Learning Semantically Meaningful Embeddings Using Linear Constraints

Shuyu Lin
University of Oxford
slin@robots.ox.ac.uk

Bo Yang
University of Oxford
bo.yang@exeter.ox.ac.uk

Robert Birke
ABB Corporate Research
robert.birke@ch.abb.com

Ronald Clark
Imperial College London
ronald.clark@imperial.ac.uk

Abstract

Learning an interpretable representation is an essential task in machine learning, as many fields, such as legislation and healthcare, require explainability in the decision-making process where costly consequences can be easily incurred. In this paper, we propose a simple embedding learning method that jointly optimises for an auto-encoding reconstruction task and for estimating the corresponding attribute labels associated with the raw data. We restrict the attribute estimation model to be linear, constraining the learnt embedding space to be close to the interpretable attribute space. As a result, we are able to interpret the learnt embedding as a mixture of different attributes, i.e. semantic information has been embedded in the latent representation. Furthermore, as the linear mapping is fully invertible, we are able to generate any data samples from a list of specified attributes.

1. Introduction

Recently, there has been an increasing interest in the machine learning community in learning an interpretable latent representation of high-dimensional raw data, such as natural images. A popular approach focuses on using disentanglement as a means to pursue interpretability for deep neural networks. The argument is that many generative factors of a system are independent and a set of disentangled explanatory factors can be discovered, then a one-to-one correspondence between the true generative factor and the discovered explanatory factor might be found and, hence, a human might be able to interpret these learnt factors.

A key issue of the above claim lies in the mismatch between the statistical patterns in a dataset and the well-established human concepts. A fully unsupervised learning approach is able to discover the statistical regularities of a dataset and, hence, derive a set of independent (disentangled) factors, but these factors are unlikely to be aligned perfectly with existing human concepts, because many human concepts, such as gender, hair length and cloth styles, are in fact strongly correlated (entangled). This can be shown in Figure 1 where disentangled factors of variations in human faces are learnt using β-VAE algorithm [7]; each learnt code (which is believed to be statistically uncorrelated) actually embeds multiple human concepts. For example, the codes interpreted as background brightness, hair colour and hair length factors are also strongly associated with gender. This indicates that disentanglement alone does not guarantee interpretability. In fact, we would like to point out that discovering an embedding that incorporates different human concepts in a completely unsupervised fashion (i.e. with no reference to any labels of the human concepts) is unlikely to succeed exactly due to the subtle but prevalent misalignment between the pure statistical patterns in the data and the human definitions of various concepts.

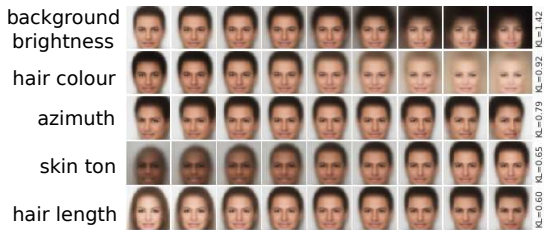


Figure 1. Many approaches have been proposed for learning interpretable representations (here we show β-VAE which aims to learn a disentangled representation – adapted from [3]). Notice that a single latent code does not correspond to a single human concept, such as the 'skin tone' code affects both skin tone and boldness.

In this paper, we consider the interpretability of a learnt latent representation as a key objective. Such latent representation is essential for applications in many domains such as health care, policy making and legislation, where explainability is a key design factor for any decision-making algorithms involved. In addition, organizing latent codes in

such a way allows us to select and reuse specific latent factors for different downstream tasks. To achieve our goal, we learn a smooth low dimensional latent embedding using a variational auto-encoder (VAE) [5] and constrain the learnt embedding by requiring it to be within a linear mapping away from a set of human understandable attribute labels. Such framework allows us to form an embedding space which is close to a human interpretable attribute space under a very simple and invertible transformation, leading to certain high-level abstract concepts to be encouraged to be embedded. At the same time, there is still sufficient flexibility in the model for low-level details to be abstracted away through the data-driven bottom-up reconstruction task.

In summary, our contributions are:

- **A simple (linear) relationship** is imposed between the embedding and the attribute which **acts as a constraint** to guide the network to learn an interpretable representation.
- The mapping between the attributes and the latent embedding is **invertible**, so samples that satisfy a requested list of attributes can be easily generated.
- This is **the first work that combines invertible and non-invertible branches** which allows both abstraction of the raw data as well as conversion between the learnt embedding and human concepts.

2. Our Proposal

Our proposed model is illustrated in Figure 2, where we inherit a simple VAE to learn a smooth latent embedding $\boldsymbol{z} \in R^{d_z}$ of any high dimensional input data $\boldsymbol{x} \in R^{d_x}$ and we require a list of d_a attributes $\boldsymbol{a}$ to be estimated under an additional linear mapping from $\boldsymbol{z}$. An attribute a_i can be represented as either a continuous value, such as temperature or age, or a discrete value, such as gender or emotion. We denote the model parameters in the encoder, the decoder and the linear attribute estimator as θ, ϕ and γ respectively. As for the learning objective, the reconstruction of the input data through the auto-encoder leads to a reconstruction error loss:

$$\mathcal{L}_{\mathrm{r}}(\theta, \phi) = |\boldsymbol{x} - \hat{\boldsymbol{x}}|^2, \tag{1}$$

where $\hat{\boldsymbol{x}}$ indicates the reconstructed data at the output of the auto-encoder. The attribute estimation leads to an attribute prediction loss. For a continuous attribute, the loss can be expressed as a simple least square error:

$$\mathcal{L}_{\mathrm{a}}^{i}(\theta, \gamma) = |a_i - \hat{a}_i|^2, \tag{2}$$

and for a discrete attribute as a cross entropy loss:

$$\mathcal{L}_{\mathrm{a}}^{i}(\theta, \gamma) = -\sum_{k=1}^{K_i} p_k \log q_k, \tag{3}$$

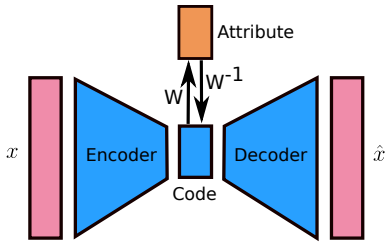


Figure 2. In addition to a common auto-encoder learning scheme, we add a light-weight attribute estimation module which is constrained to be a linear model. By jointly training the auto-encoder and the linear attribute estimator, we are able to learn an embedding that is optimally linearly separable for different attributes, leading to a semantically meaningful embedding.

where a_i is the ground truth attribute label, $\hat{a}_i$ is the predicted attribute label, p denotes a one-hot vector with 1 appearing at the index corresponding to the true category, q denotes an estimated probability distribution for all categories given by the linear estimator and K_i denotes the number of categories for the i-th attribute a_i.

The overall loss for our model is the sum of the reconstruction error given in Equation 1 and the attribution prediction error given in either Equation 2 or 3, i.e.

$$\mathcal{L}(\theta, \phi, \gamma) = \mathcal{L}_{\mathrm{r}}(\theta, \phi) + \sum_{i=1}^{d_a} \mathcal{L}_{\mathrm{a}}^{i}(\theta, \gamma). \tag{4}$$

By minimizing the loss $\mathcal{L}(\theta, \phi, \gamma)$ w.r.t. θ, ϕ and γ together, we are able to find a compact representation of the raw data while the representation is in a latent space that is closely related to the interpretable attributes, as the representation is optimized to be a linear mixture of the given attributes. The optimization of $\mathcal{L}(\theta, \phi, \gamma)$ can be done using gradient descent algorithm with ADAM optimizer. After the model has been trained, we are able to predict the attributes for any raw data. At the same time, as the linear attribute estimator can be easily inverted, we can also generate an input data sample given a list of specified attributes.

2.1. Direct Inference Using Specified Attributes

The invertability of the attribute estimator is a key component of our proposal, which enables direct inference of input samples given a list of specified attributes. Depending on whether the attributes are continuous or discrete, the inversion computation is different. Here we give details for both situations.

2.1.1 Continuous Attributes

For continuous attributes, the linear attribute estimator can be modelled as a matrix multiplication, illustrated in Figure 3, i.e.

$$\boldsymbol{W}\boldsymbol{z} + \boldsymbol{b} = \boldsymbol{a}, \tag{5}$$

where a weight matrix $\boldsymbol{W}^{d_a \times d_z} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_{d_a}]^T$ containing d_a weight vector $\boldsymbol{w}_i$s (each corresponds to the weights coupled with the latent embedding $\boldsymbol{z}$ to estimate a specific attribute value), and $\boldsymbol{b}^{d_a \times 1}$ is a bias vector. This linear estimator can be implemented using a neural network of a single fully connected layer with no activation. The inverse estimation of predicting a latent embedding $\boldsymbol{z}$ given an attribute vector $\boldsymbol{a}$ can be made by re-arranging Equation 5, giving

$$\boldsymbol{z} = (\boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T(\boldsymbol{a} - \boldsymbol{b}). \tag{6}$$

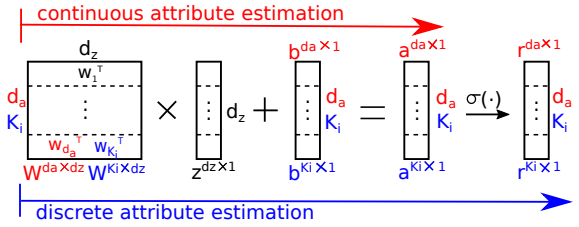


Figure 3. Computation of the linear attribute estimator for continuous (red) and discrete (blue) attributes. Discrete attributes require an additional softmax activation.

2.1.2 Discrete Attributes

The inversion process for discrete attributes is slightly less straight-forward. This is because for discrete attributes, we need to implement a multi-class classifier and this requires a softmax function $\sigma(\cdot)$ to be applied on the output vector $\boldsymbol{a}$ of the matrix multiplication defined in Equation 5. Note that the dimension of $\boldsymbol{a}$ is different in the continuous and discrete cases. As illustrated in Figure 3, we consider an attribute classification at a time and the output of the softmax $\boldsymbol{r} \in R^{K_i \times 1}$ represents a discrete probability distribution over all K_i categories for the current attribute. The category associated with the highest probability will be the prediction of the attribute estimator. The k-th dimension of $\boldsymbol{r}$ is defined as

$$r_k = \sigma(\boldsymbol{a})_k = \frac{\exp\left(\boldsymbol{w}_k^T\boldsymbol{z} + b_k\right)}{\sum_{j=1}^{K_i}\exp\left(\boldsymbol{w}_j^T\boldsymbol{z} + b_j\right)}. \tag{7}$$

It is clear that non-linearity has been introduced by the softmax activation, but we illustrate here that the inverse estimation from $\boldsymbol{r}$ to $\boldsymbol{z}$ can be computed in a closed form. Firstly, we compute the ratios between all $\boldsymbol{r}$'s elements and its final element, denoting the results as $\hat{\boldsymbol{r}}$. Then we have

$$\hat{r}_k = \frac{r_k}{r_{K_i}} = \begin{cases} \exp\left(\hat{\boldsymbol{w}}_k^T\boldsymbol{z} + \hat{b}_k\right), & 1 \le k \le K_i - 1 \\ 1, & k = K_i \end{cases}$$

where $\hat{\boldsymbol{w}}_k^T = \boldsymbol{w}_k^T - \boldsymbol{w}_{K_i}^T$ and $\hat{b}_k = b_k - b_{K_i}$. Now considering the first $(K_i - 1)$ elements of $\hat{\boldsymbol{r}}$ and taking the natural logarithm, we have $K_i - 1$ of the following equations:

$$\hat{\boldsymbol{w}}_k^T\boldsymbol{z} + \hat{b}_k = \ln\left(\hat{r_k}\right),\ 1 \le k \le K_i - 1. \tag{8}$$

We can organise them as a matrix equation similar to the one given in Equation 6, i.e. $\hat{\boldsymbol{W}}\boldsymbol{z} + \hat{\boldsymbol{b}} = \hat{\boldsymbol{r}}_{1:K_i-1}$, where $\hat{\boldsymbol{W}}^{(K_i-1)\times d_z} = [\hat{\boldsymbol{w}}_1, \cdots, \hat{\boldsymbol{w}}_{K_i-1}]^T$ and $\hat{\boldsymbol{b}}^{(K_i-1)\times 1} = [\hat{b}_1, \cdots, \hat{b}_{K_i-1}]^T$. Therefore, the latent embedding for a discrete attribute can be estimated from a discrete probability distribution by

$$\boldsymbol{z} = (\hat{\boldsymbol{W}}^T\hat{\boldsymbol{W}})^{-1}\hat{\boldsymbol{W}}^T\left(\hat{\boldsymbol{r}}_{1:K_i-1} - \hat{\boldsymbol{b}}\right). \tag{9}$$

3. Related Work

Visually grounded imagination is defined in [9] as the ability to generate images based on some given semantic concepts. Several works [10, 9, 8] have attempted this task in a VAE framework with the objective of learning a latent embedding that contains information of both the raw data (images) and the semantic attribute labels. [10] takes the semantic labels as a conditional input for the latent embedding, while [9, 8] learn a joint generative model shared between the images and semantic labels. Our work shares a similar objective, but differs in the inference process where an image is generated given a requested list of attributes. Our method has a closed-form inversion to complete the inference task, whereas all above works have to learn a separate inference model for such inference.

Flow-based generative models are proposed by [1, 2] to refer to generative models with invertible and easily differentiable transformations between the data and latent variable, naturally leading to the benefit of exact inference of a new data observation given a query latent embedding. [4] has shown that high-resolution realistic images can be generated using a well designed flow model. Our idea of keeping the transformation between the learnt embedding and the attribute space invertible and differentiable is inspired by Flow. However, our work differs in that we allow the transformation between the raw data and the embedding to be non-invertible, leaving more freedom for the low level details to be abstracted out. We believe a sensible combination between invertible and non-invertible functions is likely to lead to the optimal learning outcome, i.e. a latent embedding with preferrable properties.

4. Experiment

We carried out two experiments on the MNIST handwritten digit dataset [6] to demonstrate the performance of our method. Firstly, we show that learning through our method can form a latent embedding that is significantly more linearly separable for the human understandable semantic information, in comparison with a different learning approach where the latent embedding and the linear classifier are trained separately. As shown in Table 1, our

learning approach gives a clear better classification accuracy than both AE or VAE on a held-out test set of 10k images. Specifically, the huge gap of about 31% in the classification accuracy between our approach and the separate training approach in an AE case indicates that the additional attribute estimator acts to organize the latent space to be significantly more linearly separable. Furthermore, the significant improvement on accuracy from the variational training indicates that the smooth constraint imposed by the VAE objective helps to form an embedding close to the ideal attribute space and with our linear classification constraint we are able to push the learnt embedding one step closer. Moreover, learning with our approach can also improve the reconstruction quality of the latent embedding, as shown by the clear gap in Figure 4(b).

	Linear Classification Accuracy	
Training Method	AE	VAE
Separate	39.1 %	76.5%
Joint (Ours)	70.1%	79.1%

Table 1. Comparison in classification accuracy for MNIST dataset between our learning approach and a learning approach where the latent embedding and the linear classifier are trained separately. The result shows that learning using our approach forms a latent embedding that is closest to the ideal attribute space.

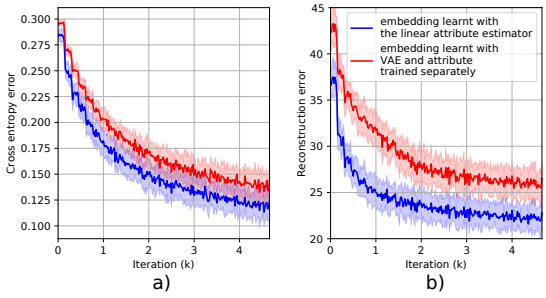
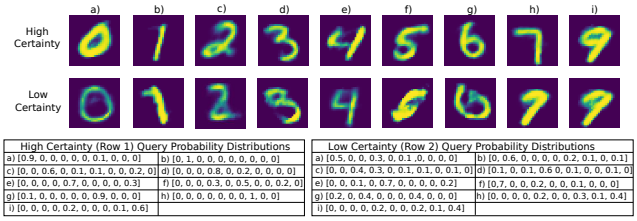


Figure 4. Comparison in validation error for the linear classification (a) and the VAE reconstruction (b) on MNIST dataset between our model (blue line) and a separate training learning approach (red line). The embedding learnt using our model is more linearly separable (hence, better embedding of the attribute information) and reaches a better reconstruction quality.

Secondly, we also demonstrate the ability of our method to generate input samples from user specified attribute labels. In the MNIST example, the learnt attribute is a discrete probability distribution over all 10 classes of digits. As shown in Figure 5, we list several examples from a set of randomly generated attribute requests using the inversion process offered by our approach. In cases where the requested distributions are certain about one class, the generated images show sensible forms of the corresponding digits. In cases where the distribution has uncertainty across several classes, the generated images are more like in transition states between the digits.



High Certainty (Row 1) Query Probability Distributions	
a) [0.9, 0, 0, 0, 0, 0, 0.1, 0, 0, 0]	b) [0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
c) [0, 0, 0.6, 0, 0.1, 0.1, 0, 0, 0.2, 0]	d) [0, 0, 0, 0, 0.8, 0, 0.2, 0, 0, 0, 0]
e) [0, 0, 0, 0, 0.7, 0, 0, 0, 0, 0.3]	f) [0, 0, 0, 0.3, 0, 0.5, 0, 0, 0.2, 0]
g) [0.1, 0, 0, 0, 0, 0, 0.9, 0, 0, 0]	h) [0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
i) [0, 0, 0, 0, 0.2, 0, 0, 0, 0.1, 0.6]	

Low Certainty (Row 2) Query Probability Distributions	
a) [0.5, 0, 0, 0.3, 0, 0.1, 0, 0, 0, 0]	b) [0, 0.6, 0, 0, 0, 0, 0, 0.2, 0.1, 0, 0.1]
c) [0, 0, 0.4, 0.3, 0, 0.1, 0.1, 0, 0.1, 0]	d) [0.1, 0, 0.1, 0.6 0, 0.1, 0, 0, 0.1, 0]
e) [0, 0, 0.1, 0, 0.7, 0, 0, 0, 0, 0.2]	f) [0.7, 0, 0, 0.2, 0, 0, 0.1, 0, 0, 0]
g) [0.2, 0, 0.4, 0, 0, 0, 0.4, 0, 0, 0]	h) [0, 0, 0, 0, 0.2, 0, 0, 0.3, 0.1, 0.4]
i) [0, 0, 0, 0, 0.2, 0, 0, 0.2, 0.1, 0.4]	

Figure 5. Generated images given a query attribute as a probability distribution over 10 digit categories. High certainty indicates the query distributions are peaked at one category, whereas low certainty refers to distributions activated over several categories.

5. Conclusion and Future Work

In this paper, we propose a novel learning approach to deriving a semantically meaningful latent embedding. We restrict the transformation between the latent embedding and the attribute space to be simple (linear) and this naturally forms an abstract latent representation that embeds the semantic information expressed in the attribute labels. Inference of different input samples given a list of request attributes is easy in our model, as the transformation between the latent embedding and the attributes is fully invertible. In the future, we would like to extend the linear transformation to more complex invertible mappings, allowing more flexibility in the semantic embedding.

References

[1] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation, 2014. 3

[2] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp, 2016. 3

[3] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018. 1

[4] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*, pages 10236–10245, 2018. 3

[5] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. 2

[6] Y. LeCun. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998. 3

[7] Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-vae: Learning basic visual concepts with a constrained variational framework. 2016. 1

[8] Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. Joint multimodal learning with deep generative models. *CoRR*, abs/1611.01891, 2016. 3

[9] Ramakrishna Vedantam, Ian Fischer, Jonathan Huang, and Kevin Murphy. Generative models of visually grounded imagination. *CoRR*, abs/1705.10762, 2017. 3

[10] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. *Lecture Notes in Computer Science*, 2016. 3