ON MINIMAL VARIATIONS FOR UNSUPERVISED REPRESENTATION LEARNING

Vivien Cabannes *Alberto Bietti* *Randall Balestriero*

Meta AI

ABSTRACT

Unsupervised representation learning aims at describing raw data efficiently to solve various downstream tasks. It has been approached with many techniques, such as manifold learning, diffusion maps, or more recently self-supervised learning. Those techniques are arguably all based on the underlying assumption that target functions, associated with future downstream tasks, have low variations in densely populated regions of the input space. Unveiling minimal variations as a guiding principle behind unsupervised representation learning paves the way to better practical guidelines for self-supervised learning algorithms.

***Index Terms*—** Self-supervised learning, unsupervised learning, minimal variations, first principles.

1. INTRODUCTION

Data is everywhere, but it is often too unstructured or high dimensional to leverage classical statistics on their raw form. Recent advances in machine learning have succeeded in exploiting parts of the millions terabytes of unlabeled data contained on the internet. This was achieved by creating self-supervised tasks to be solved by the machine, inciting it to learn good representations of text [1, 2]. Those "foundational" representations are now being leveraged to solve several "downstream" tasks on languages [3]. Similar developments have been made on other high-dimensional data such as images, videos or audio speeches [4, 5, 6]. Despite their rapid progress, the training of self-supervised learning (SSL) models remains challenging and lacks theoretical foundations.

Learning without supervision has been historically referred to as unsupervised learning. While at first sights, the literature bodies on unsupervised learning and self-supervised learning seem relatively disjoint, connections have been made between the two [7, 8]. This work provides further insights on their links through the concept of minimal variations, detailed in Section 3. Theory is verified on synthetic experiments in Section 4. This understanding could be leveraged in future work to improve SSL algorithms in practical settings with limited compute resources.

2. SETTING AND CONTEXT

In the following, $\mathcal{X}$ shall be a Hilbert space (*i.e.* endowed with a scalar product) and $\mathcal{Y}$ an output space. A distribution ρ_X is assumed to have generated a dataset $(X_i)_{i\in[n]}$ of independent samples $X_i \sim \rho_X$ for $i \in [n]$.[1] Our goal is to find a representation $\varphi : \mathcal{X} \to \mathbb{R}^p$ for a small p, such that for relevant downstream distributions ρ on pairs of input/output and loss functions $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, one can efficiently minimize the subsequent population risk

$$\mathcal{R}(f; p, \ell) = \mathbb{E}_{(X,Y)\sim\rho}\left[\ell(f(X), Y)\right], \tag{1}$$

based on i.i.d. samples (X_i, Y_i). More exactly, the optimal functions f should easily be approached under the form $f = g \circ \varphi$ for g in a small class of functions. Typically, g would be a linear function, a.k.a. a linear probe. Reducing the search of $f : \mathcal{X} \to \mathcal{Y}$ in a potentially big function space to the search of $g : \mathbb{R}^p \to \mathcal{Y}$ in a much smaller one, will drastically improve sample efficiency [9].

To learn the representation φ, SSL leverages augmentations of data. It defines $t : \mathcal{X} \times \Xi \to \mathcal{X}$ a transformation parameterized by $\xi \in \Xi$. For example, Ξ could be $\mathcal{X}$ and $t(X, \xi) = X + \xi$. Assuming that transformations $t(X, \xi)$ do not fundamentally change the semantic of the input, any pairs of augmented and original points should be close in the features space $\varphi(\mathcal{X})$. This is put in equations through the minimization of the variational quantity

$$\mathbb{E}_{X,\xi}\left[d(\varphi(t(X, \xi)), \varphi(X))\right], \tag{2}$$

for d a notion of similarity in $\mathbb{R}^p$, *e.g.* the square loss $d(x, x') = \|x - x'\|^2$. In practice, φ is often taken as a neural network, and its learning is conducted through the optimization of its parameters. Equation (2) is trivially minimized by setting φ to a constant. To avoid such a "collapse" phenomenon, one should encourage diversity in the representation, for instance by using the constraint

$$\mathbb{E}_X[\varphi(X)\varphi(X)^\top] = I. \tag{3}$$

Classical self-supervised techniques such as SimCLR [4], Barlow Twins [10] and VICReg [11] can be understood as implementing different specifications of such a scheme [8] (respectively, d would be the cosine similarity, some cross-correlation measure, and the square hinge loss).

[1] The set $\{1, \cdots, n\}$ is denoted $[n]$.

3. MINIMAL VARIATIONS

3.1. Classical hypothesis

This section reviews classical assumptions about the nature of downstream tasks with respect to the input distribution ρ_X. It shows how those assumptions are related to the idea that future target functions have low variations on highly populated regions of the input space.

Often praised in semi-supervised learning setting, the *cluster assumption* states that the support of ρ_X have several connected components and that downstream classification tasks are likely to respect this structure, *i.e.* labels shall be constant over each connected component a.k.a. cluster [12]. In other terms, one expects the decision boundary[2] between classes to be situated in regions of the input space where there is no density. Yet, on big or poorly curated datasets, classes might not be separated by no-density regions. In such a setting, the cluster assumption is relaxed as the *low-density separation hypothesis*, assuming that downstream decision boundaries will fall in low-density regions (*i.e.* where ρ_X is small). For example, in a balanced binary classification problem where $\mathcal{Y} = \{-1, 1\}$ and $\mathrm{d}\rho(x \mid y) \propto \exp(-\|x + y\mu\| / \sigma^2)\,\mathrm{d}x$, the optimal decision boundary is the hyperplane $\mu^\perp$ which does minimize the value of $\rho_X(A)$ for any hyperplane A that cross $[-\mu, +\mu]$.

In regression settings, it is often assumed that the downstream target functions will be smooth on densely populated regions of the input space [13]. Variations of functions are measured through quantities such as

$$\mathcal{J}(f) = \mathbb{E}\left[\|\nabla f(X)\|^q\right], \tag{4}$$

for $q = 1$ (total variation), $q = 2$ (Dirichlet energy) or higher (q-Laplacian). Interestingly, binary classification problems are often approached through the learning of a continuous surrogate function f whose sign is taken as the classification rule [14]. From a classification perspective, variations of f are only needed in order for f to change sign, and the low-variation hypothesis states that those variations should take place in sparsely populated areas of the input space. This is coherent with the low-density separation hypothesis stating that f should change sign in sparsely populated regions.

3.2. Embedding for minimal variations

This section extends on classical unsupervised techniques as aiming to minimize the criterion (4) under the orthonormal constraint (3).

Assuming low-variation of downstream tasks, it is natural to design φ in order to represent a maximum number of low-variation functions as $g \circ \varphi$. Considering linear probes, the span of $(\varphi_i)_{i\in[p]}$ could be searched as a p-dimension space of functions with minimal variations according to the criterion (4). Put in equations with $\varphi : \mathcal{X} \to \mathbb{R}^p$, this reads

$$\operatorname*{arg\,min}_{\varphi;\text{s.t. (3)}} \max_{w\in\mathbb{R}^p;\|w\|=1} \mathcal{J}(w^\top \varphi(\cdot)),$$

under the coverage constraint (3). Such a φ is an ideal data representation to solve downstream tasks with linear probes, as long as solutions verify the low-variation hypothesis. With D denoting the Jacobian, the formulation with $q = 2$ translates as

$$\varphi = \operatorname*{arg\,min}_{\varphi \text{ s.t. (3)}} \mathbb{E}\left[\|D\varphi(X)\|_F^2\right]. \tag{5}$$

In practice, this formulation is favored for analytical reasons. By making $\mathcal{J}(f)$ a quadratic form, it reveals the operator $\mathcal{L}$ that represents it.[3] In particular, equation (5) is solved explicitly with φ_i the i-th eigenfunctions of $\mathcal{L}$.[4] The proof is a simple application of the Rayleigh-Ritz formula, that defines eigenfunctions recursively through the formula

$$\begin{aligned}\varphi_i = \operatorname*{arg\,min}_{\varphi:\mathcal{X}\to\mathbb{R}} \langle\varphi, \mathcal{L}\varphi\rangle = \mathbb{E}\left[\|\nabla\varphi(X)\|^2\right]\\ \text{s.t. } \mathbb{E}\left[\varphi_i(X)\varphi_j(X)\right] = \delta_{ij} \quad \forall\, j < i.\end{aligned}$$

In the literature, this approach is often referred to as spectral embedding (the space is embedded through the spectral decomposition of the operator). It is particularly well suited for the cluster assumption, since the null space of $\mathcal{L}$ is nothing but the span of the indicator functions of each connected component of ρ_X, which has motivated its use for clustering and manifold regularization [15]. Under mild assumptions, $\mathcal{L}$ is indeed a diffusion operator (when ρ_X has a density and compact support $\mathcal{L}f = -\Delta f + \langle\nabla \log \rho_X, \nabla f\rangle$), which links it to diffusion maps [16], label propagation [17] and Langevin dynamics [18].

Since the 2000s, the criterion (4) has been approached in a non-parametric fashion based on finite differences, leveraging graph Laplacians [17, 19]. Based on samples $(X_i)_{i\in[n]}$, it aims at minimizing

$$\mathcal{E}_g(\varphi) = \sum_{i,j\in[n]} \left[k_\sigma(X_i, X_j)\left\|\varphi(X_i) - \varphi(X_j)\right\|^2\right], \tag{6}$$

with k a notion of similarity to perform finite differences and σ a scaling parameter (*e.g.* $k_\sigma(x, x') = \exp(-\|x - x'\|^2 / \sigma^2)$), and subject to the empirical version of the constraint (3),

$$\frac{1}{n^2}\sum_{i,j\in[n]} \varphi(X_i)\varphi(X_i)^\top = I.$$

[2] For a predictor $f : \mathcal{X} \to \mathcal{Y}$, the input space $\mathcal{X}$ is partitioned into decision regions $\mathcal{X}_y = \{x \mid f(x) = y\}$ indexed by $y \in \mathcal{Y}$, "decision boundaries" refers to the boundaries of those regions.

[3] Some minor mathematical precautions should be taken to deal with this weighted Sobolev pseudo-norm, we will omit them in this paper.

[4] The solution of φ is unique up to orthonormal transformations $U\varphi$ for $U \in \mathbb{R}^{p\times p}$ orthogonal, and to permutation of eigenfunctions associated with the p-th eigenvalue of $\mathcal{L}$.

3.3. Consistency results

This section discusses limiting behaviors of the methods described previously, namely SSL (2), graph Laplacian (6) and Dirichlet energy (5).

Graph Laplacians have arguably two convergence properties. On the one hand, keeping the scale σ constant, as the number of samples n goes to infinity, the empirical minimizer minimizes the following measure of variations

$$\mathbb{E}\left[\|d(X)f(X) - k_\sigma * f(X)\|^2\right], \text{ with}$$
$$k_\sigma * f(x) = \mathbb{E}[k_\sigma(x, X)f(X)], \quad d = k_\sigma * 1,$$

which can be seen as a smoothed, reweighted version of the Dirichlet energy. The convergence happens relatively fast, typically in $O(n^{-1/2})$ in L^2-norm [20]. On the other hand, with the right scaling of σ, this finite difference method is able to converge towards the ideal solution defined by (5). Yet the convergence rates are much worse, *e.g.* in $O(n^{-1/d})$ for d the dimension of the data manifold ($d = \dim \operatorname{supp} \rho_X$) [21]. This may be understood intuitively, to measure variations with finite differences, the number of points needed grows exponentially with dimension [22].

Alternatively, (4) might be estimated directly with empirical samples and a parametric model such as neural networks or kernel methods. This enables fast convergence towards the solution of (5) within the search space of functions for φ. By not suffering from the curse of dimension and converging to the ideal operator, this approach is statistically superior [23]. Yet, it requires optimization over derivatives which can lead to computational drawbacks.

3.4. Insights for SSL

We argue that SSL objectives such as (2) can be seen as measures of variations. In particular, since $t(x, \xi)$ is supposed to be closed to x (at least semantically speaking), it behaves as a random variable (with respect to ξ) to compute finite differences at a point $x \in \mathcal{X}$. Therefore, we expect SSL algorithms either, when keeping the scale of ξ constant,[5] to converge fast to the minimizer of some smoothed version of a functional that measure variations (4) (depending on t and the distance d), or, with the right decreasing scale, to converge slowly to the ideal functional itself.

4. EXPERIMENTS

Prior sections have introduced three techniques to learn φ, SSL (2), graph Laplacian (6) and empirical Dirichlet energy (5). We have argued that they all aimed at learning the same type of functions, *i.e.* orthogonal functions that minimize variations. After implementation details, proof-of-concept experiments verify this claim.

[5] Here, Ξ is implicitly assumed to be a Banach space and the transformation to verify $t(x + \xi) = x + o(\|\xi\|)$.

4.1. Implementation details

This section reviews implementation details based on empirical samples $(X_i)_{i\in[n]}$. Experiments were made with the following specification of the self-supervised learning objective

$$\mathcal{E}_s(\varphi) = \frac{1}{n}\sum_{i\in[n]} \|\varphi(X_i) - \varphi(X_i + \sigma\xi_i)\|^2, \tag{2}$$

with ξ_i a random unit Gaussian variable, and σ a scale parameter. The empirical version of Dirichlet energy reads

$$\mathcal{E}_e(\varphi) = \frac{1}{n}\sum_{i\in[n]} \|\nabla\varphi(X_i)\|^2, \tag{5}$$

which, in the case of deep networks, is related to double backpropagation [24] and has been used in other contexts [25, 26]. Finally, the graph Laplacian objective is nothing but (6).

In experiments, the orthogonality constraints was relaxed as a penalty reading

$$\Omega(\varphi) = \Big\|\frac{1}{n^2}\sum_{i\in[n]} \varphi(X_i)\varphi(X_i)^\top - I\Big\|^2,$$

while the final objective is $\mathcal{E}(\varphi) + \lambda\Omega(\varphi)$.

Self-supervised learning is known to be quite unstable to changes in hyperparameters. In experiments, the scale parameters (standard deviation of augmentation in SSL, and kernel scaling in graph Laplacian) were set to match the width of the half-moon dataset (which was itself generated with Gaussian noise). Stochastic gradient descent parameters (learning rate scheduling, batch size) were tuned to succeed the sole minimization of Ω.[6] Finally, the regularizer λ was set to approximately balance the penalty and the objective at hand (the learning rate was divided by λ accordingly). The representation φ was parameterized with a fully connected neural network with five hidden layers, each containing a hundred neurons. The code is available online at `https://github.com/VivienCabannes/laplacian`.

4.2. Consistency results.

This section checks the claim that the three objectives (2), (5) and (5) are learning similar functions. It proceeds with the two half-moons dataset (Figure 1).

In this setting, the eigenfunctions of $\mathcal{L}$ are related to the Fourier basis on the union of two segments and are relatively stable under smoothing of the differential functional. Beside the constant function, the null space of $\mathcal{L}$ is made of φ_1 the difference of the indicator functions of both half-moons. The second two eigenfunctions φ_2 and φ_3 are first-mode waves on each component. Figure 1 reports the learned φ for the three

[6] Note that because the expectation is inside the norm in Ω a naive mini-batch strategy does not provide unbiased stochastic estimate of its gradient. We overcame this issue by considering large batches.

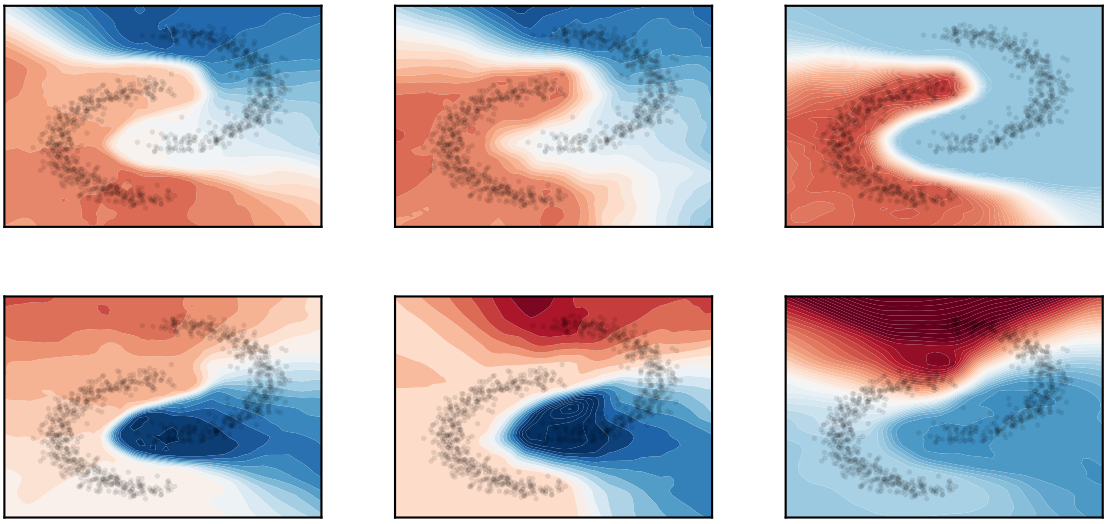

Fig. 1: Functions learned with $p = 2$ for SSL (2) on the right, graph Laplacian (6) in the middle, and Dirichlet energy (5) on the right. Datapoints $(X_i)_{i\in[n]}$ are represented as black dots, while the functions $\varphi_i(X)$ are represented through the level regions in different colors.

methods with $p = 2$. All methods recover φ_1 (top), the first two recover φ_2 while the third one recovers a mixture $\cos(\theta)\varphi_2 + \sin(\theta)\varphi_3$ for some $\theta \in [0, 2\pi]$, which also minimizes (5). Table 1 is concerned with $p = 5$, and the downstream task that consists in predicting if x was in the top (or bottom) of the left (or right) half-moon. This generates a classification problem with four different classes. Such a task is ideal to evaluate our argumentation since the first five eigenfunctions of the diffusion operator $\mathcal{L}$ discriminate those four parts of the space with linear probing. The results are satisfying.

SSL (2)	energy (5)	graph (6)
96.14 ± 0.16	$\mathbf{97.32} \pm 0.16$	95.15 ± 0.19

Table 1: Accuracy on downstream task with linear probing to check eigenspace retrieval (random is 0.25).

4.3. Discussion

While the previous experiments are made on small synthetic data, some behaviors are worse mentioning. First, the SSL objective (2) and the graph Laplacian (6) lead to similar results. Yet, graph Laplacian only uses samples on the data manifold, while augmented data in SSL gets out of it. At first sight, it seems better to restrict computations of finite differences to the manifold: the method would scale with the intrinsic dimension of data instead of the explicit input dimension [21]. In practice, on the contrary, people do use aggressive color jittering leading to unnatural augmented images. We notice in experiments that this prevents neural networks from taking arbitrary values outside the support of the data. On the other hand, graph Laplacian can exhibit high values outside the manifold, making it vulnerable to distribution shift or adversarial attacks [27].[7]

In high dimensional input space, the Dirichlet energy method (5) is supposed to exhibit much better statistical properties [23]. In practice, however, it suffers from some computational drawbacks. More specifically, for neural networks with two hidden layers with both one hundred neurons, graph Laplacian and SSL find similar solutions as the one in Figure 1 while the Dirichlet energy method tends to collapse to basic orthogonal functions such as $\varphi_i = \cos(2\pi\omega_i \langle e_i, x\rangle)$ for some small ω and some unit vector e_i. This behavior vanishes with deeper networks.

Finally, when p gets big, the different functions φ_i learned are hard to parse visually. While a solution for φ are waves with increasing modes, in practice the networks learn an orthogonal transformation of it, *i.e.* $\varphi \leftarrow U\varphi$ for $U \in \mathbb{R}^{p\times p}$ a random orthogonal matrix. If those different modes were to correspond to features in the original data, it would be natural to ask for (φ_i) to describe those local regions of the input space associated with features. This suggests room for future improvements of SSL methods.

5. CONCLUSION

This paper unveiled the link between novel self-supervised learning techniques and classical unsupervised learning ones. Key to all those methods is the low-variation hypothesis. In future work, we hope to leverage this understanding to provide practical guidelines to design self-supervised learning algorithms and deploy them in the wild without having to rely on expensive hyperparameters validation.

[7] Additionally, rote that SSL gets fresh samples for each ξ_i at each optimization epoch, which reduces in-samples bias.

6. REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[2] Aakanksha Chowdhery et al., "PaLM: Scaling language modeling with Pathways," 2022.

[3] Tom Brown et al., "Language models are few-shot learners," 2020.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020.

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin, "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments," in *NeuRIPS*, 2020.

[6] Alec Radford, "Robust speech recognition via large-scale weak supervision," 2022.

[7] Jeff Z. HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma, "Provable guarantees for self-supervised deep learning with spectral contrastive loss," in *NeurIPS*, 2021.

[8] Randall Balestriero and Yann LeCun, "Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods," in *NeurIPS*, 2022.

[9] Vladimir Vapnik, *The Nature of Statistical Learning*, Springer, 1995.

[10] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny, "Barlow Twins: Self-supervised learning via redundancy reduction," in *ICML*, 2021.

[11] Adrien Bardes, Jean Ponce, and Yann Lecun, "VICReg: Variance-invariance-covariance regularization for self-supervised learning," in *ICLR*, 2022.

[12] Philippe Rigollet, "Generalization error bounds in semi-supervised classification under the cluster assumption," *JMLR*, 2007.

[13] Jesper van Engelen and Holger Hoos, "A survey of semi-supervised learning," *Machine Learning*, 2020.

[14] Peter Bartlett, Michael Jordan, and Jon Mcauliffe, "Convexity, classification, and risk bounds," *JASA*, 2006.

[15] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *JMLR*, 2006.

[16] Ronald Coifman and Stéphane Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, 2006.

[17] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *ICML*, 2003.

[18] Dominique Bakry, Ivan Gentil, Michel Ledoux, et al., *Analysis and geometry of Markov diffusion operators*, vol. 103, Springer, 2014.

[19] Mikhail Belkin and Partha Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comp.*, 2003.

[20] Ulrike von Luxburg, Mikhail Belkin, and Olivier Bousquet, "Consistency of spectral clustering," *Annals of Stat.*, 2008.

[21] Matthias Hein, Jean-Yves Audibert, and Ulrike von Luxburg, "Graph Laplacians and their convergence on random neighborhood graphs," *JMLR*, 2007.

[22] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux, "Label propagation and quadratic criterion," *Semi-Supervised Learning*, 2006.

[23] Vivien Cabannes, Loucas Pillaud-Vivien, Francis Bach, and Alessandro Rudi, "Overcoming the curse of dimensionality with Laplacian regularization in semi-supervised learning," in *NeurIPS*, 2021.

[24] Harris Drucker and Yann Le Cun, "Improving generalization performance using double backpropagation," *IEEE transactions on neural networks*, 1992.

[25] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville, "Improved training of Wasserstein GANs," in *NeuRIPS*, 2017.

[26] Alberto Bietti, Grégoire Mialon, Dexiong Chen, and Julien Mairal, "A kernel perspective for regularizing deep neural networks," in *ICML*, 2019.

[27] Judy Hoffman, Daniel A. Roberts, and Sho Yaida, "Robust learning with Jacobian regularization," 2019.