Deep Generative Models

Disentangled Representation Learning

Hamid Beigy

Sharif University of Technology

February 24, 2025

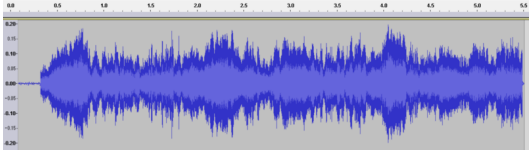
Table of contents

Introduction

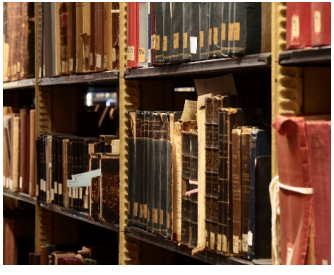
How do machine learning algorithms understand complex and unstructured inputs?

Computer vision

Computational speech



Natural language processing



Robotics

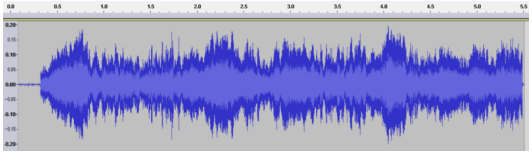
Credit: Aditya Grover

Representing input raw data

Computer vision

$\mathbf{x} = (x_1, \ldots, x_d)$

Computational speech



$\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ and $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})$

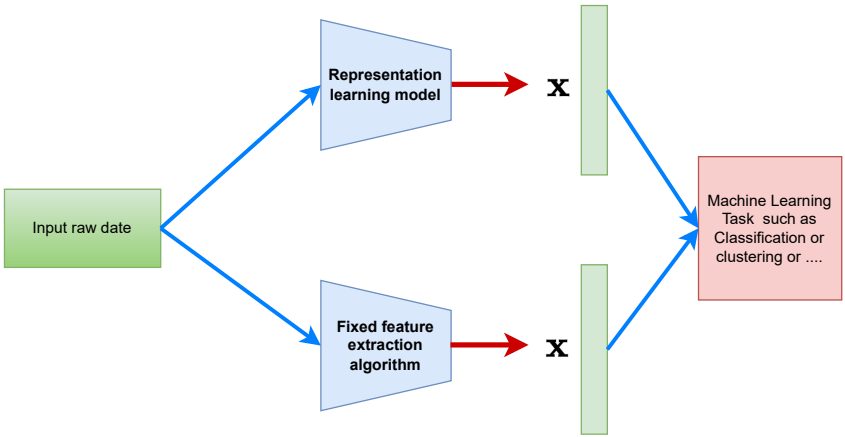
Natural language processing

$\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ and $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})$

Robotics

This process may be: handicrafted or **learning-based**.

Representation learning

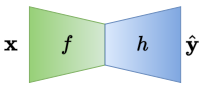


Representation Learning:

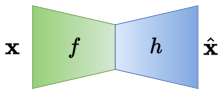
1. Supervised representation learning
2. Unsupervised representation learning

1. Let $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be the dataset, where
   - $\mathbf{x}_i \in \mathcal{X}$ is input raw data and
   - $y_i \in \mathcal{Y}$ is supervised signal
2. A supervised representation learning algorithm trains parameterized feature extractor f by solving a supervised task on D.
3. Feature extractor $f : \mathbb{R}^I \mapsto \mathbb{R}^d$ maps an input representation $\mathbf{x}$ to a feature representation $f(\mathbf{x}) \in \mathbb{R}^d$, where $d \ll I$.
4. Depending on the supervised task, an additional function, $h : \mathbb{R}^d \mapsto \mathbb{R}^O$, yields the output representation to evaluate a supervised objective function given feature representation $f(\mathbf{x})$.
5. The objective is to minimize training loss function $\hat{\mathbf{R}}(f, h)$ such as a cross-entropy loss to obtain pre-trained $\widehat{f}$ and $\widehat{h}$ as follows:

$$\widehat{f}, \widehat{h} = \arg\min_{f,h} \left\{ \hat{\mathbf{R}}(f, h) \right\}$$



1. Let $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be the dataset, where $\mathbf{x}_i \in \mathcal{X}$ is input raw data.
2. A unsupervised representation learning algorithm trains parameterized feature extractor f by solving an unsupervised task on D.
3. Feature extractor $f : \mathbb{R}^I \mapsto \mathbb{R}^d$ maps an input representation $\mathbf{x}$ to a feature representation $f(\mathbf{x}) \in \mathbb{R}^d$, where $d \ll I$.
4. For example, auto-encoders consider $h : \mathbb{R}^d \mapsto \mathbb{R}^I$ reconstruct the input as



5. Auto-encoders are trained by minimizing the following objective function.

$$\widehat{f}, \widehat{h} = \arg\min_{f,h} \left\{ \hat{\mathbf{R}}(f, h) \right\}$$

$$\hat{\mathbf{R}}(f, h) = \frac{1}{n} \sum_{i=1}^{n} \| h(f(\mathbf{x}_i)) - \mathbf{x}_i \|^2$$

What makes a representation good?

1. Priors for representation learning
2. Smoothness and the curse of dimensionality

   Smoothness is useful assumption but it is insufficient to deal with the curse of dimensionality because the number of up/down of the target functions may grow exponentially with the number of relevant interacting factors.
3. Distributed representations
4. Depth and Abstraction
   - Deep architectures promote the reuse of features
   - Deep architectures can potentially lead to progressively more abstract features at higher layers of representations
5. Disentangling Factors of Variation

Priors for representation learning

Why explicitly dealing with representations is interesting?

They can be convenient to express many general priors about the world around us (Bengio, Courville, and Vincent 2013).

Examples of such general-purpose priors are the following:

1. **Smoothness:** Function f (to be learned) is **smooth**, if $\mathbf{x} \approx \mathbf{y}$ implies $f(\mathbf{x}) \approx f(\mathbf{y})$.
2. **Multiple explanatory factors:** The data generating distribution is generated by different underlying **factors**.
3. **Hierarchical organization of explanatory factors:** The concepts that are useful for describing the world can be defined in terms of other concepts, in a hierarchy, with more abstract concepts higher in the hierarchy, defined in terms of less abstract ones.
4. **Semi-supervised learning:** With inputs $\mathbf{x}$ and target $\mathbf{y}$ to predict, a subset of the factors explaining $\mathbf{x}$'s distribution explain much of $\mathbf{y}$, given $\mathbf{x}$. Hence, representations that are useful for $p(\mathbf{x})$ tend to be useful when learning $p(\mathbf{y} \mid \mathbf{x})$.
5. **Shared factors across tasks:** With many $\mathbf{y}$s of interest or many learning tasks in general, tasks are explained by factors that are shared with other tasks.

Priors for representation learning (Cont.)

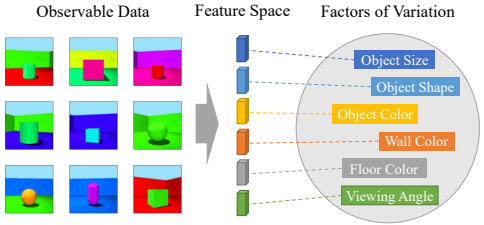
6. **Manifolds:** Probability mass concentrates near regions that have a much smaller dimensionality than the original space where the data live.
7. **Natural clustering:** Different values of categorical variables such as object classes are associated with separate manifolds.

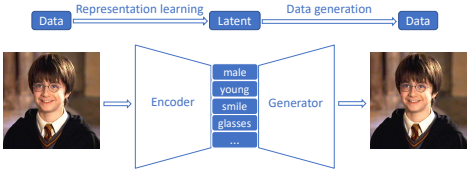
   More precisely, the local variations on the manifold tend to preserve the value of a category, and a linear interpolation between examples of different classes in general involves going through a low-density region, i.e., $p(\mathbf{x} \mid \mathbf{y} = \mathbf{i})$ for different $\mathbf{i}$ tend to be well separated and not overlap much.
8. **Temporal and spatial coherence:** Consecutive (from a sequence) or spatially nearby observations tend to be associated with the same value of relevant categorical concepts or result in a small move on the surface of the high-density manifold.
9. **Sparsity:** For any given observation $\mathbf{x}$, only a small fraction of the possible factors are relevant.
10. **Simplicity of factor dependencies:** In good high-level representations, the factors are related to each other through simple, typically linear dependencies.

Disentangled representation

1. When humans observe an object, they seek to understand the various properties of this object such as
   - shape,
   - size,
   - color

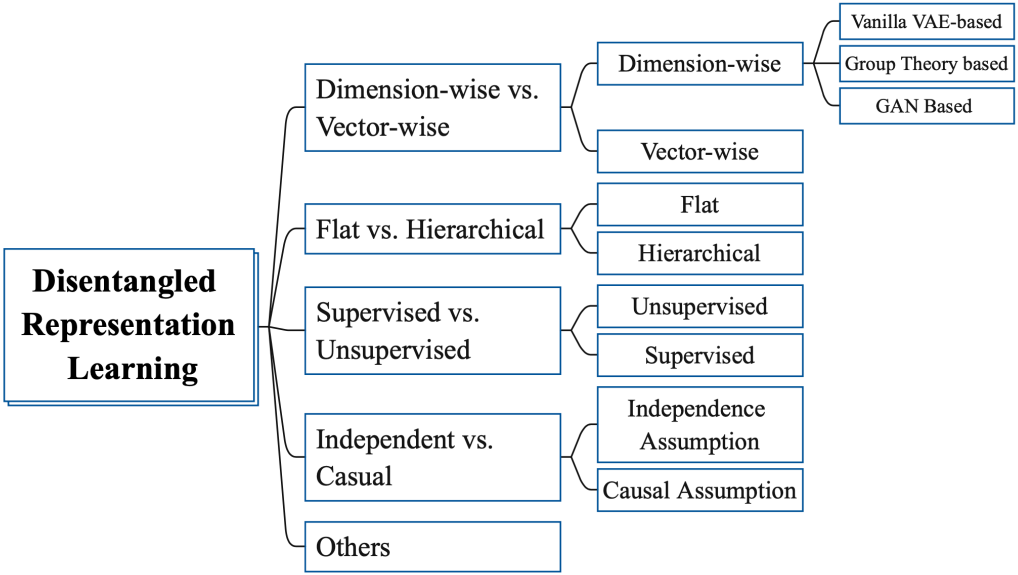
   with certain prior knowledge (Wang et al. 2024).





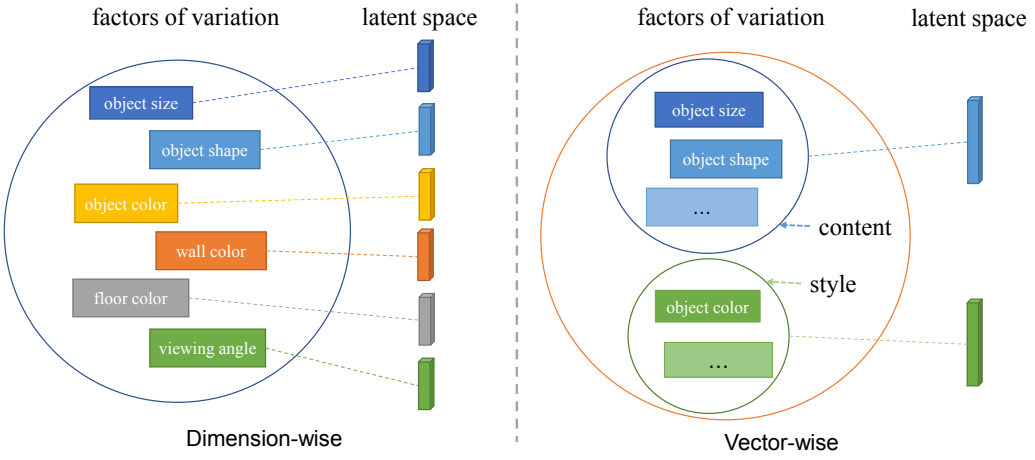
2. A disentangled representation can be defined as one where single latent units are sensitive to changes in single generative factors, while being relatively invariant to changes in other factors (Bengio, Courville, and Vincent 2013).

Categorization of disentangled representation learning approaches



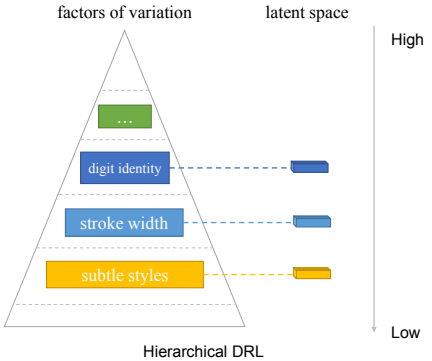
Dimension-wise vs Vector-wise DRL

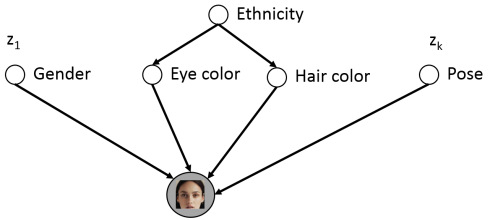
1. In dimension-wise methods, generative factors are fine-grained and a single dimension (or several dimensions) represents one generative factor.
2. In vector-wise methods, generative factors are coarse-grained and different vectors represent different types of semantic meanings.



Flat vs Hierarchical DRL

1. In flat DRL, all the factors are parallel and at the same abstraction level.
2. In hierarchical DRL, the factors of variation have different levels of semantic abstraction (hierarchical structures), either dependent or independent across levels.

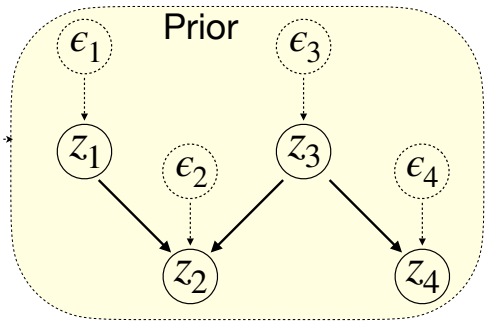




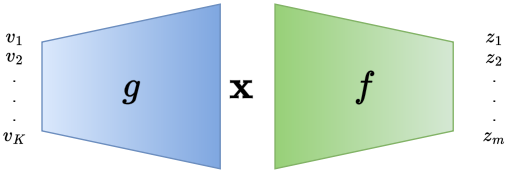
Supervised vs Unsupervised DRL

1. In unsupervised learning, the goal is automated discovery of interpretable factorized latent representations.
2. The pure unsupervised DRL is theoretically impossible without inductive bias on methods and data sets.
3. In other words, disentanglement itself does not occur naturally.
4. In supervised DRL, the learner has access to annotations (labels) of the representation for a very limited number of observations, for example through human annotation.
5. The supervised DRL setting is not universally applicable, especially when the observations are not human interpretable.
6. Hence, a completely unsupervised approach would be elegant, collecting a small number of human annotations is simple and cheap.

1. In independent DRL, that latent factors are statistically independent, so that they are supposed to be independently disentangled through independent or factorial regularization.
2. The causal DRL, underlying factors are not independent and hold certain causal relations.
3. Casual DRL methods potentially achieve more interpretable and robust representations via disentangling causal factors.



1. Given a dataset $D = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, where each data point is associated with M labeled factor of variation $\mathbf{v} = (v_1, \ldots, v_K)$.
2. Assume that there exists a mapping from $\mathbf{x}$ to m groups of latent representations $\mathbf{z} = (z_1, \ldots, z_m)$ which follows distribution $q(\mathbf{z} \mid \mathbf{x})$.



3. Disentangled representation learning can be defined as **a process of decorrelating information in the data into separate informative representation, each of which corresponds to a concept defined by humans**.
4. Important properties of disentangled representation (Do and Tran 2020).
   - Informativeness
   - Separability and independence
   - Interpretability

1. Informativeness of a particular representation (or a group of representation) z_i w.r.t the data $\mathbf{x}$ is defined as mutual information between z_i and $\mathbf{x}$:

$$I(\mathbf{x}, z_i) = \int_{\mathbf{x}} \int_{z} p_d(\mathbf{x})\, q(z_i \mid \mathbf{x}) \log \frac{q(z_i \mid \mathbf{x})}{q(z_i)} dz\ d\mathbf{x}$$

   where

$$q(z_i) = \int_{\mathbf{x}} p_d(\mathbf{x})\, q(z_i \mid \mathbf{x}) d\mathbf{x}$$

2. To represent the data faithfully, a representation z_i should be informative of $\mathbf{x}$, meaning $I(\mathbf{x}, z_i)$ should be large.

3. Since $I(\mathbf{x}, z_i) = H(z_i) - H(z_i \mid \mathbf{x})$, a large value of $I(\mathbf{x}, z_i)$ means that $H(z_i \mid \mathbf{x}) \approx 0$ given that $H(z_i)$ can be chosen to be relatively fixed.

4. In other words, if z_i is informative w.r.t to $\mathbf{x}$, then $q(z_i \mid \mathbf{x})$ usually has small variance.

5. Assume that there exists a mapping from $\mathbf{x}$ to m groups of latent representations $\mathbf{z} = (z_1, \ldots, z_m)$ which follows distribution $q(\mathbf{z} \mid \mathbf{x})$.

1. Two representation z_i and z_j are separable w.r.t the data $\mathbf{x}$ if they do not share common information about $\mathbf{x}$, that is:

$$I(\mathbf{x}, z_i, z_j) = 0$$

where $I(\mathbf{x}, z_i, z_j)$ denotes multivariate mutual information defined as:

$$I(X, Y, Z) = \sum_x \sum_y \sum_z p(x, y, z) \ln \frac{p(x, y)\, p(x, z)\, p(y, z)}{p(x, y, z)\, p(x)\, p(y)\, p(z)}$$

2. $I(\mathbf{x}, z_i, z_j)$ can be decomposed into standard bivariate mutual information terms as:

$$I(\mathbf{x}, z_i, z_j) = I(z_i, z_j) - I(z_i, z_j \mid \mathbf{x})$$

3. If $I(\mathbf{x}, z_i, z_j) > 0$, then if z_i and z_j contain redundant information about $\mathbf{x}$.
4. Achieving separability w.r.t to $\mathbf{x}$ does not guarantee that z_i and z_j are separable in general.
5. z_i and z_j are fully separable or statistically independent if and only if $I(z_i, z_j) = 0$.
6. If we have access to all representations z, we can generally say that representation z_i is fully separable from $z_{\neq i}$ if and only if $I(z_i, z_{\neq i}) = 0$.
7. **There is a trade-off between informativeness, independence, and the number of latent variables.**

Interpretability

1. Obtaining independence and informative representations does not guarantee **interpretability by humans**.
2. To achieve interpretability, we should provide model with a set of predefined concepts $\mathbf{v}$.
3. In this case, a representation z_i is interpretable w.r.t v_k if it only contains information about v_k.
4. Full interpretability can be defined as

$$I(z_i, v_k) = H(z_i) = H(v_k)$$

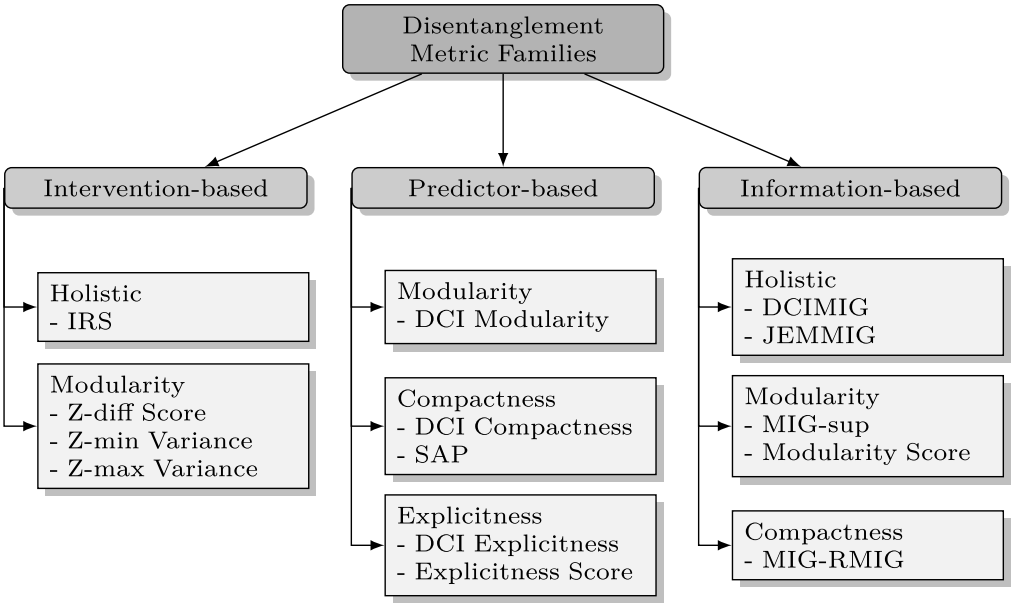
5. If we want z_i to generalize beyond the observed v_k, the model should accurately predict v_k given z_i.

Evaluating DRL methods

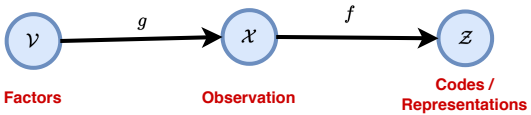
Properties disentangled representation

1. Eastwood and Williams proposed a framework for the evaluation of disentangled representations (Eastwood and Williams 2018).
2. They proposed three desirable properties of a disentangled representation: explicitness, compactness, and modularity.
   - Explicitness: The amount of information that a representation captures about the underlying factors of variation. This property is called informativeness in (ibid.).
   - Compactness: The degree to which each underlying factor is captured by a single code variable. This property is called completeness in (ibid.).
   - Modularity: The degree to which a representation factorizes or disentangles the underlying factors of variation, with each variable (or dimension) capturing at most one generative factor. This property is called disentanglement in (ibid.).
3. **Holistic methods** capture two or more properties in a single score.

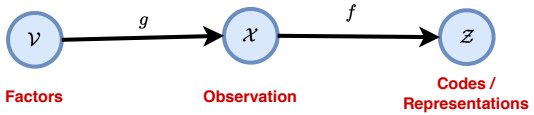
Taxonomy of supervised disentanglement metrics (Carbonneau et al. 2023).



1. The metrics in this family evaluate disentanglement by fixing factors and creating subsets of data points.

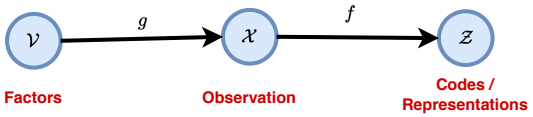


2. Codes and factors in the subsets are compared to produce a score.
3. To sample the fixed-size data subsets, these methods discretize the factor space.
4. This sampling procedure necessitates large quantities of diverse data samples to produce a meaningful score.
5. **Advantages:** These metrics do not make any assumptions on the factor–code relations.
6. **Disadvantages:** There are several hyper-parameters to adjust such as the size and the number of data subsets, the discretization granularity, classifier hyper-parameters, or the choice of a distance function.



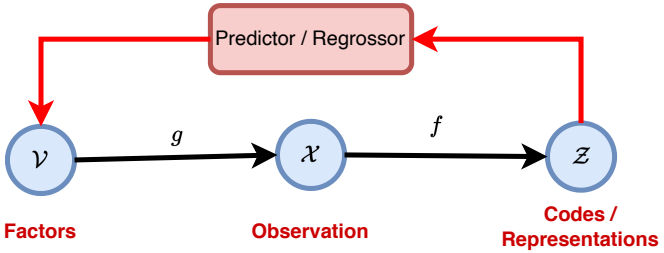
1. The Z-diff metric (β-VAE metric) selects pairs of instances to create batches. In a batch, a factor v_i is chosen randomly.
2. Then, a fixed number of pairs are formed with samples $\mathbf{v}^1$ and $\mathbf{v}^2$ that have the same value for the chosen factor ($v_i^1 = v_i^2$).
3. Pairs are represented by the absolute difference of the codes associated with the samples ($\mathbf{p} = |\mathbf{z}^1 - \mathbf{z}^2|$).
4. The intuition is that code dimensions associated with the fixed factor should have the same value, which means a smaller difference than the other code dimensions.
5. The mean of all pair differences in the subset creates a point in a final training set.
6. The process is repeated several times to constitute a sizable training set.
7. Finally, a linear classifier is trained on the dataset to predict which factor was fixed.
8. The accuracy of the classifier is the Z-diff score.

Intervention-Based Metrics (Z-Min Variance)



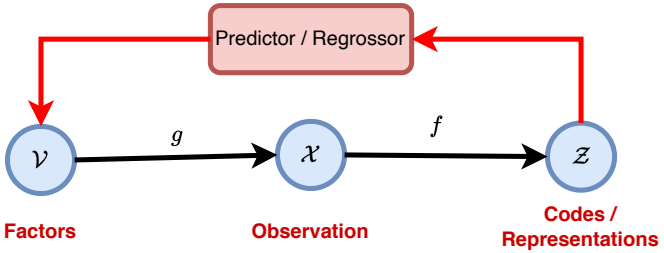
1. In Z-min variance (FactorVAE metric), code dimensions encoding a factor should be equal if the factor value is the same.
2. All codes are normalized by their standard deviation computed over the complete dataset.
3. For a subset, a factor is randomly selected and fixed at a random value. The subset contains sampled instances for which the selected factor is fixed at the selected value.
4. Variance is computed over the normalized codes in the subset. The code dimension with the lowest variance is associated with the fixed factor.
5. Several subsets are created and the factor–code associations are used as data points in a majority vote classifier.
6. The Z-min Variance score is the mean accuracy of the classifier.

Other metrics such as Z-Max Variance (R-FactorVAE) and Interventional Robustness Score were proposed.

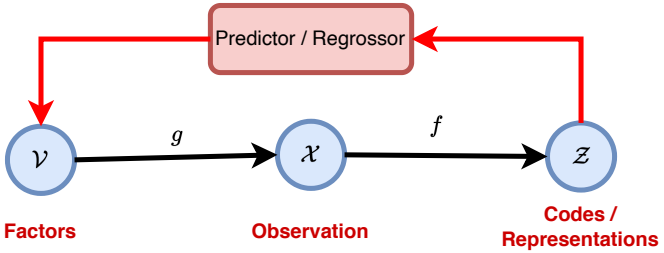


1. These metrics train regressors or classifiers to predict factor realizations from codes ($r(\mathbf{z}) \mapsto \mathbf{v}$).
2. Then, the predictor is analyzed to assess the usefulness of each code dimension in predicting the factors.
3. These methods are naturally suited to measure explicitness.
4. They are typically equipped to deal with continuous factors as well as categorical factors simply by choosing an appropriate predictor.
5. However, compared to information-based metrics, they require more design choices and hyperparameter tuning. This means that a metric is more likely to behave differently from one implementation to another.

Predictor-Based Metrics (DCI score)



1. Eastwood and Williams proposed a complete framework to evaluate disentangled representations instead of a single metric (Eastwood and Williams 2018).
2. They report separate scores for modularity, compactness, and explicitness, which they call disentanglement, completeness, and informativeness (DCI).
3. Regressors are trained to predict factors from codes. Modularity and compactness are estimated by inspecting the regressor's inner parameters to infer predictive importance weights R_{ij} for each factor and code dimension pair.
4. They use a linear lasso regressor or a random forest for nonlinear factor–code mappings.
5. For lasso regressor, the importance weights R_{ij} are the magnitudes of the weights learned by the model, while the Gini importance of code dimensions is used with random forests.



1. The compactness for factor v_i is given by

$$C_i = 1 + \sum_{j=1}^{d} p_{ij} \log_d p_{ij}$$

where p_{ij} is the probability that code dimension z_j is important to predict v_i.

2. These probabilities for all factors obtained by dividing each importance weight by the sum of all importance weights related to this factor:

$$p_{ij} = \frac{R_{ij}}{\sum_{k=1}^{d} R_{kj}}$$

3. The compactness of the whole representation is the average compactness over all factors.

1. The modularity for code dimension z_j is given by

$$D_j = 1 + \sum_{i=1}^{m} p_{ij} \log_m p_{ij}$$

   where p_{ij} is the probability that code dimension z_j is important to predict v_i.

2. These probabilities are for all factors obtained as:

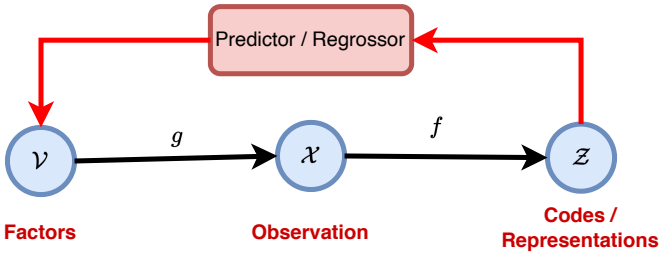
$$p_{ij} = \frac{R_{ij}}{\sum_{k=1}^{m} R_{kj}}$$

3. The modularity score for the whole representation is a weighted average of the individual code dimension modularity scores $\sum_{j=1}^{d} \rho_j D_j$.

4. The scores are weighted by ρ_j to account for codes that are less important to predict factors.

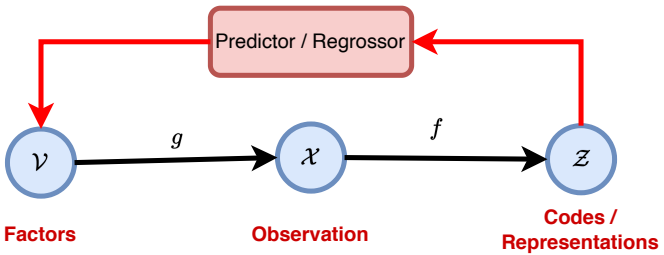
$$\rho_j = \frac{\sum_{i=1}^{m} R_{ij}}{\sum_{k=1}^{d} \sum_{i=1}^{m} R_{ik}}$$

Predictor-Based Metrics (DCI score)

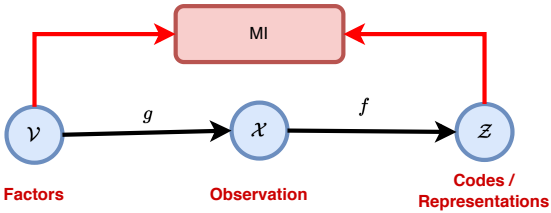


1. The prediction error of the regressor measures the explicitness of the representation.
2. With normalized inputs and outputs, it is possible to compute the estimation error for a completely random mapping and use it to normalize the score between 0 and 1.
3. A representation is not explicit if the mean squared error (MSE) of the predictor is higher than the expected MSE between two uniformally distributed random variables ($\mathbf{x}$ and $\mathbf{y}$). It can be shown that $MSE = \mathbb{E}\left[(\mathbf{x} - \mathbf{y})^2\right] = \frac{1}{6}$.
4. Thus, explicitness can be written as $1 - 6MSE$.

Predictor-Based Metrics (Explicitness score)



1. Ridgeway and Mozer use a classifier trained on the entire latent code to predict factor classes, assuming that factors have discrete values.
2. They suggest using a simple classifier such as logistic regression and report classification performance using the AUC-ROC.
3. The final score is the average AUC-ROC over all classes for all factors.
4. The AUC-ROC minimal value is 0.5, which means that the score needs to be normalized to obtain a value between 0 and 1.



1. Information-based metrics compute a disentanglement score by estimating the MI between factors and codes.

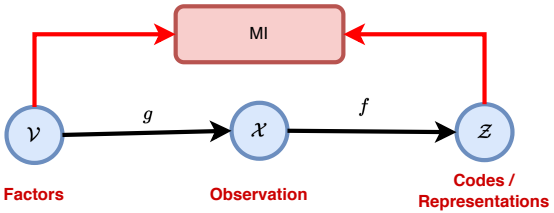
$$I(v,z) = \sum_{i=1}^{B_v} \sum_{j=1}^{B_z} p(i,j) log \frac{p(i,j)}{p(i) \times p(j)}$$

   - Factor and code spaces are discretized in B_v and B_z bins, and $p(i)$ and $p(j)$ are estimated as the proportion of samples assigned to bins i and j, respectively, over all samples.
   - Similarly, $p(i,j)$ is the proportion of samples assigned to both bins i and j.

2. These methods require fewer hyperparameters than intervention- and predictor-based metrics.

3. In addition, they do not make assumptions on the nature of the factor–code relations.

Information-Based Metrics (Mutual Information Gap)



1. Mutual Information Gap (MIG) computes the MI between each code and factor, $I(v_i, z_j)$.
2. Then, the code dimension with maximum MI is identified $I(v_i, z_*)$ for each factor.
3. Next, the second highest MI, $I(v_i, z_o)$, is subtracted from this maximal value.
4. This difference constitutes the gap, which is normalized by the entropy of the factor.

$$MIG = \frac{I(v_i, z_*) - I(v_i, z_o)}{H(v_i)}$$

5. The MIG score of all factors is averaged to report one score.

Information-Based Metrics (JEMMIG)

1. MIG verifies that the information related to a factor is expressed by only one code dimension (compactness).
2. However, modularity is not directly measured. For instance, a code dimension could contain information about more than one factor.
3. Joint entropy minus mutual information gap (JEMMIG) addresses this drawback by including the joint entropy of the factor and its best code as

$$JEMMIG = H(v_i, z_*) - I(v_i, z_*) + I(v_i, z_o)$$

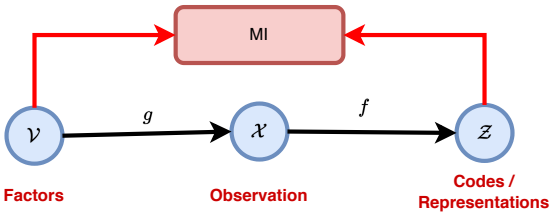
4. This metric indicates a high disentanglement quality with a lower score.
5. The maximum value is bounded by $H(v_i) + logB_z$, where B_z is the number of bins used in the code space discretization.
6. Hence, the normalized version of JEMMIG is being used

$$\widehat{JEMMIG} = 1 - \frac{H(v_i, z_*) - I(v_i, z_*) + I(v_i, z_o)}{H(v_i) + logB_z}$$

7. JEMMIG is reported as the average for all factors.

1. DCIMIG is a metric inspired by DCI and MIG. But it reports a single score for all three properties.
   - As MIG, it computes MI gaps between factors and code dimensions.
   - As DCI, it analyzes a factor–code importance matrix.
2. Then, factor with $I(v_*, z_j)$ and $I(v_o, z_j)$ is identified for each code and obtain the gap $R_j = I(v_*, z_j) - I(v_o, z_j)$.
3. Each of these gaps R_j relates to a code dimension and the factor for which MI is maximal.
4. For each factor v_i, finds all associated gaps R_j and use them as score S_i for this factor.
5. If there are more than one R_j associated with the factor, S_i equals the highest v_i. If there are none, $S_i = 0$.
6. Finally, the metric is the sum of all scores normalized by the total factor entropy

$$DCIMIG = \frac{\sum_{i=1}^{m} S_i}{\sum_{i=1}^{m} H(v_i)}$$



1. The factor v_*, which shares the maximum MI for each code dimension z_j, is identified.
2. This maximal MI value ($I(v_*, z_j)$) is then compared with MI values of all other factors

$$modularity_j = 1 - \frac{\sum_{i \in \mathcal{V}_{\neq *}} I(v_i, z_j)^2}{(m-1) I(v_*, z_j)^2}$$

   where $\mathcal{V}_{\neq *}$ as the set of all factors except v_* and m as the number of factors.
3. The average modularity score over all codes is reported.

Conclusions

Metric	Modularity	Compactness	Explicitness	Calibrated	Robust to Noise	Robust to Non-measured Factors	Nonlinear Relation	Discretization-free	Few Hyper-parameters	Data Efficient
Z-diff [1]	✗	✗	✗	✓	✓	✓	✗	✗	✗	✓
Z-min Variance [2]	✗	✗	✗	✓	✓	✓	✗	✗	✗	✓
Z-max Variance [3]	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
IRS [4]	✓	✗	✓	✗	n/a	✗	✗	✗	✗	✓
DCI - Lasso [5]	✓	✗	✓	✗	✓	✓	✗	✓	✓	✓
DCI - Random Forest [5]	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓
Explicitness Score [6]	✗	✗	✓	✗	n/a	✓	✗	✗	✓	✗
SAP [7]	✗	✓	✓	✓	n/a	✓	✗	✓	✓	✓
MIG-RMIG [8], [9]	✗	✓	✗	✓	✗	✓	✗	✗	✓	✓
MIG-sup [10]	✓	✗	✗	✓	✗	✗	✗	✗	✓	✓
JEMMIG [9]	✓	✓	✓	✓	✗	✓	✗	✗	✓	✓
Modularity Score [6]	✗	✗	✗	✗	✓	✗	✗	✗	✓	✓
DCIMIG [11]	✓	✗	✓	✓	✗	✓	✗	✗	✓	✓

Conclusions

1. Experimental results show different limitations for each metric.
2. Discretization hinders reliability under limited amount of data, noise, and nonlinear factor–code relations.
3. Predictor-based metrics, when parameterized carefully, are the best performing family of solutions.
4. It is better that each disentanglement property should be measured separately for better interpretability.

References

References

- Bengio, Yoshua, Aaron C. Courville, and Pascal Vincent (2013). "Representation Learning: A Review and New Perspectives". In: *IEEETransactions on Pattern Analysis and Machine Intelligence* 35.8, pp. 1798–1828.
- Carbonneau, Marc-Andre et al. (2023). "Measuring Disentanglement: A Review of Metrics". In: *IEEE Transactions on Neural Networks and Learning Systems*.
- Do, Kien and Truyen Tran (2020). "Theory and Evaluation Metrics for Learning Disentangled Representations". In: *International Conference on Learning Representations*.
- Eastwood, C. and C. K. I. Williams (2018). "A framework for the quantitative evaluation of disentangled representations". In: *International Conference on Learning Representations*.
- Wang, Xin et al. (2024). "Disentangled Representation Learning". In: *IEEETransactions on Pattern Analysis and Machine Intelligence* 46.12, pp. 9677–9696.

Questions?