Variational Canonical Correlation Analysis

(VCCA)

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

December 13, 2020

Contents

1 Introduction 2

1.1 Multiview Learning 2

1.2 VCCA......................... 2

2 Mathematical Formulation 3

2.1 Formulation 3

2.2 Universe subspace 4

3 Algorithm 5

3.1 VCCA algorithm 5

4 Documentation of API 7

4.1 Package organization 7

4.2 Methods 8

5 Example 12

5.1 Example 1 12

5.2 Example 1 12

6 Learning Outcome 13

6.1 13

6.2 13

A References 14

Chapter 1

Introduction

1.1 Multiview 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.

1.2 VCCA

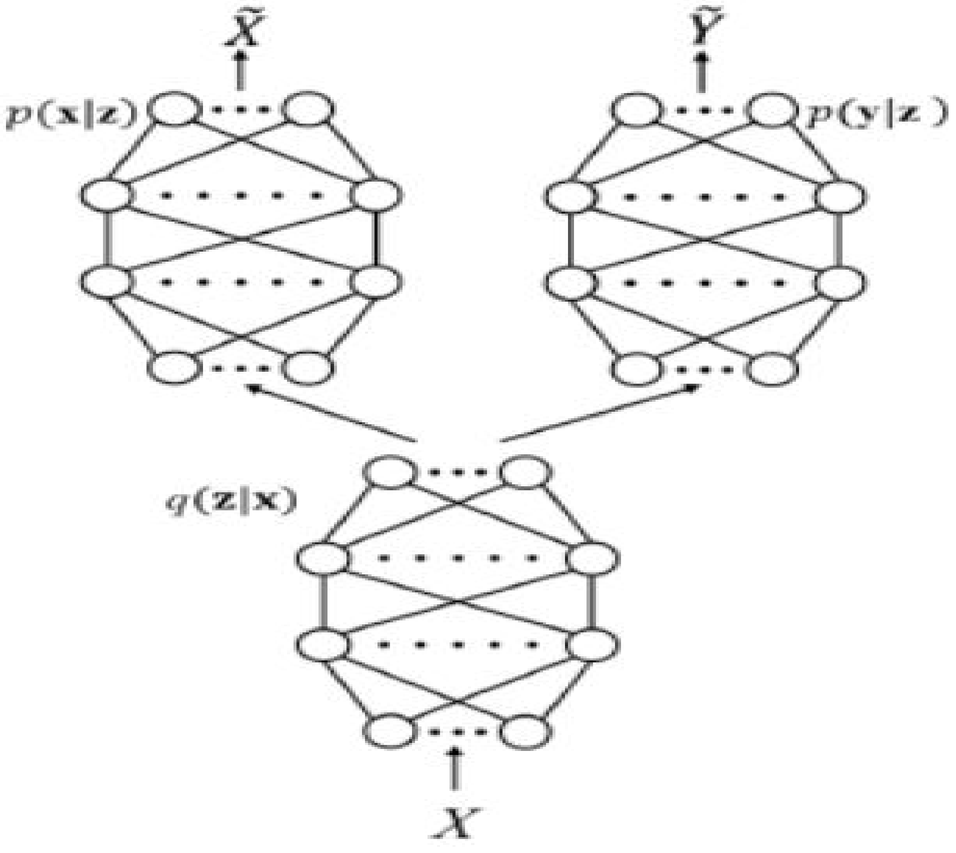
The need for VCCA arised from the fact that its predecessor DCCA had a disadvantage that it did not provide a model for generating samples from the latent space.

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

Chapter 2

Mathematical Formulation

2.1 Formulation

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 ∈ $R^{dz\times1}$, and CCA aims to maximize the joint probability distribution of x and y:

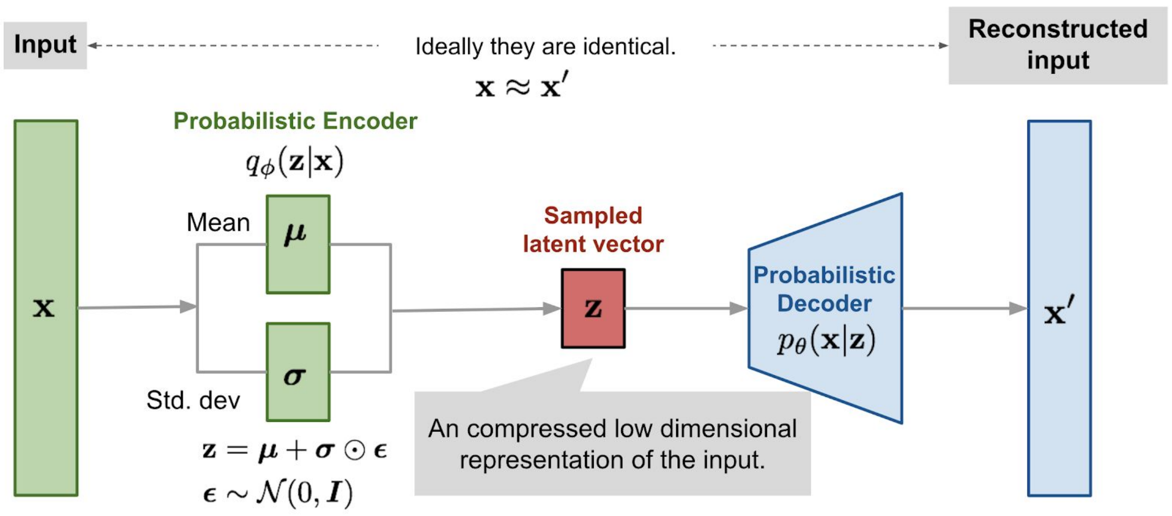
$$p(\mathbf{x},\mathbf{y},\mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})p(\mathbf{y}|\mathbf{z}),$$
$$p(\mathbf{x},\mathbf{y}) = \int p(\mathbf{x},\mathbf{y},\mathbf{z})d\mathbf{z}.$$

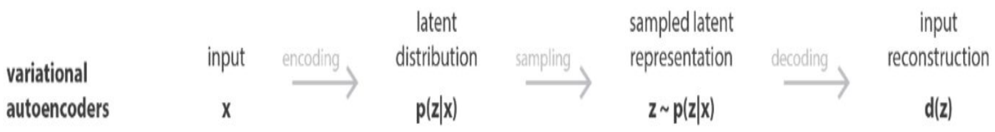
Deep variational canonical correlation analysis (VCCA) was extended from variational auto-encoders (VAE) .First, the mean vector μi and the diagonal covariance matrix i of xi are calculated by the encoder f_X. Then, L instances, $\{z^{(l)}_i\}_{l=1}^{L}$, are randomly sampled from the distribution N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

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

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 Variational Autoencoder


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



The loss function for Variational Autoencoder is calculated as:

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

2.3 KL Divergence

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

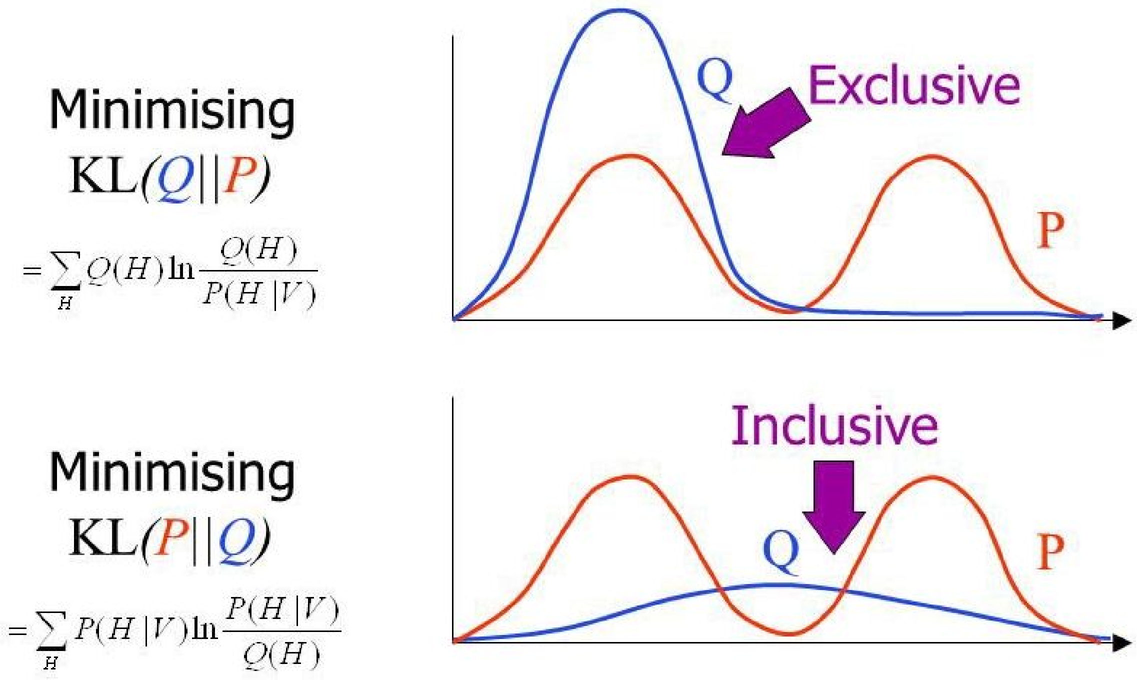
The KL divergence between two distributions Q and P is often stated using the following notation:

- 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

ALGORITHM

Input : Two vectors X and Y of dimensionality d1 and d2 respectively, which are 2 different views.
Output: Representation of views in a latent space of given dimension.

1) Input is first passed to the fit method, which trains the variational autoencoder, 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) Random data is sampled of the dimensions of the bottleneck, and the corresponding output is checked, after performing the epochs selected.

Chapter 4

Documentation of API

4.1 Package organization

Vcca.py → The file containing the fit() method and various parameters being set for the neural networks training

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

Utils.py → Contains the code to concat two datasets

Code.py → The file to test run the VCCA code

Parameters in vcca.py

1. **x_view**- vector showing a view.
2. **y_view**- vector showing a view.
3. **epochs**- no of times whole training data has to be trained.
4. **ZDIMS**- Dimensions of the reduced vector(ZDIMS<=input_dims).
5. **input_dim**- dimensions of input vector.

Methods in vcca.py

fit() → Takes in the arguments as the first view, second view, bottleneck dimension, and number of epochs, and trains our neural network on the dataset provided.

regenerate() → Returns an regenerated view from the random input taken i.e array of bottleneck dimensions and presents a view.

EXAMPLE

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

x_view = Variable(torch.randn(10000, 100))
y_view = Variable(torch.randn(10000, 100))
epochs=1
fit(x_view,y_view,ZDIMS=10,input_dim=100,epochs)
```

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 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.