CCA to Normilizing Flows

Boeva Galina, group M05-304a

October 4, 2024

Introduction

When observations consist of multiple views or modalities of the same underlying source of variation, a learning algorithm should efficiently account for the complementary information to alleviate learning difficulty [1] and improve accuracy. A well-established method for two-view analysis is given by canonical correlation analysis (CCA) [2], a classical subspace learning technique that extracts the common information between two multivariate random variables by projecting them onto a subspace. CCA, as a standard model for unsupervised two-view learning, has been used in a broad range of tasks such as dimensionality reduction, visualization and time series analysis [3]. A modified formulation of probabilistic CCA is presented, then this linear probabilistic layer is extended to an interpretable deep generative multi-view network. The proposed model captures the variations of the views by a shared latent representation, describing the common underlying sources of variation, i.e. the essence of multi-view data, and a set of view-specific latent factors.

Probabilistic CCA

The probabilistic generative model for the graphical model in Figure 1.1 is defined as:

$$\phi \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{I}_{d_0}), 0 < d_0 \leqslant \min(d_1, d_2) \tag{1}$$

$$\boldsymbol{z_1}|\phi \sim \mathcal{N}(\boldsymbol{W_1}\phi + \boldsymbol{\mu}_{\varepsilon_1}, \boldsymbol{\Psi}_1), \boldsymbol{W_1} \in \mathbb{R}^{d_1 \times d_0}, \boldsymbol{\Psi}_1 \succeq 0 \tag{2}$$

$$\boldsymbol{z_2}|\phi \sim \mathcal{N}(\boldsymbol{W_2}\phi + \boldsymbol{\mu}_{\varepsilon_2}, \boldsymbol{\Psi}_2), \boldsymbol{W_2} \in \mathbb{R}^{d_2 \times d_0}, \boldsymbol{\Psi}_2 \succeq 0 \tag{3}$$

where ϕ is the shared latent representation. The maximum likelihood estimate of the parameters of this model can be expressed in terms of the canonical correlation directions as:

$$\hat{\boldsymbol{W}}_1 = \boldsymbol{\Sigma}_{11}\boldsymbol{U}_1\boldsymbol{M}, \hat{\boldsymbol{W}}_2 = \boldsymbol{\Sigma}_{22}\boldsymbol{U}_2\boldsymbol{M}$$

$$\hat{\boldsymbol{\Psi}}_1 = \boldsymbol{\Sigma}_{11} - \hat{\boldsymbol{W}}_1\hat{\boldsymbol{W}}_1^T, \hat{\boldsymbol{\Psi}}_2 = \boldsymbol{\Sigma}_{22} - \hat{\boldsymbol{W}}_2\hat{\boldsymbol{W}}_2^T$$

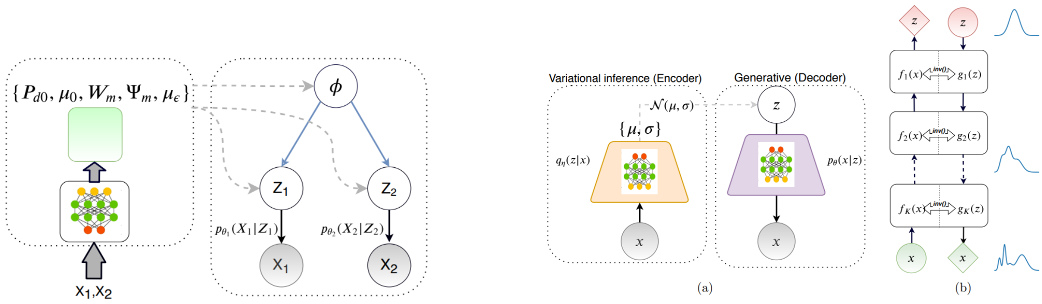

Figure 1: 1. Graphical representation of the deep probabilistic CCA model, where the blue edges belong to latent linear probabilistic CCA model and the black edges represent the deep nonlinear observation networks (decoders) $p_{\theta_m}(\mathbf{x}_m|\mathbf{z}_m) = g_m(\mathbf{z}_m;\theta_m)$. Shaded nodes denotes observed views and dashed line represent the stochastic samples drawn from the approximate posteriors. 2. Schematic representation of (a) a vanilla Variational Auto-Encoder model, and (b) a Normalizing Flow model.

$$\hat{\boldsymbol{\mu}}_{\varepsilon_1} = \boldsymbol{\mu}_1 - \hat{\boldsymbol{W}}_1\boldsymbol{\mu}_0, \hat{\boldsymbol{\mu}}_{\varepsilon_2} = \boldsymbol{\mu}_2 - \hat{\boldsymbol{W}}_2\boldsymbol{\mu}_0$$

where $\boldsymbol{M} = \boldsymbol{P}_{d_0}^{1/2}\boldsymbol{R}$ is the square root of matrix $\boldsymbol{P}_{d_0}$ and $\boldsymbol{R}$ is an arbitrary rotation matrix and the residual errors terms can be defined as $\boldsymbol{\varepsilon}_1 := \boldsymbol{z}_1 - \boldsymbol{W}_1\phi$ and $\boldsymbol{\varepsilon}_2 := \boldsymbol{z}_2 - \boldsymbol{W}_2\phi$. This probabilistic graphical model induces conditional independence of $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ given ϕ. The parameter $\boldsymbol{\mu}_0$ is not identifiable by maximum likelihood.

In contrast to the results in [4], where $\boldsymbol{\mu}_0 = 0$, here we introduce $\boldsymbol{\mu}_0$ as an extra degree of freedom.

Normalizing Flows

Another line of work [5] that has received a large amount of interest recently is to directly estimate the distribution of the data by normalizing flows. The normalizing flow is a chain of smooth and invertible transformations (bijections) to construct a complex probability density by transforming a simple base density, such as a standard normal distribution, exploiting the change of variable formula. Given a random variable $\boldsymbol{z} \sim p(\boldsymbol{z})$ and an invertible and differentiable mapping $g : \mathbb{R}^n \to \mathbb{R}^n$, with inverse mapping $f = g^{-1}$, the probability density function of the transformed variable $\boldsymbol{x} = g(\boldsymbol{z})$ can be described by the change of variable formula as

$$p(\boldsymbol{x}) = p(\boldsymbol{z})\left|\det \boldsymbol{J}_g\right|^{-1} = p(f(\boldsymbol{x}))\left|\det \boldsymbol{J}_f\right| \tag{4}$$

This formula provides a framework for probabilistic generative modeling.

Authors of the NICE [6] model proposed using the following family of transformations for g_θ:

$$\boldsymbol{x} = g_\theta(\boldsymbol{z}) = \begin{cases} \boldsymbol{x}_{1:d} = \boldsymbol{z}_{1:d} \\ \boldsymbol{x}_{d+1:n} = \boldsymbol{z}_{d+1:n} + m_\theta(\boldsymbol{z}_{1:d}), \end{cases}$$

where $1 < d < n$, and m_θ is an arbitrary neural network with d inputs and $n - d$ outputs. This transformation is called additive coupling.

The inverse transformation is computed with the same ease, and the Jacobian is equal to 1. That is,

$$p_{\boldsymbol{x}}(\boldsymbol{x}) = p_{\boldsymbol{z}}(g^{-1}(\boldsymbol{x})),$$

which is a fairly strong constraint on the model.

Furthermore, since

$$\boldsymbol{x}_{1:d} = \boldsymbol{z}_{1:d},$$

the first d channels of the vector $\boldsymbol{x}$ coincide with the coordinates of the normal noise $\boldsymbol{z}$, meaning that these channels of $\boldsymbol{x}$ are not modeled. Due to this, the expressive power of the NICE model was relatively low.

Later, the authors of NICE proposed using fixed permutations of features/channels $\boldsymbol{x}$ between layers of normalizing flows, which became the basis for the work on RealNVP. Using permutations allows all output channels to be affected by the transformation $g_\theta(\boldsymbol{z})$; moreover, the gradient of the permutation is easily computed.

$$\boldsymbol{x} = g_\theta(\boldsymbol{z}) = \begin{cases} \boldsymbol{x}_{1:d} = \boldsymbol{z}_{1:d} \\ \boldsymbol{x}_{d+1:n} = \exp(s_\theta(\boldsymbol{z}_{1:d})) \odot \boldsymbol{z}_{d+1:n} + m_\theta(\boldsymbol{z}_{1:d}), \end{cases}$$

where $\odot$ denotes element-wise multiplication, and s_θ is a neural network that can be arbitrary but is usually chosen to have the same architecture as m_θ. This transformation is called affine coupling.

The resulting mapping is also easily inverted, and its Jacobian is equal to:

$$\det(\boldsymbol{J}_{g^{-1}}) = \exp\left(\sum_{i=d+1}^{n} (s_\theta(\boldsymbol{z}_{1:d}))_i\right)$$

Note that, as in the case of additive coupling, a significant portion of the channels remains unchanged when using affine coupling. To ensure that the transformation $g_\theta(\boldsymbol{x})$ models the distribution of $\boldsymbol{x}$ in all channels, different subsets of d channels are left unchanged on different layers.

Conclusion

This work presents a modified formulation of probabilistic canonical correlation analysis (CCA) and extends it to an interpretable deep generative multi-view network. Additionally, the integration of normalizing flows enhances the model's ability to estimate complex data distributions, providing a robust framework for probabilistic generative modeling.

References

[1] Arjun Chaudhuri and Mark Ligas. Consequences of value in retail markets. *Journal of retailing*, 85(3):406–419, 2009.

[2] Harold Hotelling. Relations between two sets of variates. In *Breakthroughs in statistics: methodology and distribution*, pages 162–190. Springer, 1992.

[3] Fengnian Xia, Han Wang, Di Xiao, Madan Dubey, and Ashwin Ramasubramaniam. Two-dimensional material nanophotonics. *Nature photonics*, 8(12):899–907, 2014.

[4] Francis R Bach and Michael I Jordan. A probabilistic interpretation of canonical correlation analysis. 2005.

[5] Mahdi Karami. Advances in probabilistic generative models: Normalizing flows, multi-view learning, and linear dynamical systems. 2020.

[6] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.