Variational Canonical Correlation Analysis

(VCCA)

And

Variational Canonical Correlation Analysis Private

(VCCA-Private)

Aditya Sharma(0801CS171008)
Ketan Likhi(0801CS171033)

December 13, 2020

Contents

Chapter 1

Introduction

1.1 Multi View Learning

Multi-view learning is an emerging direction in machine learning which considers learning with multiple views to improve the gener-alization performance. Multi-view learning is also known as data fusion or data integration from multiple feature sets.

1.2 CCA

Canonical correlation analysis (CCA) is a way of measuring the linear relationship between two multidimensional variables. It finds two bases,one for each variable,that are optimal with respect to correlations and, at the same time, it finds the corresponding correlations.In other words, it finds the two bases in which the correlation matrix between the variables is diagonal and the correlations on the diagonal are maximized. The dimensionality of these new bases is equal to or less than the smallest dimensionality of the two variables.

1.3 VCCA

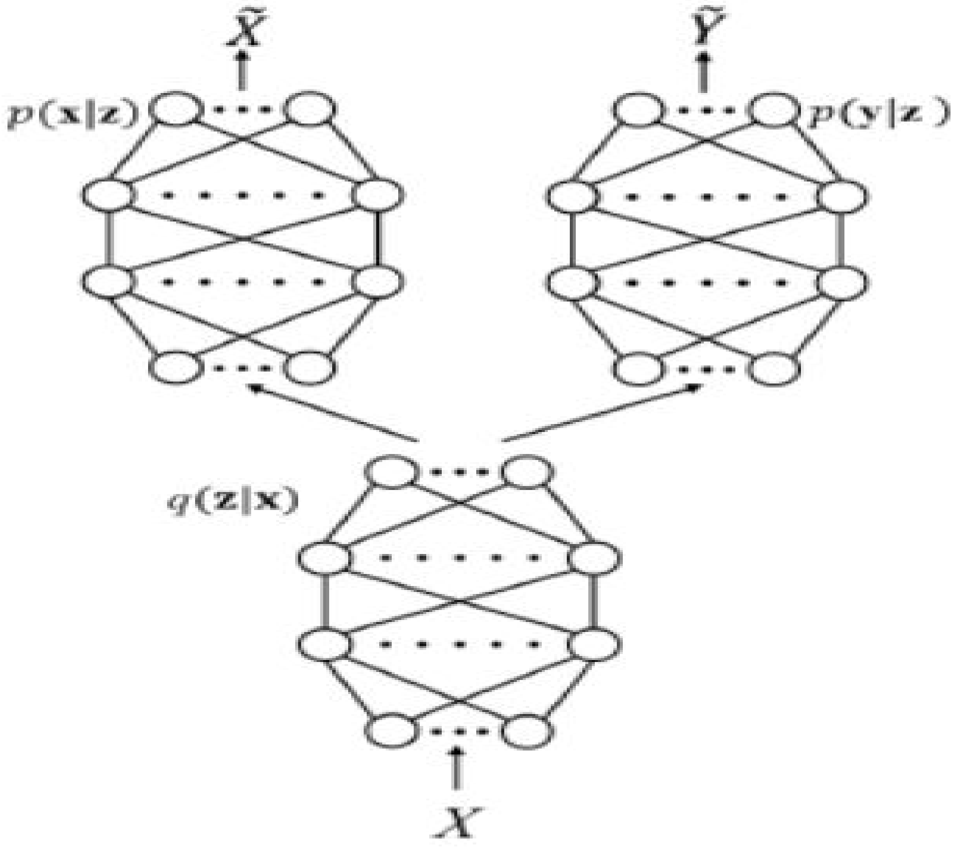
The need for VCCA arised from the fact that its predecessor DCCA had a disadvantage that it did not provide a model for generating samples from the latent space.

VCCA is a deep multi-view learning model that extends the latent variable model interpretation of linear CCA to nonlinear observation models parameterized by deep neural networks.

The VCCA is built on the foundational concepts of Variational Autoencoders and KL Divergence, before understanding the mathematical formulation of VCCA lets understand in brief what Variational Autoencoder and KL Divergence is.

A **Variational Autoencoder** can be defined as being an autoencoder whose training is regularised to avoid overfitting and ensure that the latent space has good properties that enable generative process**.**

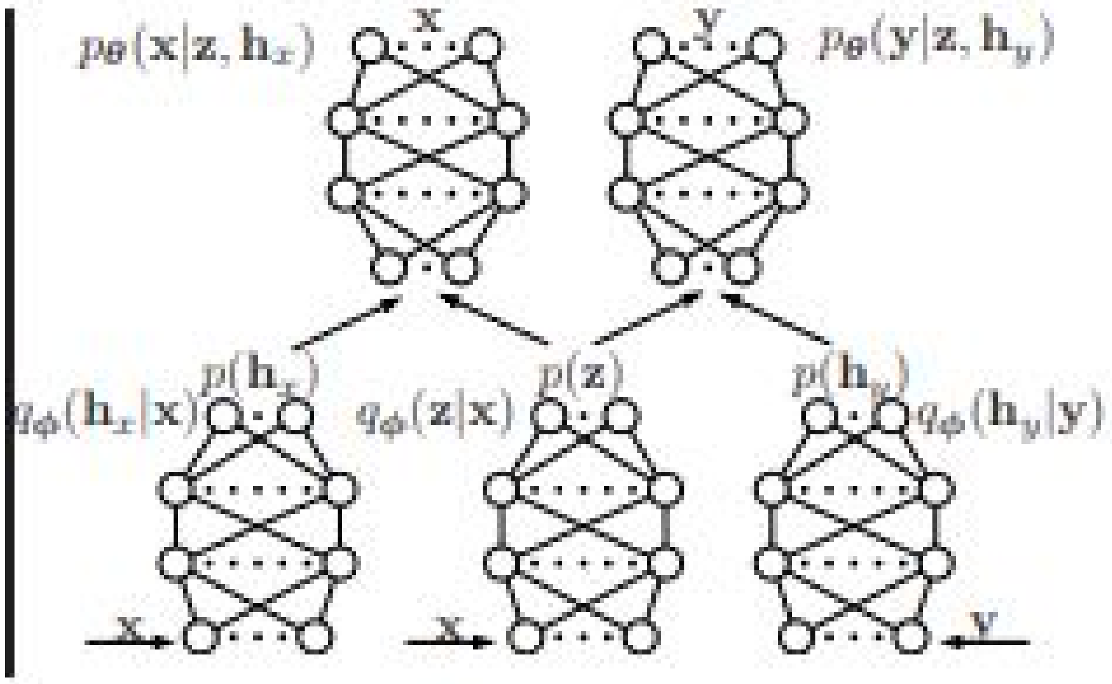
KL Divergence is a way of measuring the matching between two distributions(e.g. threads)


$\tilde{X}$
$\tilde{Y}$
$p(\mathbf{x}|\mathbf{z})$
$p(\mathbf{y}|\mathbf{z})$
$q(\mathbf{z}|\mathbf{x})$
X

SCHEMATIC DIAGRAM OF DVCCA

1.4 VCCA Private

A potential disadvantage of VCCA is that it assumes the common latent variables z are sufficient 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.


$p_\theta(\mathbf{x}|\mathbf{z},\mathbf{h}_x)$
$\mathbf{x}$
$\mathbf{y}$
$p_\theta(\mathbf{y}|\mathbf{z},\mathbf{h}_y)$
$p(\mathbf{h}_x)$
$p(\mathbf{z})$
$p(\mathbf{h}_y)$
$q_\phi(\mathbf{h}_x|\mathbf{x})$
$q_\phi(\mathbf{z}|\mathbf{x})$
$q_\phi(\mathbf{h}_y|\mathbf{y})$
$\mathbf{x}$
$\mathbf{x}$
$\mathbf{y}$

Schematic diagram DVVCA-Private

Chapter 2

Mathematical Formulation

2.1 Formulation for VCCA

VCCA explains CCA from the perspective of a probabilistic latent variable model. It assumes that the instances x and y from two views are independently conditioned on the multivariate latent variable $z \in R^{dz\times 1}$, and CCA aims to maximize the joint probability distribution of x and y:

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

Deep variational canonical correlation analysis (VCCA) was extended from variational auto-encoders (VAE) .First, the mean vector μi and the diagonal covariance matrix i of xi are calculated by the encoder f_X. Then, L instances, $\{z^{(l)}_i\}_{l=1}^{L}$, are randomly sampled from the distribution N (μi,i). Finally, decoders gx and gy reconstruct instances x and y, respectively. The objective function of VCCA is:

$$\min_{\mathbf{f}_x,\mathbf{g}_x,\mathbf{g}_y} \frac{1}{N}\sum_{i=1}^{N} D_{KL}\left(q(\mathbf{z}_i|\mathbf{x}_i)||p(\mathbf{z}_i)\right) + \qquad (
$$
$$\frac{\lambda}{NL}\sum_{i=1}^{N}\sum_{l=1}^{L}\left(\log p\left(\mathbf{x}_i|\mathbf{z}_i^{(l)}\right) + \log p\left(\mathbf{y}_i|\mathbf{z}_i^{(l)}\right)\right)$$

where the first term denotes the Kullback-Leibler (KL) divergence between the posterior distributions q(zi|xi) and p (zi), and the latter two terms denote the expectations of the log likelihood under the approximate posterior distributions, which are equivalent to the reconstruction error. In testing, the mean vector μ_i is used as the input features.

2.2 Formulation for VCCA - Private

Under this model, the data likelihood is defined by

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

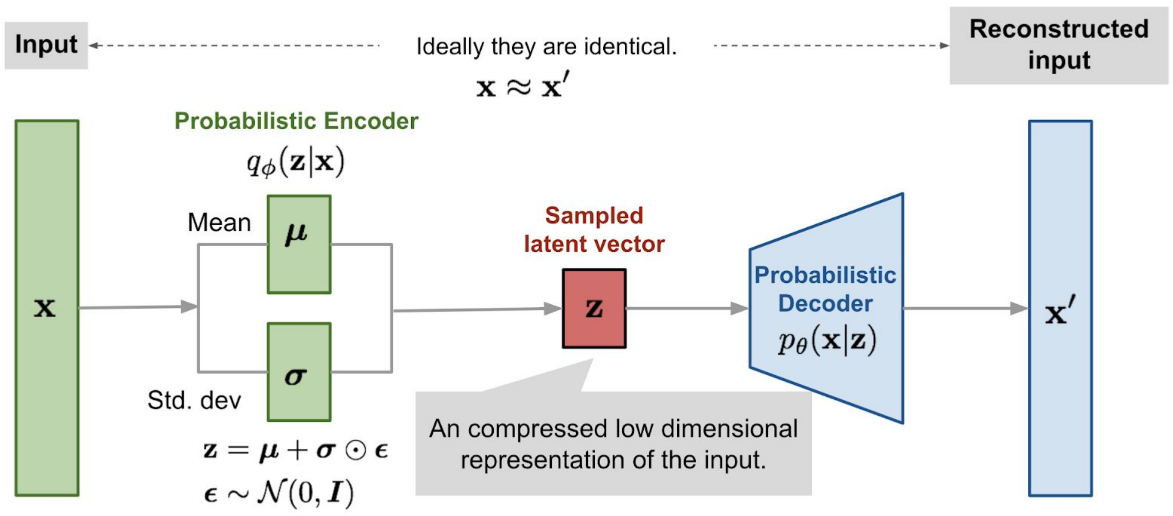
$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) = \iiint p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{h}_x, \mathbf{h}_y) d\mathbf{z}\, d\mathbf{h}_x\, d\mathbf{h}_y. \tag{8}$$

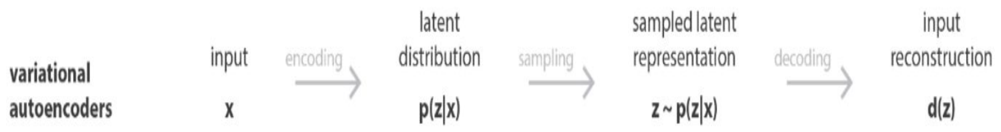
Variational lower bound on the data log likelihood for VCCA private is :

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) \geq \mathcal{L}_{\text{private}}(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}, \boldsymbol{\phi}) := -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \\ -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{h}_x|\mathbf{x})||p(\mathbf{h}_x)) - D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{h}_y|\mathbf{y})||p(\mathbf{h}_y)) \\ +\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}),\, q_{\boldsymbol{\phi}}(\mathbf{h}_x|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}, \mathbf{h}_x)\right] \\ +\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}),\, q_{\boldsymbol{\phi}}(\mathbf{h}_y|\mathbf{y})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z}, \mathbf{h}_y)\right]. \tag{10}$$

In particular, we draw samples of z and hx from their corresponding approximate posteriors, and concatenate their samples as inputs to the DNN parameterizing p(x|z, hx).

2.3 Variational Autoencoder


Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
Probabilistic Encoder
$q_\phi(\mathbf{z}|\mathbf{x})$
Mean
$\boldsymbol{\mu}$
Std. dev
$\boldsymbol{\sigma}$
$\mathbf{x}$
Sampled latent vector
$\mathbf{z}$
Probabilistic Decoder
$p_\theta(\mathbf{x}|\mathbf{z})$
$\mathbf{x}'$
$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$
$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$
An compressed low dimensional representation of the input.



The loss function for Variational Autoencoder is calculated as:

$$\text{loss} = C \,\| x - \hat{x} \|^2 + \text{KL}[\, N(\mu_x, \sigma_x), N(0, I)\,] = C \,\| x - f(z) \|^2 + \text{KL}[\, N(g(x), h(x)), N(0, I)\,]$$

2.4 KL Divergence

The Kullback-Leibler Divergence score, or KL divergence score, quantifies how much one probability distribution differs from another probability distribution.

The KL divergence between two distributions Q and P is often stated using the following notation:

- KL(P || Q)

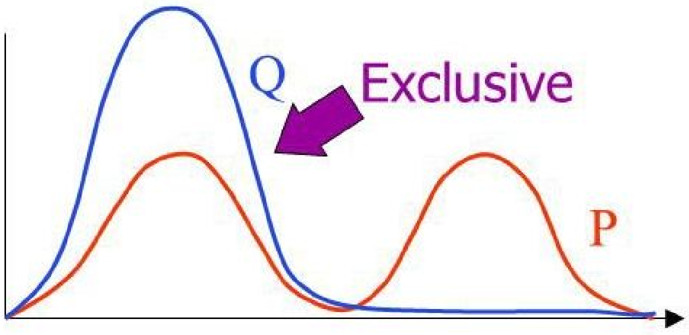
Where the “||” operator indicates “*divergence*” or Ps divergence from Q.

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

$$D_{KL}(P||Q) = \int P(x) \log \frac{P(x)}{Q(x)} dx$$

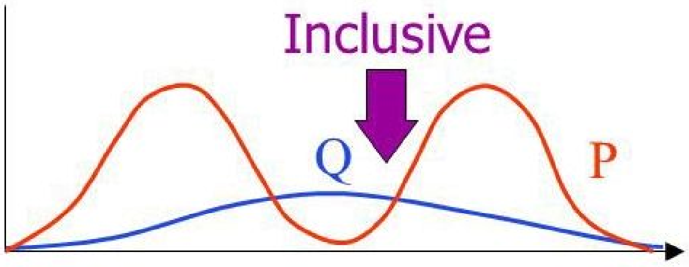
Minimising
KL(*Q*||*P*)

$$= \sum_H Q(H) \ln \frac{Q(H)}{P(H|V)}$$



Minimising
KL(*P*||*Q*)

$$= \sum_H P(H|V) \ln \frac{P(H|V)}{Q(H)}$$



Chapter 3 ALGORITHMS

3.1 ALGORITHM (VCCA)

Input : vectors X and Y of dimensionality input_dim respectively
Output: Representation of views in a latent space of given dimension(ZDIMS).

1) Input is first passed to the fit method, which trains the variational autoencoder(VAE), i.e. our neural network,
2) Cross-entropy loss is calculated between the original vector and the produced vector.
3) The weights and biases are adjusted through back backpropagation.
4) Steps 2-3 are repeated for the whole dataset according to the number of epochs set.
5) New data when given as input in transform function is transformed into 2 vectors of bottleneck dimensions(i.e. Latent space view) and is returned.

3.2 ALGORITHM (VCCA-Private)

Input : vectors X and Y of dimensionality input_dim respectively
Output: Representation of views in a latent space of given dimension(ZDIMS+PDIMS).

1) Input X and Y are first passed to the fit method, which trains our variational autoencoders(VAE), i.e. our neural network,
2) Cross-entropy loss is calculated between the original vectors and the produced vectors.
3) The weights and biases are adjusted through back backpropagation.
4) Steps 2-3 are repeated for the whole dataset according to the number of epochs set.
5) New data when given as input in transform function is transformed into 2 vectors of bottleneck dimensions(i.e. Latent space view) and is returned.

Chapter 4

Documentation of API

4.1 Package organization

A. Files in VCCA:

a. **vcca.py** → The file containing the fit() method, loss() method, and finally transform() method.

b. **autoencoder.py** → The file containing the Autoencoder class, which inherits from nn.module of pytorch, which indeed is our neural network.

B. Files in VCCA-Private:

a. **vcca_pvt.py** → The file containing the fit() method, loss() method, and finally transform() method.

b. **autoencoder_pvt.py** → The file containing the Autoencoder class, which inherits from nn.module of pytorch, which indeed is our neural network.

C. Common Files:

a. **utils.py** → Contains the code to concat two datasets

b. **code.py** → The file to test run the VCCA code.

Parameters in VCCA():

1. **x_view**- An array of multiple vectors of the same dimension(input_dim).
2. **y_view-** An array of multiple vectors of the same dimension(input_dim).
3. **epochs**- no of times whole training data has to be trained.
4. **ZDIMS**- Size of the reduced vector(z)(ZDIMS<=input_dim).
5. **input_dim**- dimensions of input vector.

Parameters in VCCAPrivate():

All parameters are **common** with **VCCA** and 1 new parameter is also there:

1. **PDIMS-** Size of private variable vector.

Methods for users:

fit()→ Takes in 2 argument vectors or array of vectors each of dimension(input_dim).

transform() → Takes 2 input arguments(vectors which need to be transformed to bottleneck dimensions), returns 2 vectors of size ZDIMS(VCCA), ZDIMS+PDIMS(VCCA Private)

Chapter 5 Examples

5.1 EXAMPLE (VCCA)

Code:

```
from vcca import VCCA
import torch
from torch.autograd import Variable

x_view = Variable(torch.randn(10000, 100))
y_view = Variable(torch.randn(10000, 100))
epochs=2
vcca=VCCA(epochs,batch_size=120,ZDIMS=30,input_dim=100)
vcca.fit(x_view,y_view)
x,y=vcca.transform(x_view,y_view)
print(x.shape,y.shape)
```

Output:

```
torch.Size([10000, 30]) torch.Size([10000, 30])
```

5.2 Example(VCCA Private)

Code:

```
from vcca_pvt import VCCAPrivate
import torch
from torch.autograd import Variable


x_view = Variable(torch.randn(10000, 100))
y_view = Variable(torch.randn(10000, 100))
epochs=1
vccaPvt=VCCAPrivate(epochs,batch_size=120,ZDIMS=30,PDIMS=30,input
_dim=100)
vccaPvt.fit(x_view,y_view)
x_latent,y_latent = vccaPvt.transformPrivate(x_view,y_view)
print(f"x_latent  shape  =  {x_latent.shape},y_latent  shape  =
{y_latent.shape}")
```

OUTPUT:

```
x_latent shape = torch.Size([10000, 60]),y_latent shape = torch.Size([10000, 60])
```

Chapter 6

Learning Outcome

6.1 Learnt about multiview learning, and how probability distribution can further enhance the CCA, and give a new aspect to it.

6.2 Variational Autoencoders as neural networks, deep generative model.

6.3 How KL divergence can help find the similarity between distributions.

6.4 How VCCA itself might fail in giving the desired regeneration and how there was a need for VCCA private.

6.4 Learnt about the team work, and the way to analyse a research paper.

Appendix A

References

- Wang, Weiran et al. “Deep Variational Canonical Correlation Analysis.” *ArXiv* abs/1610.03454 (2016): n. Pag.
- Guo, Chenfeng and Dongrui Wu. “Canonical Correlation Analysis (CCA) Based Multi-View Learning: An Overview.” *ArXiv* abs/1907.01693 (2019): n. pag.