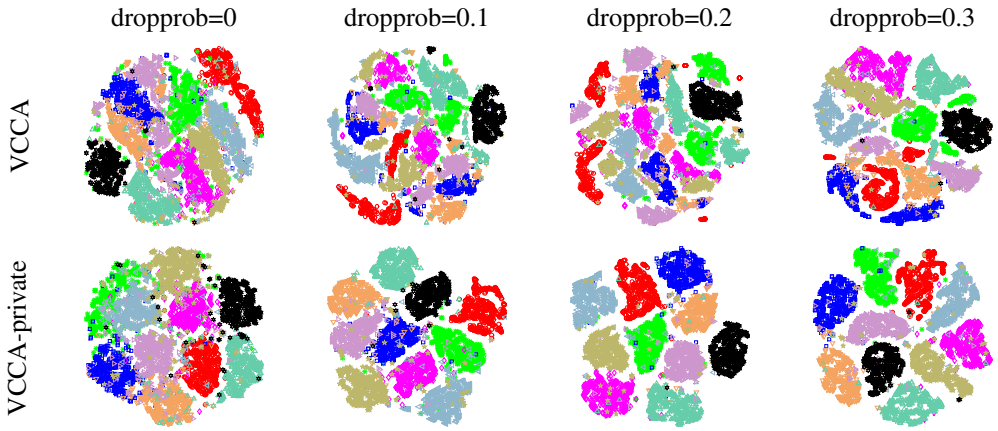


Figure 4: 2D t-SNE embeddings of the extracted shared variables $\mathbf{z}$ on the test set by VCCA (top row) and VCCA-private (bottom row) for different dropout rates. $d_z = 40$ for both algorithms.

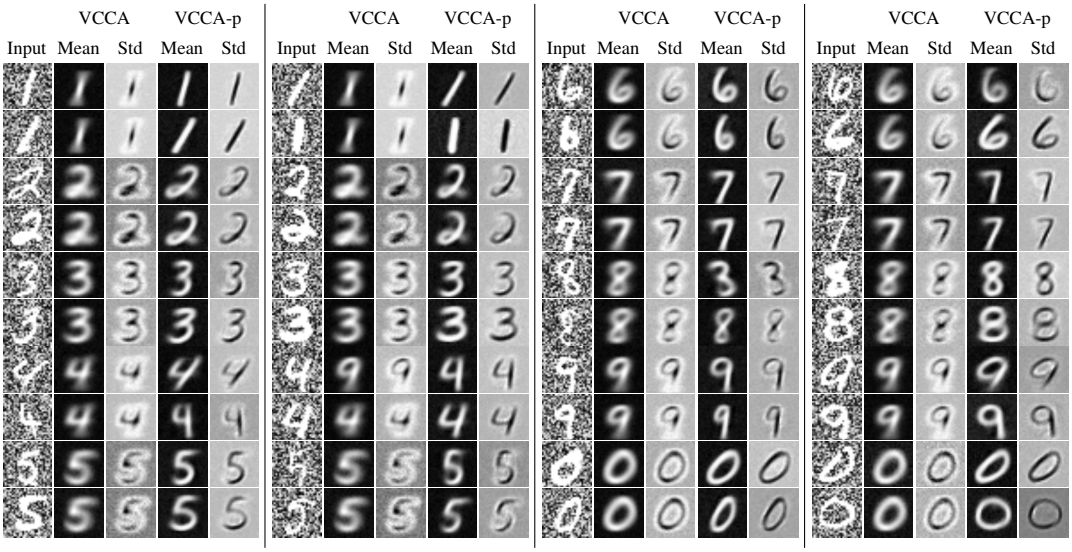


Figure 5: Sample reconstruction of view 2 images from the test set by VCCA and VCCA-private.

in the projection well; dropout significantly improves the performance of both VCCA and VCCA-private, with the latter slightly outperforming the former. While such class separation can also be achieved by DCCA/contrastive loss as well, these methods can not naturally generate samples in the input space. On the other hand, such separation is not achieved by multi-view autoencoders.

The effect of private variables on reconstructions We show sample reconstructions (mean and standard deviation) by VCCA for the view 2 images from the test set in Figure 5 (columns 2 and 3). We observe that for each input, the mean reconstruction of $\mathbf{y}_i$ by VCCA is a prototypical image of the same digit, regardless of the individual style in $\mathbf{y}_i$. This is to be expected, as $\mathbf{y}_i$ contains an arbitrary image of the same digit as $\mathbf{x}_i$, and the variation in background noise in $\mathbf{y}_i$ does not appear in $\mathbf{x}_i$ and can not be reflected in $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$; thus the best way for $\mathbf{p}_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z})$ to model $\mathbf{y}_i$ is to output a prototypical image of that class to achieve on average small reconstruction error. On the other hand, since $\mathbf{y}_i$ contains little rotation of the digits, this variation is suppressed to a large extent in $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ (it is no longer the major variation in $\mathbf{z}$ as in the original inputs).

We show sample reconstructions by VCCA-private for the same set of view 2 images in Figure 5 (columns 4 and 5). With the help of private variables $\mathbf{h}_y$ (as part of the input to $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z}, \mathbf{h}_y)$), the model does a much better job in reconstructing the styles of $\mathbf{y}$. And by disentangling the private variables from the shared variables, $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ achieves even better class separation than VCCA does.

Table 1: Performance of features extracted by different methods for downstream tasks: Classification error rates of linear SVMs on MNIST, mean phone error rate (PER) over 6 folds on XRMB, and mean average precision (mAP) for unimodal retrieval on MIR-Flickr.

Method	Noisy MNIST Error rate (%, ↓)	XRMB PER (%, ↓)	Flickr mAP (↑)
Original inputs	13.1	37.6	0.480
CCA	19.1	29.4	0.529
DCCA	2.9	25.4	0.573
DCCAE	2.2	25.4	0.573
Contrastive	2.7	24.6	0.565
MVAE	11.7	29.4	0.477
VCCA	3.0	28.0	0.605
VCCA-private	2.4	25.2	0.609

We also note that the standard deviation of the reconstruction is low within the digit and high outside the digit, implying that $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z}, \mathbf{h}_y)$ is able to separate the background noise from the digit image.

Disentanglement of private/shared variables In Figure 6 (in Appendix) we provide the 2D t-SNE embeddings of the shared variables $\mathbf{z}$ (top row) and the private variables $\mathbf{h}_x$ (bottom row) learned by VCCA-private. In the embedding of $\mathbf{h}_x$, digits with different identities but the same rotation are mapped close together, and the rotation varies smoothly from left to right, confirming that the private variables contain little class information but mainly style information.

Finally, we give the test error rates of linear SVMs applied to the features learned with different models in Table 1. VCCA-private is comparable in performance to the best previous approach (DCCAE), while having the advantage that it can also generate.

4.2 XRMB SPEECH-ARTICULATION DATASET

We now consider the task of learning acoustic features for speech recognition. We use data from the Wisconsin X-ray microbeam (XRMB) corpus (Westbury, 1994), which contains simultaneously recorded speech and articulatory measurements from 47 American English speakers. We follow the setup of Wang et al. (2015a,b) and use the learned features for speaker-independent phonetic recognition.[2] The two input views are standard 39D acoustic features (13 mel frequency cepstral coefficients (MFCCs) and their first and second derivatives) and 16D articulatory features (horizontal/vertical displacement of 8 pellets attached to several parts of the vocal tract), each then concatenated over a 7-frame window around each frame to incorporate context. The speakers are split into disjoint sets of 35/8/2/2 speakers for feature learning/recognizer training/tuning/testing. The 35 speakers for feature learning are fixed; the remaining 12 are used in a 6-fold experiment (recognizer training on 8 speakers, tuning on 2 speakers, and testing on the remaining 2 speakers). Each speaker has roughly $50K$ frames. We remove the per-speaker mean and variance of the articulatory measurements for each training speaker, and remove the mean of the acoustic measurements for each utterance. All learned feature types are used in a "tandem" speech recognizer (Hermansky et al., 2000), i.e., they are appended to the original 39D features and used in a standard hidden Markov model (HMM)-based recognizer with Gaussian mixture observation distributions.

Each algorithm uses up to 3 ReLU hidden layers, each of 1500 units, for the projection and reconstruction mappings. For VCCA/VCCA-private, we use Gaussian observation models as the inputs are real-valued. In contrast to the MNIST experiments, we do not learn the standard deviations of each output dimension on training data, as this leads to poor downstream task performance. Instead, we use isotropic covariances for each view, and tune the standard deviations by grid search. The best model uses a smaller standard deviation (0.1) for the view 2 than for view 1 (1.0), effectively putting more emphasis on the reconstruction of articulatory measurements. Our best performing VCCA model uses $d_z = 70$, while the best performing VCCA-private model uses $d_z = 70$ and $d_{h_x} = d_{h_y} = 10$.

[2] As in Wang and Livescu (2016), we use the Kaldi toolkit (Povey et al., 2011) for feature extraction and recognition with hidden Markov models. Our results do not match Wang et al. (2015a,b) (who instead used the HTK toolkit (Young et al., 1999)) for the same types of features, but the relative results are consistent.

The mean phone error rates (PER) over 6 folds obtained by different algorithms are given in Table 1. Our methods achieve competitive performance in comparison to previous deep multi-view methods.

4.3 MIR-FLICKR DATASET

Finally, we consider the task of learning cross-modality features for topic classification on the MIR-Flickr database (Huiskes and Lew, 2008). The Flickr database contains 1 million images accompanied by user tags, among which 25000 images are labeled with 38 topic classes (each image may be categorized as multiple topics). We use the same image and text features as in previous work (Srivastava and Salakhutdinov, 2014; Sohn et al., 2014): the image feature is 3857 dimensional real-valued vector, composed of Pyramid Histogram of Words (PHOW) (Bosch et al., 2007), GIST (Oliva and Torralba, 2001), and MPEG-7 descriptors (Manjunath et al., 2001), while the text feature is a 2000-dimensional binary vector of frequent tags.

Following the same protocol as Sohn et al. (2014), we train multi-view representations using the unlabelled data,[3] and use projected image features of the labeled data (further divided into splits of 10000/5000/10000 samples for training/tuning/testing) for training and evaluating a classifier that predicts the topic labels, corresponding to the unimodal query task in Srivastava and Salakhutdinov (2014); Sohn et al. (2014). For each algorithm, we select the model which achieves the highest mean average precision (mAP) on the validation set, and report its performance on the test set.

Each algorithm uses up to 4 ReLU hidden layers, each of 1024 units, for the projection and reconstruction mappings. For VCCA/VCCA-private, we use Gaussian observation models with isotropic covariance for image features, with standard deviation tuned by grid search, and a Bernoulli model for text features. In this experiment, we also found it helpful to tune an additional trade-off parameter for the text-view likelihood (cross-entropy); the best VCCA/VCCA-private models prefer a large trade-off parameter of the level 10^4, emphasizing the reconstruction of the sparse text-view inputs. Our best performing VCCA model uses $d_z = 1024$, while the best performing VCCA-private model uses $d_z = 1024$ and $d_{h_x} = d_{h_y} = 16$.

As shown in Table 1, VCCA/VCCA-private achieve significantly higher mAPs than other methods considered here. Being much easier to train, the performance of our methods are competitive with the previous state-of-the-art mAP result of 0.607 achieved by the multi-view RBMs of Sohn et al. (2014) under the same setting.

5 CONCLUSIONS

We have proposed variational canonical correlation analysis (VCCA), a deep generative method for multi-view representation learning. Our method embodies a natural idea for multi-view learning: the multiple views can be generated from a small set of shared latent variables. VCCA is parameterized by DNNs and can be trained efficiently by backpropagation, and is therefore scalable. We have also shown that, by modeling the private variables that are specific to each view, the VCCA-private variant can disentangle shared/private variables and provide higher-quality reconstructions.

In the future, we will explore other prior distributions such as mixtures of Gaussians or discrete random variables, which may enforce clustering in the latent space and in turn work better for discriminative tasks. We will also explore other observation models, including replacing the auto-encoder objective with that of adversarial networks (Goodfellow et al., 2014; Makhzani et al., 2016; Chen et al., 2016). Another direction is to explicitly incorporate the structure of the inputs, such as the sequence structure of speech and text and the spatial structure of images.

ACKOWLEDGEMENTS

This research was supported by NSF grant IIS-1321015. The opinions expressed in this work are those of the authors and do not necessarily reflect the views of the funding agency. This research used GPUs donated by NVIDIA Corporation.

[3] As in Sohn et al. (2014), we exclude about 250000 samples which contain fewer than two tags.

REFERENCES

S. Akaho. A kernel method for canonical correlation analysis. In *Proceedings of the International Meeting of the Psychometric Society (IMPS2001)*, 2001.

G. Alain, Y. Bengio, L. Yao, J. Yosinski, E. Thibodeau-Laufer, S. Zhang, and P. Vincent. GSNs: Generative stochastic networks. *Information and Inference*, 5(2):210–249, 2016.

G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *Int. Conf. Machine Learning*, pages 1247–1255, 2013.

F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

F. R. Bach and M. I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Dept. of Statistics, University of California, Berkeley, 2005.

A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. 2016.

X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. arXiv:1606.03657 [cs.LG], 2016.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. pages 2672–2680, 2014.

K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. DRAW: A recurrent neural network for image generation. In *Int. Conf. Machine Learning*, pages 1462–1471, 2015.

K. M. Hermann and P. Blunsom. Multilingual distributed representations without word alignment. In *Int. Conf. Learning Representations*, 2014. arXiv:1312.6173 [cs.CL].

H. Hermansky, D. P. W. Ellis, and S. Sharma. Tandem connectionist feature extraction for conventional HMM systems. In *IEEE Int. Conf. Acoustics, Speech and Sig. Proc.*, pages 1635–1638, 2000.

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

M. J. Huiskes and M. S. Lew. The mir flickr retrieval evaluation. In *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*, pages 39–43, 2008. doi: 10.1145/1460096.1460104. URL `http://doi.acm.org/10.1145/1460096.1460104`.

Y. Jia, M. Salzmann, and T. Darrell. Factorized latent spaces with structured sparsity. pages 982–990, 2010.

D. Kingma and J. Ba. ADAM: A method for stochastic optimization. In *Int. Conf. Learning Representations*, 2015.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. arXiv:1312.6114 [stat.ML], 2014.

D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. pages 3581–3589, 2014.

A. Klami, S. Virtanen, and S. Kaski. Bayesian canonical correlation analysis. *Journal of Machine Learning Research*, pages 965–1003, 2013.

P. L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *Int. J. Neural Syst.*, 10 (5):365–377, 2000.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

A. Lu, W. Wang, M. Bansal, K. Gimpel, and K. Livescu. Deep multilingual correlation for improved word embeddings. In *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT 2015)*, 2015.

A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow. Adversarial autoencoders. In *Int. Conf. Learning Representations*, 2016.

B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions on circuits and systems for video technology*, 11(6):703–715, 2001.

T. Melzer, M. Reiter, and H. Bischof. Nonlinear feature extraction using generalized canonical correlation analysis. In *Int. Conf. Artificial Neural Networks*, pages 353–360, 2001.

R. Memisevic, L. Sigal, and D. J. Fleet. Shared kernel information embedding for discriminative inference. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(4):778–790, 2012.

V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Int. Conf. Machine Learning*, pages 807–814, 2010.

J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Ng. Multimodal deep learning. In *Int. Conf. Machine Learning*, pages 689–696, 2011.

A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely. The Kaldi speech recognition toolkit. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Int. Conf. Machine Learning*, pages 1278–1286, 2014.

M. Salzmann, C. H. Ek, R. Urtasun, and T. Darrell. Factorized orthogonal latent spaces. In *Int. Workshop on Artificial Intelligence and Statistics*, pages 701–708, 2010.

K. Sohn, W. Shang, and H. Lee. Improved multimodal deep learning with variation of information. pages 2141–2149, 2014.

K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. pages 3465–3473, 2015.

N. Srivastava and R. Salakhutdinov. Multimodal learning with deep boltzmann machines. *Journal of Machine Learning Research*, 15:2949–2980, 2014.

N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

L. J. P. van der Maaten and G. E. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

S. Virtanen, A. Klami, and S. Kaski. Bayesian CCA via group sparsity. In *Int. Conf. Machine Learning*, pages 457–464, 2011.

C. Wang. Variational Bayesian approach to canonical correlation analysis. *IEEE Trans. Neural Networks*, 18(3):905–910, 2007.

W. Wang and K. Livescu. Large-scale approximate kernel canonical correlation analysis. In *Int. Conf. Learning Representations*, 2016. arXiv:1511.04773 [cs.LG].

W. Wang, R. Arora, K. Livescu, and J. Bilmes. Unsupervised learning of acoustic features via deep canonical correlation analysis. In *IEEE Int. Conf. Acoustics, Speech and Sig. Proc.*, 2015a.

W. Wang, R. Arora, K. Livescu, and J. Bilmes. On deep multi-view representation learning. In *Int. Conf. Machine Learning*, pages 1083–1092, 2015b.

J. R. Westbury. *X-Ray Microbeam Speech Production Database User's Handbook Version 1.0*, 1994.

F. Yan and K. Mikolajczyk. Deep correlation for matching images and text. In *IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, pages 3441–3450, 2015.

S. J. Young, D. Kernshaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. The HTK book version 2.2. Technical report, Entropic, Ltd., 1999.

A ADDITIONAL T-SNE VISUALIZATION OF NOISY MNIST

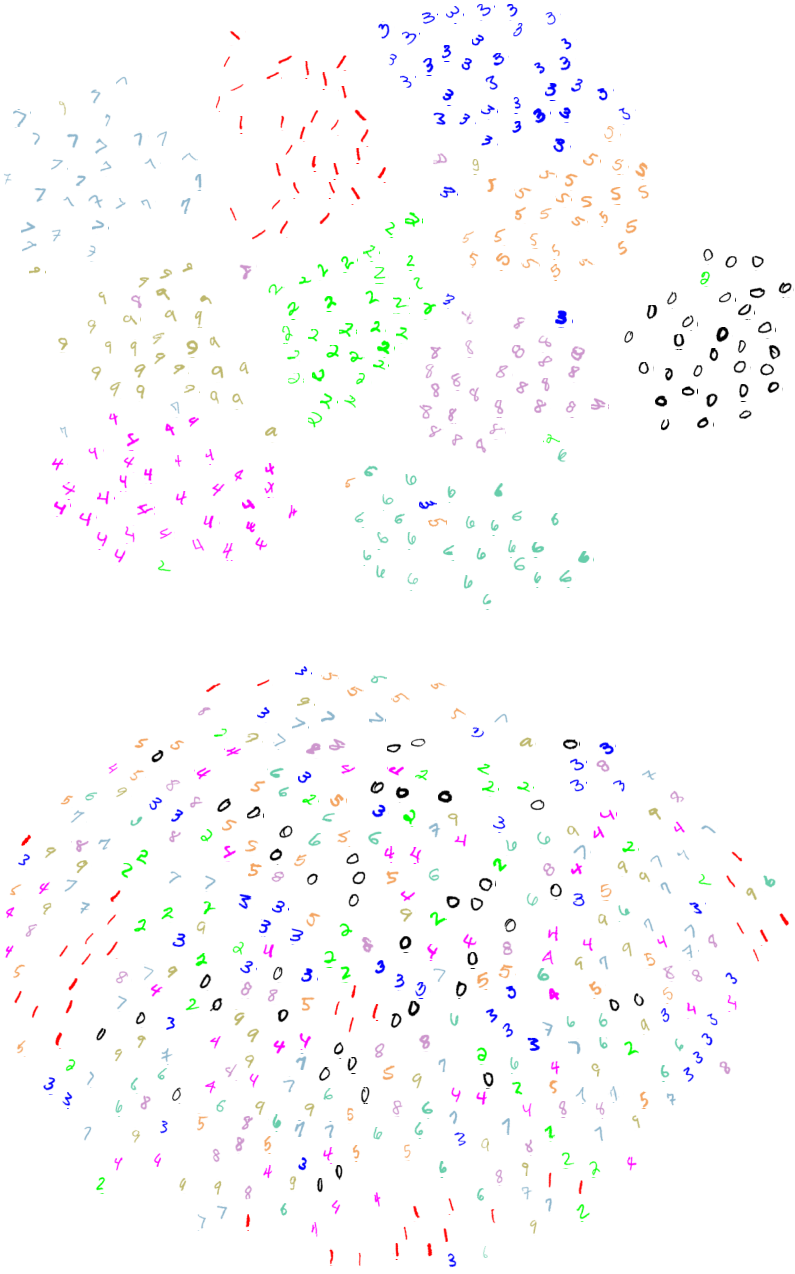


Figure 6: 2D t-SNE embedding of the shared variables $\mathbf{z} \in \mathbb{R}^{40}$ (top) and private variables $\mathbf{h}_x \in \mathbb{R}^{30}$ (bottom).