1 Introduction

Clustering is denoising is compression is dimension reduction is classification is learning[citation needed].

Compared to unsupervised learning, classifiers for labeled data is a mostly solved problem[citation needed]. This makes converting an unsupervised learning problem to a classification problem a common machine learning approach. RLHF can be seen as one such example. Clustering/self-supervision is another, where labels are inferred from the data itself.

Self-supervised classification generally requires substantial ad hoc hyperparameter tuning. The simplest example is k-means clustering, where the number of clusters is directly specified. Other approaches can find an optimal number of clusters by first constructing intermediate graph representation of the data. This graph can be partitioned into clusters that maximize modularity[3], which is defined as

$$\mathcal{H} = \frac{1}{2m}\sum_{c}(\, e_c - \gamma\frac{K_c^2}{2m}) \tag{1}$$

where m is the average degree of the graph, e_c is the number of edges in cluster c, and K_c is the number of nodes in cluster c. This equation has a nicely intuitive interpretation. Modularity $\mathcal{H}$ of a graph is given by the sum of how well-connected its clusters are, defined as the difference between the number of edges in the cluster and the expected number of edges given the number of nodes in the graph and average degree of a node.

Although optimizing modularity is NP-hard, it can be approximated using the leiden algorithm[5].

Though modularity ensures that clusters are well-connected, the number of clusters returned is dependent on γ, which cannot be inferred from the data. A value of k must also be selected for the input graph.

Clustering can be thought of as a form of denoising. Clusters represent the underlying "signal" of a data set. Variation within clusters is "noise".

We define a *typology* as an injective mapping of a random vector to a set of *informative central cases*.We define an *ontology* as a function that encodes a random vector $\mathbf{x}$ as a composition of typologies applied to $\mathbf{x}$.

Let Ω be an ontology of data X consisting of n typologies. Let Θ be a typology in Ω and $\hat{\mathbf{x}}$ be an encoding of a vector $\mathbf{x} \in X$. Let $\Theta_0(\mathbf{x}) = \mathbf{0}$.

$$\Omega(\mathbf{x}) = \bigoplus_{i=1}^{n} \Theta_i(\mathbf{x} - \Theta_{i-1}(\mathbf{x})) \tag{2}$$

We propose that by minimizing the self-supervised loss function

$$\mathcal{L}(\Omega) = \mathbb{E}||\mathrm{sum}[\Omega(\mathbf{x})] - \mathbf{x}||^2 \tag{3}$$

we can obtain a maximally informative ontology.

Noise2self[1] applies self-supervision to denoising. DEWAKSS[4] uses noise2self selects optimal hyperparameters for generating a k-NN represeantation of single cell RNA-seq by using an arbitrary graph as a weighted affinity kernel

We define a typology Θ as a partition $\mathcal{J}$ of data X such that each subset $J \in \mathcal{J}$ is associated with a central member $\hat{x}_J \in \hat{X}_{\mathcal{J}}$. $\hat{X}_{\mathcal{J}}$ is given by applying an estimator $\bigoplus_{J\in\mathcal{J}} \mathbb{E}[J]$

2 Methods

2.1 Dimension Reduction

Many parameters are strongly correlated (Fig. 2b). This is undesirable because each parameter additively contributes to distance used for clustering, resulting in disproportionate weight being given to phenotypes captured by multiple parameters. Linear methods of dimenison reduction (e.g. PCA) assume that all variables are independent and can be linearly combined. We could not assume that all of our measured input parameters were independent, so we instead used an autoencoder for dimension reduction.

An autoencoder is a neural network architecture widely used for denoising and image recognition. It works by encoding the input data into a lower dimensional representation that can be decoded with minimal loss. By extracting this lower dimensional encoding (the "bottleneck" or "embedding" layer), an autoencoder can be used for dimension reduction[6]. This results in an embedding that corresponds to the information content of the input data rather than absolute distance in phenotype space.

2.1.1 Pruning

We trained four autoencoders using embedding layers of 2, 3, 7, and 14 dimensions. We selected the 2-dimensional embedding based on Akaike Information Criterion[2] (Table 1; Fig. 1a), defined as

$$AIC = 2k - 2ln(\hat{L}) \tag{4}$$

where k is the number of parameters and $\hat{L}$ is a likelihood function, which we define as $1 - MSE$.

2.2 Clustering Algorithm

Euclidean distance between embeddings is used to compute a k-nearest neighbors graph. The graph is then partitioned into clusters by modularity[3], which is defined as

$$\mathcal{H} = \frac{1}{2m} \sum_{c} (e_c - \gamma \frac{K_c^2}{2m}) \tag{5}$$

where m is the average degree of the graph, e_c is the number of edges in cluster c, and K_c is the number of nodes in cluster c. This equation has a nicely intuitive interpretation. Modularity $\mathcal{H}$ of a graph is given by the sum of how well-connected its clusters are, defined as the difference between the number of edges in the cluster and the expected number of edges given the number of nodes in the graph and average degree of a node.

Because optimizing modularity is NP-hard, we used the leiden algorithm to approximate an optimal solution[5].

Though modularity ensures that clusters are well-connected, the number of clusters returned is dependent on γ, which cannot be inferred from the data. A value of k must also be selected for the input graph.

2.3 Hyperparameter Selection

We performed clustering for 100 random γ values between 0.01 and 3.0 for k values ranging from 3 to 53.

We selected k and γ based on four validation metrics. Mean silhouette width was calculated from the euclidean distance etween embryos.

2.3.1 k Selection

References

[1] Joshua Batson and Loc Royer. "Noise2Self: Blind Denoising by Self-Supervision". In: *CoRR* abs/1901.11365 (2019). arXiv: 1901.11365. URL: http://arxiv.org/abs/1901.11365.

[2] Joseph E Cavanaugh and Andrew A Neath. "The Akaike information criterion: Background, derivation, properties, application, interpretation, and refinements". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 11.3 (2019), e1460.

[3] Jörg Reichardt and Stefan Bornholdt. "Statistical mechanics of community detection". In: *Phys. Rev. E* 74 (1 July 2006), p. 016110. DOI: 10.1103/PhysRevE.74.016110. URL: https://link.aps.org/doi/10.1103/PhysRevE.74.016110.

[4] Andreas Tjärnberg et al. "Optimal tuning of weighted kNN- and diffusion-based methods for denoising single cell genomics data". In: *PLOS Computational Biology* 17.1 (Jan. 2021), pp. 1–22. DOI: 10.1371/journal.pcbi.1008569. URL: https://doi.org/10.1371/journal.pcbi.1008569.

[5] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. "From Louvain to Leiden: guaranteeing well-connected communities". In: *Scientific reports* 9.1 (2019), pp. 1–12.

[6] Yasi Wang, Hongxun Yao, and Sicheng Zhao. "Auto-encoder based dimensionality reduction". In: *Neurocomputing* 184 (2016). RoLoD: Robust Local Descriptors for Computer Vision 2014, pp. 232–242. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2015.08.104. URL: https://www.sciencedirect.com/science/article/pii/S0925231215017671.

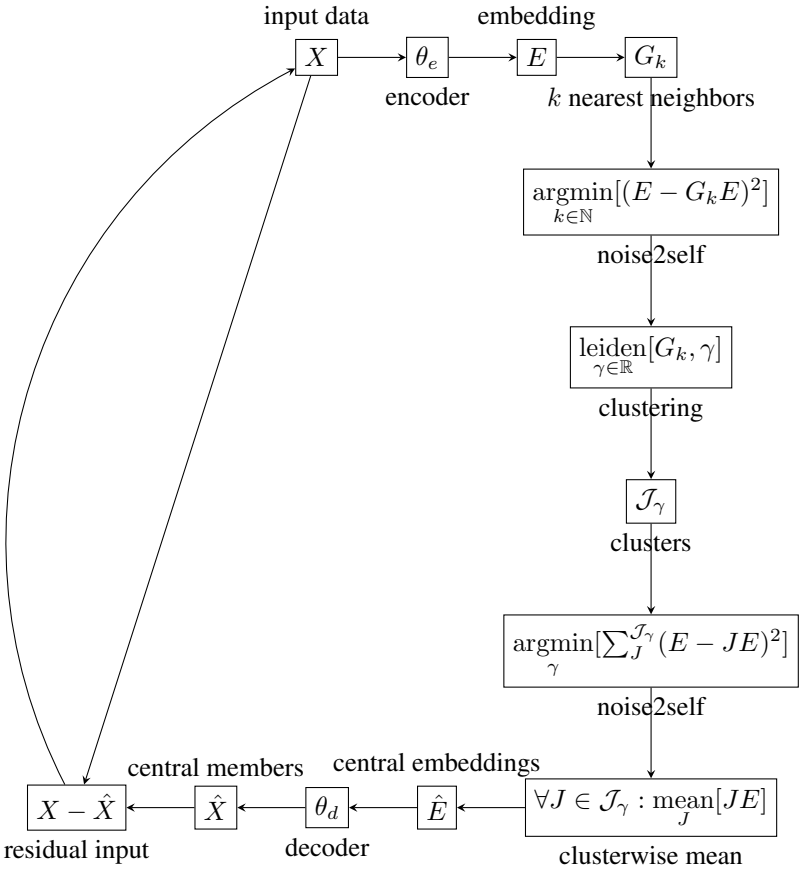

Figure 1: Typology block.