SIMPLICIAL REGULARIZATION

Jose Gallego-Posada[*]
Mila, Université de Montréal
gallegoj@mila.quebec

Patrick Forré
University of Amsterdam
p.d.forre@uva.nl

ABSTRACT

Inspired by the fuzzy topological representation of a dataset employed in UMAP (McInnes et al., 2018), we propose a regularization principle for supervised learning based on the preservation of the simplicial complex structure of the data. We analyze the behavior of our proposal in contrast with the *mixup* (Zhang et al., 2018) framework on dimensionality reduction and classification tasks. Our experiments show how simplicial regularization induces more appropriate learning biases and alleviates some of the shortcomings of state-of-the art methods for regularization based on data augmentation.

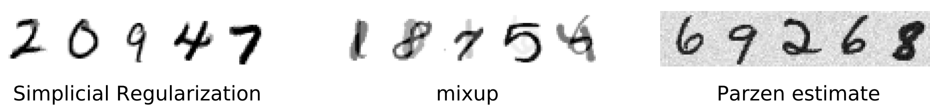

Figure 1: Comparison of artificial samples generated from several vicinal distributions on MNIST.

The richness of the class of neural networks as function approximators is well documented (Hornik, 1991; Cybenko, 1989; Telgarsky, 2015) and comes at the expense of having to regularize such functional families. Recent works by Zhang et al. (2018) and Verma et al. (2019) have studied the properties of regularization induced by considering convex combinations of (latent representations of) data samples. The authors argue that these techniques lead to learning “flatter” representations.

The idea of using data augmentation as a form of regularization is not new. Chapelle et al. (2000) introduced the unifying framework of *Vicinal Risk Minimization* (VRM) in which the empirical risk objective $\mathcal{R}_{\text{ERM}}(f)$ is modified via the introduction of an improved density estimate or *vicinal distribution*, $\mathbb{P}_{\text{est}}(x, y)$. The vicinal distribution acts as a probabilistic model from which artificial samples are generated towards regularizing the model f.

$$\mathcal{R}_{\text{ERM}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i) \qquad \Rightarrow \qquad \mathcal{R}_{\text{VRM}}(f) = \mathbb{E}_{(x,y)\sim\mathbb{P}_{\text{est}}}\left[\ell(f(x), y)\right] \tag{1}$$

The properties of the vicinal distribution $\mathbb{P}_{\text{est}}$ induce different learning biases. For example, a Parzen windows estimate, corresponding to the addition of independent Gaussian noise to the covariates, is equivalent to ridge regularization in linear regression (Chapelle et al., 2000).

In *mixup*, Zhang et al. (2018) propose a vicinal distribution via a *data-agnostic* augmentation procedure based on random convex combinations of arbitrary datapoints. For example, for a multi-class classification problem with inputs x_i and one-hot targets y_i, *mixup* generates virtual training examples $(\tilde{x}, \tilde{y})$ by sampling indices i and j independently and setting:

$$(\tilde{x}, \tilde{y}) = \lambda(x_i, y_i) + (1-\lambda)(x_j, y_j), \tag{2}$$

where λ follows a distribution with support on $[0, 1]$, such as $\lambda \sim \text{Beta}(\alpha, \alpha)$, for $\alpha > 0$.

In spite of their successful empirical results, the use of the *mixup* as a regularization scheme is mainly justified heuristically. In this work, we study regularization based on the **preservation of topological structures** present in the dataset. Our main contribution[1] is the introduction of **simplicial regularization**: we show how the preservation of the simplicial complex structure of the data naturally gives rise to a regularization framework that generalizes *mixup*. Figure 1 illustrates the differences between sets of samples produced by the Parzen windows estimate, *mixup*, and our proposed simplicial regularization scheme.

[*]Part of this research was done as a MSc student at the University of Amsterdam.

[1]Our code is available at: https://github.com/jgalle29/simplicial_regularization

SIMPLICIAL REGULARIZATION

The core ingredient of *mixup*'s vicinal distribution is the generation of random convex combinations of uniformly and *independently* selected datapoints form the dataset. Although this independent sampling contributes to the simplicity of the method, it is a naïve assumption. There is a priori no reason that the addition of two arbitrary datapoints should carry semantic meaning, see the middle panel of Figure 1. Furthermore, even when the semantics of the input resulting from said addition can be determined, they may not coincide with any of the two original labels.

Simplicial regularization addresses both of these difficulties by giving up on the independence assumption. Instead, we 1) infer a simplicial complex on the dataset following UMAP (McInnes et al., 2018), and 2) use the localized, correlated structure of this simplicial complex to design a vicinal distribution which better reflects the manifold-like properties of real data distributions.

From now on, we concentrate on a a supervised learning task for which we have access to a dataset of input-target pairs $\mathfrak{D} = \{(x_i, y_i)\}_{i=1}^N$ and a model class $\{f : \mathcal{X} \to \mathcal{Y} \,|\, f \in \mathcal{F}\}$, where $\mathcal{X}$ and $\mathcal{Y}$ are affine spaces, and $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a loss function. In the following we present the main ingredients of our proposed simplicial regularization framework. We refer the readers to Appendix A.1 for definitions of the concepts from algebraic geometry and fuzzy set theory employed below.

INFERRING THE SIMPLICIAL COMPLEX

UMAP is a dimensionality reduction technique, scalable to real world data and competitive with t-SNE (van der Maaten & Hinton, 2008). Given a dataset $\Phi = \{\phi_i\}_{i=1}^M$ embedded in a metric space, UMAP constructs a *sparse*, fuzzy simplicial complex $\mathcal{K}_\Phi = \texttt{FuzzTop}(\Phi)$. Informally, the information in $\mathcal{K}_\Phi$ reflects the $\tilde{n}$-nearest neighbors structure of Φ. Then UMAP proceeds to optimize the low-dimensional representations $\Psi = \{\psi_i\}_{i=1}^M$ such that the corresponding fuzzy simplicial complex $\mathcal{K}_\Psi = \texttt{FuzzTop}(\Psi)$ matches $\mathcal{K}_\Phi$ as closely as possible. In other words, we want the local neighborhood structure on the high and low-dimensional spaces to agree. Appendix A.2 provides an overview of the inner workings of the `FuzzyTop` algorithm used to construct fuzzy topological representations of metric spaces.

The first step for applying simplicial regularization on the learning of the model $f : \mathcal{X} \to \mathcal{Y}$ is to construct the fuzzy simplicial complex representation of our training set $\mathcal{K}_\mathfrak{D} = \texttt{FuzzTop}(\mathfrak{D})$[2]. Each of the simplices $\sigma \in \mathcal{K}$ corresponds to a subset of the training datapoints and is endowed with a value $\mu_{\mathcal{K}_\mathfrak{D}}(\sigma) \in [0, 1]$ denoting the membership of the simplex σ as an element of the set $\mathcal{K}_\mathfrak{D}$. Furthermore, the membership of a simplex is upper bounded by the lowest membership of all its constituent lower-order simplices: for all $\sigma' \subset \sigma \in \mathcal{K}_\mathfrak{D}$ we have that $\mu_{\mathcal{K}_\mathfrak{D}}(\sigma) \leq \mu_{\mathcal{K}_\mathfrak{D}}(\sigma')$.

The simplicial complex formalism provides a natural generalization of *mixup* to higher-dimensional simplices. In fact, *mixup* can be interpreted as a complete graph which contains simplices of order 0 and 1 (the datapoints and all edges between them), with a constant membership function. In contrast, $\mathcal{K}_\mathfrak{D}$ is carefully constructed so that the local neighborhood structure, and thus the generated linear interpolations, respect the manifold structure of the dataset. The sparsity level and the maximum simplex order of $\mathcal{K}_\mathfrak{D}$ are determined by the neighborhood size $\tilde{n}$. This allows the method to scale to large datasets by avoiding the storage of a quadratic number of interactions among the datapoints.

DESIGNING THE VICINAL DISTRIBUTION

A morphism is a map between mathematical objects, which preserves the internal structure of the domain when transforming in to the codomain. In set theory, morphisms are functions; in linear algebra, the vector space structure is preserved by linear transformations; while in topology, continuous functions preserve the neighborhood structure. In this work, we advocate for a regularization paradigm that aims to preserve the inferred simplicial complex structure of the dataset, $\mathcal{K}_\mathfrak{D}$.

Consider two arbitrary simplicial complexes $\mathcal{K} \subset \mathcal{X}$ and $\mathcal{L} \subset \mathcal{Y}$. The structure-preserving functions between simplicial complexes are called **simplicial maps**. A simplicial map $f : \mathcal{K} \to \mathcal{L}$ is a function which acts linearly on convex combinations over the simplices of $\mathcal{K}$. Note that we do *not* require

[2] We may decide to only use the inputs x_i and not the targets y_i when constructing $\mathcal{K}_\mathfrak{D}$, so that $\mathcal{K}_\mathfrak{D}$ can be reused for different learning tasks in which the target information might change.

f to be a globally linear transformation, but only to *commute with convex combinations of vertices forming a simplex in* $\mathcal{K}$. In other words, if σ is a k-simplex of $\mathcal{K}$ with (ordered) vertex set σ^0, we require that for every convex coefficient k-vector $\boldsymbol{\lambda}$:

$$f\left(\boldsymbol{\lambda}^\top \sigma^0\right) := f\left(\sum_{j=1}^{k} \lambda_j \sigma_j^0\right) = \sum_{j=1}^{k} \lambda_j f(\sigma_j^0) =: \boldsymbol{\lambda}^\top f\left(\sigma^0\right) \tag{3}$$

The symbol $f\left(\sigma^0\right)$ denotes the collection of evaluations of the predictor f on the *vertices* of the simplex σ. For example, for an M-class classification problem, $f\left(\sigma^0\right)$ consists of a matrix of size $\dim(\sigma) \times M$, where row i represents the predicted label distribution for the i-th vertex of σ. Equation (3) highlights the fact that a simplicial map is uniquely determined by a base function $f_0 : \mathcal{K}^0 \to \mathcal{L}$ operating over the vertices of $\mathcal{K}$, and its extension by convex interpolation on each simplex in $\mathcal{K}$.

When applying simplicial regularization to a supervised learning problem, the vertices of $\mathcal{K}^0_{\mathfrak{D}}$ correspond to the training datapoints $\{x_i\}_{i=1}^N$. Thus we take the base function as the training set labelling, i.e., $f_0(x_i) = y_i$ for $x_i \in \mathcal{K}^0_{\mathfrak{D}}$. The learning problem amounts to fit the model f to the extension of f_0 by convex interpolation over the simplices of $\mathcal{K}_{\mathfrak{D}}$.

Enforcing Equation (3) in practice is far from trivial. However, we can use the loss function ℓ to measure "how far the function f is from being a simplicial map". This suggests the definition of a vicinal risk which encourages f to **preserve the simplicial structure of** $\mathcal{K}_{\mathfrak{D}}$ when mapping to $\mathcal{L}$:

$$\mathcal{R}_{\text{SR}}(f \,|\, \mathcal{K}_{\mathfrak{D}}, f_0, \alpha) = \mathbb{E}_{\sigma \sim \mathcal{K}_{\mathfrak{D}}} \mathbb{E}_{\boldsymbol{\lambda} \sim \text{Dir}(\dim(\sigma), \alpha)} \left[\ell\left(f\left(\lambda^\top \sigma^0\right), \lambda^\top f_0\left(\sigma^0\right)\right)\right]. \tag{4}$$

The parameter $\alpha > 0$ inversely controls the level of interpolation which takes place within each simplex σ by imposing a symmetric Dirichlet distribution $\text{Dir}(\dim(\sigma), \alpha)$ over σ. For small values of α, the distribution has peaks at the corners of the simplex recovering the standard ERM setting, while for $\alpha > 1$, the distribution becomes more skewed towards the center of the simplex.

Equation (4) is presented in a suggestive notation to emphasize its connection to the VRM framework. In this case, the corresponding vicinal distribution is given by:

$$\mathbb{P}_{\text{est}}(\tilde{x}, \tilde{y} \,|\, \mathcal{K}_{\mathfrak{D}}, f_0, \alpha) \triangleq \int \mathbb{P}(\sigma \,|\, \mathcal{K}_{\mathfrak{D}}) \mathbb{P}(\boldsymbol{\lambda} \,|\, \sigma, \alpha) \underbrace{\mathbb{P}(\tilde{x} \,|\, \boldsymbol{\lambda}, \sigma)}_{\delta_{\boldsymbol{\lambda}^\top \sigma^0}(\tilde{x})} \underbrace{\mathbb{P}(\tilde{y} \,|\, f_0, \boldsymbol{\lambda}, \sigma)}_{\delta_{\lambda^\top f_0(\sigma^0)}(\tilde{y})} \,\mathrm{d}\sigma\, \mathrm{d}\boldsymbol{\lambda}. \tag{5}$$

Algorithm 1 describes the training pipeline for a supervised learning problem using simplicial regularization. Note how simplicial regularization is not embodied as an additive term alongside a supervised loss, but rather the regularized optimization objective is the vicinal risk itself. Appendix A.3 describes the `FSC-Sample` algorithm for efficient sampling from a fuzzy simplicial complex.

Algorithm 1: Training with Simplicial Regularization.

Data: Dataset $\mathfrak{D} = \{(x_i, f_0(x_i) := y_i)\}_{i=1}^N$, maximum iterations $i_{\max}$, learning rate η, optimizer $\mathcal{O}$ (e.g. SGD), Dirichlet interpolation schedule $\{\alpha_\tau\}_{\tau=1}^{i_{\max}}$.

Compute fuzzy simplicial complex $\mathcal{K}_{\mathfrak{D}} = \texttt{FuzzTop}(\mathfrak{D})$
Initialize predictor $f_\theta : \mathcal{X} \to \mathcal{Y}$
for $\tau = 1, \ldots, i_{\max}$ **do**
  Sample $\sigma \sim \mathcal{K}_{\mathfrak{D}}$ (using `FSC-Sample`($\mathcal{K}_{\mathfrak{D}}$)), and $\boldsymbol{\lambda} \sim \text{Dir}(\dim(\sigma), \alpha_\tau)$
  Compute Monte-Carlo estimator $\hat{\mathcal{R}}_{\text{SR}} = \ell\left(f\left(\lambda^\top \sigma^0\right), \lambda^\top f_0\left(\sigma^0\right)\right)$
  Perform optimizer step $\mathcal{O}$ on θ to minimize $\hat{\mathcal{R}}_{\text{SR}}$ with learning rate η
end

Remarks. 1) $\mathcal{K}_{\mathfrak{D}}$ only needs to be computed once per dataset and can be stored and re-used for multiple training tasks. Although *mixup* does not require this computation, usually this cost is significantly lower than that of the subsequent training loop. 2) It is possible to set up a schedule $\{\alpha_\tau\}$ for the Dirichlet interpolation coefficient, such that α_τ increases with τ during training. This schedule encourages the approximation f_θ to coincide with f_0 (the training labels) at the early stages of the optimization process, and then is progressively regularized in later iterations. 3) Unlike data augmentation methods relying on data invariances, such as translations or rotations for image data, our proposed augmentation framework is automatically inferred from geometry of the *dataset*.

EXPERIMENTS

In this section, we compare the performance of simplicial regularization and *mixup* on two tasks: i) multivariate regression on a synthetic dataset which has a simplicial complex structure and ii) multi-class digit classification.

Dimensionality Reduction. We generated a synthetic random point cloud X of size 100 in $\mathbb{R}^{20}$ and ran UMAP on this dataset to produce a 2-dimensional embedding $f_0(X) = Y$ (see Figure 2), as well as the high-dimensional simplicial complex $\mathcal{K}_X$. We then used several regularization methods to train neural networks of 2-hidden-layer neural and 100 units per layer to *approximate* the UMAP-generated embedding $X \to Y$. That is, we want to find a network f such that $f(x_i) \approx y_i$, where the discrepancy is measured by $l = || \cdot ||_2$. We compare between (unregularized) ERM, *mixup*, and simplicial regularization using the simplicial complex $\mathcal{K}_X$. For the latter two, we used an increasing schedule for the regularization parameter $\alpha \in [10^{-4}, 1]$ during training.

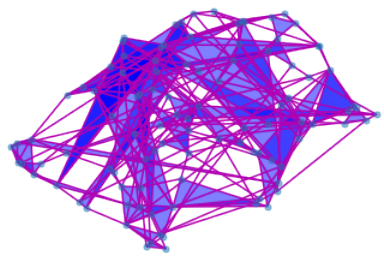

Figure 2: 2-dimensional UMAP projection of a random simplicial complex originally embedded in $\mathbb{R}^{20}$.

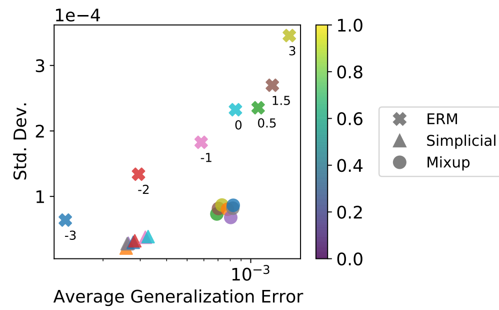

Figure 3: Effect of regularization for regression on a synthetic simplicial complex. Values by ERM markers denote evaluation interpolation strength $\log \alpha'$.

As a sanity test, we are interested in comparing the performance of these methods when applied to data with structure satisfying the simplicial complex assumption. We evaluate the generalization performance of these method according to the distributions $\mathbb{P}_{\text{est}}(\tilde{x}, \tilde{y} \,|\, \mathcal{K}_X, f_0, \alpha')$, using different interpolation strengths α'. Note that α' measures the *hardness* of the task since preserving the simplicial complex structure becomes more important as α' gets bigger and most of the evaluation points are sampled around the center of the simplices of $\mathcal{K}_X$. Moreover, α' is independent of the schedule for α used during training. The values of $\log \alpha'$ accompany the markers of ERM and matching color across different methods represent the same value of α' in evaluation

Figure 3 presents the mean and standard deviation of 30 rounds of evaluation of the generalization error of ERM, *mixup* and simplicial regularization. As expected, the performance of unregularized ERM degrades quickly as the interpolation strength increases. Moreover, we observe that the models trained using simplicial regularization outpeform those trained using *mixup* when the evaluation task is highly dependent on the simplicial complex structure of the data. In other words, when the unseen test data lies in the vicinity of *local* interpolations of the training data points, regularization methods agnostic to the topological structure of the data, like *mixup*, can lead to sub-optimal performance.

Digit Classification. In Figure 1 we illustrated the qualitative differences between the inputs produced by simplicial regularization, *mixup* and the Parzen windows estimate. In this section, we consider a digit classification task and expand our analysis of the perturbations on the data distribution induced by these techniques and the consequences at the *input* and *prediction* levels.

Figure 4 displays samples generated by interpolation of observed data points under the frameworks of simplicial regularization and *mixup* on the MNIST dataset (LeCun et al., 1998). We argue that the inputs produced by simplicial regularization correspond to *plausible, local* variations of the data at the observed samples, in line with the tangent space approximation of the underlying data manifold.

This is in stark contrast with the samples produced by *mixup*. The un-relatedness of the samples being combined leads to inputs which do not correspond to likely variations of the data distribution, and thus lack semantic meaning. Although it is possible to alleviate this issue by considering the use of *mixup* with low values of α, this approach leads to a vicinal distribution with *low interpolation* and thus, extremely close to the original data distribution, thus yielding un-regularized training. The low α regime does not address the main issue of *mixup*: its use of structure-agnostic interpolations.

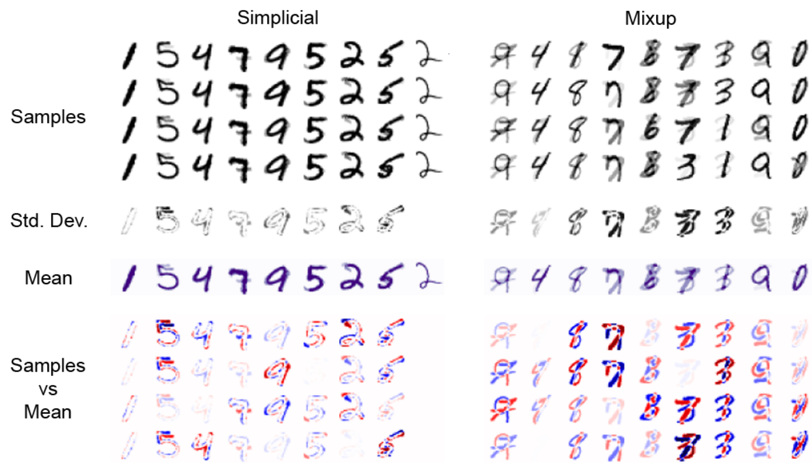

Figure 4: Inputs generated from the vicinal distributions of simplicial regularization and *mixup* with $\alpha = 1$. The variations in each column are due to sampling different convex combinations of the datapoints. In the last column of the left group a 0-simplex was sampled, leading to no interpolation.

The *mixup* scheme is also problematic regarding its proposed labels for the generated interpolations. The selection of random, unrelated, potentially very separated, data-points opens the possibility that some sections of the interpolation segment belong to regions of the input space which could be classified under a label which *does not coincide with those of the endpoints*. The target proposed by *mixup* is oblivious to this observation and induces the classifier to *only* produce predictions matching the classes of the sampled endpoints.

We trained two neural networks to classify MNIST digits using *mixup* and simplicial regularization with (weighted) cross-entropy loss. We then sampled uniform, independent triplets of points (as in the *mixup* setting) and plotted the model predictions over the simplex comprising all convex interpolations on each triplet. These results are presented in Figure 5. Although both networks reached high predictive accuracy and their predictions coincide on the corners of the simplices, they exhibit vastly dissimilar behavior when evaluated at convex combinations of these corner points.

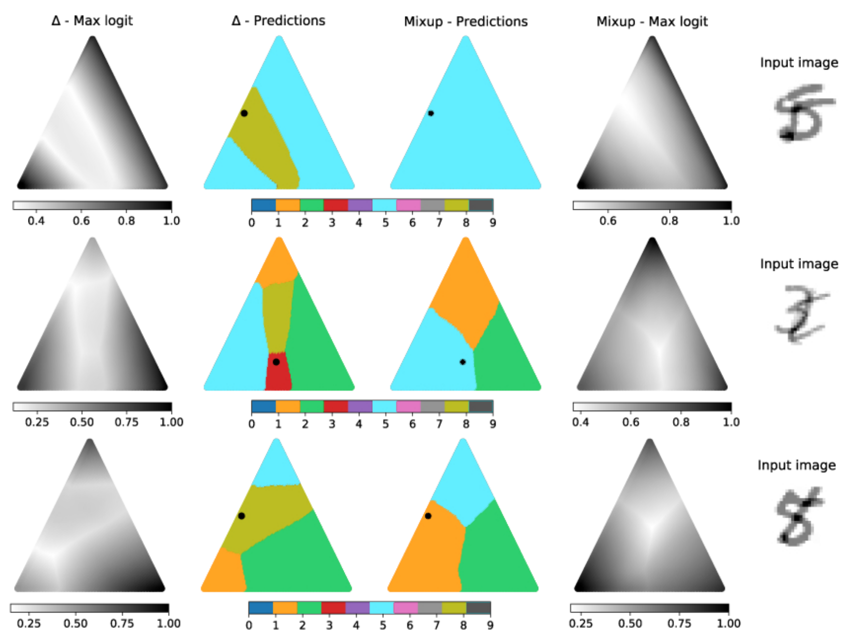

Figure 5: Behavior of classifiers on interpolation over randomly selected triplets. The image on the right corresponds to the convex combination represented by the black dot. Δ denotes simplicial reg.

Note how the *mixup* framework leads to classifiers which (frequently) only produce predictions which agree with the endpoint classes, even though a human observer could assign a different label. On the other hand, the local nature of the sampling in simplicial regularization allows for greater flexibility in the predictions of the model, as the training objective never aims to prescribe behavior on distant regions based solely on local information.

References

O. Chapelle, J. Weston, L. Bottou, and V. Vapnik. Vicinal Risk Minimization. In *NeurIPS*, 2000.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 1989.

K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2): 251–257, 1991.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv preprint 1802.03426*, 2018.

J.R. Munkres. *Elements Of Algebraic Topology*. Westview Press, 1995.

M. Telgarsky. Representation Benefits of Deep Feedforward Networks. *arXiv preprint 1509.08101*, 2015.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

V. Verma, A. Lamb, C. Beckham, A. Courville, I. Mitliagkas, and Y. Bengio. Manifold Mixup: Better Representations by Interpolating Hidden States. In *ICML*, 2019.

H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. *mixup*: Beyond Empirical Risk Minimization. In *ICLR*, 2018.

A Appendix

A.1 Definitions and Notation

For completeness, we provide definitions of several topological notions used in this manuscript. While many of these definitions can be formulated in category-theoretical terms, we favour the purely geometric language for the sake of accessibility to the general machine learning community. The definitions below follow closely the work of Munkres (1995).

A **$\mathcal{K}$-simplex** σ in $\mathbb{R}^n$ is the convex set spanned by $k+1$ linearly independent points $\{x_0, \dots, x_k\} \subset \mathbb{R}^n$. The points x_i are called *vertices*, and the convex set spanned by any non-empty subset of these vertices is called a *face* of the $\mathcal{K}$-simplex. The set of l-simplices contained in σ is denoted σ^l. For example, the set of vertices of σ is denoted σ^0 and σ^k only contains σ itself. For $k < l$, σ^l is empty.

The **standard** geometric $\mathcal{K}$-simplex, denoted Δ^k, is the convex hull of the canonical basis of $\mathbb{R}^{k+1}$.

A **simplicial complex** $\mathcal{K} \subset \mathbb{R}^n$ is a collection of simplices in $\mathbb{R}^n$, possibly of varying dimensions, such that: (i) every face of a simplex of $\mathcal{K}$ is also in $\mathcal{K}$, and (ii) the intersection of any two simplices of $\mathcal{K}$ is a face of each of them.

Intuitively, we can think of a simplicial complex $\mathcal{K}$ as made up of copies of standard simplices of several dimensions, glued together among some common faces. We can organize the relevant information about a simplicial complex into the different skeleta $\mathcal{K}^k$, for $k = 0, 1, \dots$, so that $\mathcal{K}^k$ is the set of all $\mathcal{K}$-simplices of $\mathcal{K}$.

The structure-preserving functions between simplicial complexes are called **simplicial maps**. Let $\mathcal{K}$ and $\mathcal{L}$ be geometric simplicial complexes. A simplicial map $f : \mathcal{K} \to \mathcal{L}$ is a function which acts linearly on convex combinations over the simplices of $\mathcal{K}$.

Algebraically, consider an arbitrary m-simplex $\sigma \subset \mathcal{K}$ and a point $x \in \sigma$. It is possible to represent x via its **barycentric coordinates** over σ as a weighted sum of the vertices σ^0: there exists a probability m-vector $\boldsymbol{t}^\sigma$, such that $x = \sum_{j=1}^{m} t_j^\sigma \sigma_j^0$. We say that f is a simplicial map if all x and for all σ containing x, we have that:

$$f(x) = f\left(\sum_{j=1}^{m} t_j^\sigma \sigma_j^0\right) = \sum_{j=1}^{m} t_j^\sigma f(\sigma_j^0). \tag{6}$$

Note that a simplicial map f is uniquely determined by a function $f_0 : \mathcal{K}^0 \to \mathcal{L}^0$ operating over the vertices, and its extension by convex interpolation on each simplex in $\mathcal{K}$.

Let $\alpha > 0$. We denote by $\mathrm{Dir}(k, \alpha)$ the **symmetric Dirichlet distribution** with density over the standard $k-1$ simplex given by $\frac{\Gamma(\alpha k)}{\Gamma(\alpha)^k} \prod_{i=1}^{k} x_i^{\alpha-1}$.

A **fuzzy set** is set S enriched with a membership function $\mu : S \to [0,1]$. Given fuzzy sets (S, μ) and (T, ν) a morphism of between them is a function $f : S \to T$ such that for all $s \in S$, $\mu(s) \leq \nu(f(s))$.

Recalling the notion of a simplicial complex as a collection of simplices, we define a **fuzzy simplicial complex** as a simplicial complex $\mathcal{K}$ endowed with a membership function $\mu_{\mathcal{K}} : \mathcal{K} \to [0,1]$, such that for all $\sigma \subset \sigma'$, $\mu_{\mathcal{K}}(\sigma) \geq \mu_{\mathcal{K}}(\sigma')$. In other words, the membership of a simplex is upper bounded by the lowest membership of all its constituent lower-order simplices.

A **strong negator** is a monotonous decreasing, involutive function $\neg : [0,1] \to [0,1]$ such that $\neg 0 = 1$ and $\neg 1 = 0$. For the remainder, we restrict our attention to the negator $\neg(a) = 1 - a$.

A **t-norm** is a symmetric function $\top : [0,1]^2 \to [0,1]$ satisfying for all $a, b \in [0,1]$:

- $\top(a, b) \leq \top(c, d)$ whenever $a \leq c$ and $b \leq d$,
- $\top(a, \top(b, c)) = \top(\top(a, b), c)$, and
- $\top(1, a) = a$.

Given a t-norm $\top$, its complementary **t-conorm** under a negator $\neg$ is defined by $\bot(a, b) = \neg\top(\neg a, \neg b)$. A triplet $(\top, \bot, \neg)$ where $\top$ is a t-norm, $\bot$ is a t-conorm, and $\neg$ is a strong negator is called a **De Morgan triplet** if for all $a, b \in [0,1]$ one has that $\neg\bot(a, b) = \top(\neg a, \neg b)$.

The most common example of a De Morgan triplet is the one formed by $\top_{\text{prod}}(a, b) = ab$, $\bot_{\text{sum}} = a + b - ab$ and $\neg(a) = 1 - a$. Note how the t-norm and t-conorm express the probability of intersection and union of independent events. Another important example arises by taking $\top_{\text{min}}(a, b) = \min(a, b)$, $\bot_{\text{max}} = \max(a, b)$.

A De Morgan triplet $(\top, \bot, \neg)$ structure allows us to define operations between sets, or rather, between membership functions μ and ν on a common universal set U. The **complement** of μ given by the composite function $\neg \circ \mu$. We define their **union** and **intersection** as the functions $\tau_{\mu\cup\nu} = \bot \circ (\mu, \nu)$ and $\tau_{\mu\cap\nu} = \top \circ (\mu, \nu)$, respectively.

A.2 Fuzzy representation of a dataset

The main building block of the UMAP dimensionality reduction technique is the construction of a carefully designed fuzzy topological representation of a given dataset $\mathfrak{X} \subset \mathbb{R}^n$. In practice, this representation is stored as a weighted graph with restricted neighborhood size. Next, we present a brief description of Algorithm 2. The steps involved in this construction are illustrated in Fig. 6. For more details, please consult sections 2.2 and 3.1 of McInnes et al. (2018).

Algorithm 2: `FuzzyTop`

Data: Dataset $\mathfrak{X} = \{x_i\}_{i=1}^{N} \subset \mathbb{R}^n$, number of nearest neighbors $\tilde{n}$.
Result: Fuzzy topological representation of $\mathfrak{X}$ given by $\mathcal{K}_{\mathfrak{X}}$.

```
for i = 1, ..., N do
    N_i = indices of ñ-nearest neighbors of x_i;
    Compute (X, d_i) ∈ FinEPMet ;          // Build FinEPMet d_i around i
    K_i = FinSing(X, d_i) ∈ sFuz ;          // Convert d_i into FSC
end
K_X = ⊥_{i=1}^N K_i ;                      // Combine all FSCs
```

We begin with a dataset $\mathfrak{X}$ of N points embedded in $\mathbb{R}^n$ and an arbitrary metric d. For each data point i, we construct its set of $\tilde{n}$ nearest neighbors $\mathcal{N}_i$ and record the distance to its closest neighbor in $\rho_i = \min_{j \in \mathcal{N}_i} d(x_i, x_j)$. We define a finite, extended, pseudo-metric space d_i around x_i as:

$$d_i(x_i, x_j) = \begin{cases} d(x_i, x_j) - \rho_i, & \text{for } j \in \mathcal{N}_i \\ \infty & \text{else.} \end{cases} \tag{7}$$

We define $\mu_{i,j} = \exp\left(-\frac{d_i(x_i, x_j)}{\sigma_i}\right)$ for $j \in \mathcal{N}_i$, and select $\sigma_i > 0$ in such a way that $\sum_{j \in \mathcal{N}_i} \mu_{i,j} = \log \tilde{n}$. These ingredients allow us to define a fuzzy simplicial complex $\mathcal{K}_i$ whose fuzzy 1-skeleton is given by the membership function $\mu_{i,(\cdot)}$. Note how all points in $\mathfrak{X}$ outside of the neighborhood $\mathcal{N}_i$ have membership zero.

Finally, this process is repeated for all data points in $\mathfrak{X}$ and the individual simplicial complexes are joined via a pre-determined t-conorm $\perp$.

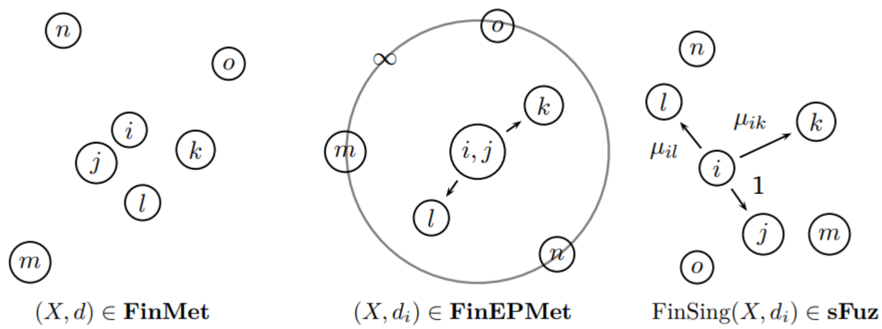

Figure 6: Visual representation of the fuzzy simplicial set construction around data point i within the UMAP framework. In this example $\tilde{n} = 3$.

A.3 Simplicial complex extension and sampling

Algorithm 2 and its output are manageable for large-scale datasets due to the restricted neighborhood size and the use of a 1-skeleton representation. Explicitly storing all simplices of higher dimensions quickly becomes a computational bottleneck due to the combinatorial number of possibilities. In this section we present a construction that extends a fuzzy 1-skeleton into a fuzzy simplicial complex, and an efficient algorithm to sample from it, while avoiding the explicit storage of memberships for high-dimensional simplices.

Algorithm 3: `FSC-Sample`

Data: 1-skeleton μ, neighborhoods $\mathcal{N}_i$, t-norm $\top$ (e.g. $\top_{\text{min}}$)
Result: A random simplex sampled from the fuzzy simplicial complex extended from μ.
Sample $u \in [1, \ldots, N]$ uniformly
$\mathcal{N} = \mathcal{N}_u \cup \blacksquare$
$\sigma = [u]$
while $\mathcal{N} \neq \{\blacksquare\}$ **do**
    foreach $t \in \mathcal{N}$ **do** $\nu_t = \top_{s \in \sigma} \mu_{s,t}$;
    $\nu_\blacksquare = 1$
    Sample u' without replacement from $\mathcal{N}$ with probabilities proportional to ν
    if $u' = \blacksquare$ **then**
        Return σ
    else
        Append u' to σ
    end
    $\mathcal{N} = (\mathcal{N} \cap \mathcal{N}_{u'}) \cup \blacksquare$
end

Consider a (sparse) symmetric $N \times N$ matrix μ representing a fuzzy 1-skeleton and denote by $\mathcal{N}_i$ the non-zero entries in row i. It is easy to see that under a t-norm $\top$, the recursive membership:

$$\mu_{\mathcal{K}}(\sigma) = \top_{\psi \in \sigma} \mu_{\mathcal{K}}(\psi) \qquad \text{for } \sigma \in \mathcal{K}^k,\ k > 2 \tag{8}$$

is a valid extension of μ onto a fuzzy simplicial complex. The base cases for these recursions are the 2-simplex memberships $\mu_{\mathcal{K}}([i, j]) = \mu_{i,j}$. Trivially, this construction implies that a simplex σ has non-zero membership if *all* its constituent sub-simplices have non-zero membership.

Note that Algorithm 3 builds the required high-dimensional simplices on the fly, and is guaranteed to terminate in at most $\tilde{n}$ iterations of the outermost cycle. The sampling of multiple simplices for the mini-batch setting can be easily parallelized.