Geometric Disentanglement by Random Convex Polytopes

Michael Joswig  
Technische Universität Berlin, Chair of Discrete Mathematics/Geometry  
MPI MiS Leipzig  
joswig@math.tu-berlin.de

Marek Kaluba  
Technische Universität Berlin, Chair of Discrete Mathematics/Geometry  
Adam Mickiewicz University, Poznań, Poland  
kaluba@math.tu-berlin.de

Lukas Ruff  
Technische Universität Berlin, Dept. of Electrical Engineering and Computer Science  
lukas.ruff@tu-berlin.de

Abstract

Finding and analyzing meaningful representations of data is the purpose of machine learning. The idea of representation learning is to extract representations from the data itself, e.g., by utilizing deep neural networks. In this work, we examine representation learning from a geometric perspective. Especially, we focus on the convexity of classes and clusters as a natural and desirable representation property, for which robust and scalable measures are still lacking. To address this, we propose a new approach called Random Polytope Descriptor that allows a convex description of data points based on the construction of random convex polytopes. This ties in with current methods for statistical disentanglement. We demonstrate the use of our technique on well-known deep learning methods for representation learning. Specifically we find that popular regularization variants such as the Variational Autoencoder can destroy crucial information that is relevant for tasks such as out-of-distribution detection.

1 Introduction

Finding a meaningful data representation $f(x) = z$ that captures the information that is relevant for some task or problem lies at the very core of any data-driven discipline such as statistics or machine learning. In contrast to handcrafted data features, representation learning [5] aims to learn such representations from the data itself. The concept of representation learning is also one of the fundamental principles of deep learning [33, 47], which set a new state of the art in recent years in numerous domains such as computer vision, speech recognition, or natural language processing. Here, representation functions $f_{\theta}$ are given by multi-layered neural networks that are parameterized by weights $\theta$ which are jointly optimized on some learning objective.

What makes a representation meaningful and which general principles are useful for representation learning is being actively discussed [5, 15, 53, 35, 54]. One prominent line of research considers the statistical nature of disentanglement [19, 41, 11, 30, 55, 35], which is rooted in the idea that a learned representation should “separate the distinct, informative factors of variations in the data” [35]. While this is a valuable notion, other important
properties of representation learning such as smoothness, the hierarchy of concepts, temporal and spatial coherence, or the preservation of natural clusters are also desired [5] and have been less studied. In this work, we focus on the convexity of classes and clusters in the image of deep neural network mappings and take a geometric perspective on representation learning.

Convexity is a very natural and fundamental representation property in machine learning in general. The separation of classes via multiple hyperplanes or half-spaces in supervised classification for instance, such as in Support Vector Machines (SVMs) [48] in the Reproducing Kernel Hilbert Space (RKHS) via the hinge loss or in deep neural networks (DNNs) [19] in the output space via the cross-entropy loss, forms polyhedra that are convex and generally unbounded. Moreover, many well-known unsupervised learning methods make explicit convexity assumptions on the support of the data representation, such as Voronoi cells in k-means clustering or ellipsoids in Gaussian Mixture Models (GMMs). Yet it is worth noting that convexity is also implicit in Gaussian prior assumptions in popular deep generative models such as Variational Autoencoders (VAEs) [27, 42, 53], Generative Adversarial Networks (GANs) [15, 13, 14], or Normalizing Flows [41, 28, 26]. Finally, one-class classification methods for support estimation such as the One-Class SVM [49] or (Deep) Support Vector Data Description (SVDD) [52, 44] also rely on convexity assumptions on the data representation via maximum-margin hyperplane and minimum enclosing hypersphere descriptions, respectively.

Although a convex representation of classes and clusters is such a natural and commonly desired property, measuring convexity in a robust and efficient way in high-dimensional spaces is a challenging and fairly non-trivial problem. An approach that would aim to exactly solve the convex hull problem of a set of data points \( \{x_1, \ldots, x_n\} \) in an idealized setting is prohibitively expensive to compute in high dimensions. Moreover, such an exact convex hull description would not be robust towards outliers and thus rather undesirable in terms of generalization.

In this work, we propose a scalable and robust method called Random Polytope Descriptor (RPD) for evaluating convexity in representation learning. Our method is based on concepts from convex geometry and constructs a polytope (a piecewise-linear, bounded convex body) around the training data in representation space. Since polytopes by themselves may also suffer from a combinatorial explosion, we construct our descriptor from random convex polytopes instead which also makes it more robust. Finally, using the proximities to such polytopes allows us to judge the geometric disentanglement of representations, i.e. how well classes and clusters are separated into convex bodies. Our main contributions are the following:

- We propose the Random Polytope Descriptor (RPD) method which is based on the construction of random convex polytopes for evaluating convexity in representation learning and theoretically prove its scalability.

- We demonstrate the usefulness of RPD in experiments on well-known supervised as well as unsupervised deep learning methods for representation learning.

- We find that popular regularization variants of deep autoencoders such as the Variational Autoencoder can destroy crucial geometric information that is relevant for out-of-distribution detection.

## 2 Related Work

How to evaluate the quality and usefulness of a representation is a difficult question to answer in general and subject of ongoing research [5, 15, 53, 55, 54]. If only one single task is of interest, a straightforward approach of course would be to evaluate the performance of a representation on some measure that is meaningful to the task at hand. Considering the supervised classification task for instance, one might evaluate the quality of a model and representation by using the accuracy measure, i.e. by how well the representation separates
some test data via hyperplanes in agreement with the respective ground-truth labels. However, even in such a well-defined standard task as classification, other representation properties such as the robustness against out-of-distribution samples [38, 20, 31, 41, 11] or adversarial attacks [50, 8, 6, 9], as well as interpretability aspects [32, 15] might be desirable and thus relevant representation quality criteria. This matter becomes even more challenging in the unsupervised or self-supervised setting, where the aim is learn more generic data representations that prove useful for a variety of downstream tasks (e.g., multi-task or transfer learning) [35, 43, 51].

Bengio et al. [5] have collected and formulated some well-known generic criteria, so-called generic priors for representation learning. These include smoothness (if $x \approx \tilde{x}$, then also $f(x) \approx f(\tilde{x})$), the hierarchy of concepts (e.g., for images going from pixel to object features), semi-supervised learning (representations for supervised and unsupervised tasks should align), temporal and spatial coherence (small variations across time and space should result in similar representations), the preservation of natural clusters (data generated from the same categorical variable should have similar representations), and statistical disentanglement (a representation should separate the distinct, informative factors of variations of the data). Different variants of the Variational Autoencoder (VAE) [27, 42, 21, 30, 25, 10] which especially emphasize statistical disentanglement are currently considered to be the state of the art for unsupervised representation learning [33, 34]. These approaches are closely related to earlier works on (non-linear) independent component analysis (ICA) which study the problem of recovering statistically independent (latent) components of a signal [12, 3, 22, 2, 24].

As mentioned above, the convexity of classes and natural clusters is another fundamental representation property that is often only expressed implicitly through certain model assumptions. The basic property that linear interpolations between data representations should result in representations from the same class or cluster naturally ties into the generic priors above. This also follows the simplicity principle from Bengio et al. [5] that “in good representations, the factors are related through simple, typically linear dependencies.”

3 Methods

The applicability of our method depends on the general clustering assumption, that the training data is composed of different unobserved groups (classes), and that the joint density of observed variables is a mixture of class-specific densities. This assumption might be exemplified by assuming that the data come from a mixture of multivariate multinomial distributions, which is the natural assumption in data analysis. Moreover we assume, that the super-level sets of the density function for each of the summands in the mixture form a convex set, except maybe for regions of very low probability.

The method is designed to work well on embeddings into feature space, where each of the classes are embedded into mutually disjoint convex sets. Such assumption is at the foundation of classical methods e.g. SVMs and kernel methods. However these methods do not take into account the finiteness of training data simply partitioning the whole feature space among classes, i.e. usually most classes are given infinite regions. While such feature might seem desired, as it affords generalization properties for models, it also increases the surface for adversarial and out-of-distribution attacks. Here we attempt to remedy this limitation by explicitly partitioning the feature space into compact convex regions and an unbounded (non-convex) “out-of-distribution” set.

The general idea is to use the convex hull $\text{conv}(X)$ of points from the given class and the distance to $\text{conv}(X)$ as dissimilarity score. As mentioned earlier, both the convex hull computation and distance from a polytope are prohibitively expensive in higher dimensions, therefore we will replace both of these steps by their approximations. In this section we introduce the definition of (approximate) dual bounding body of $X$ and show how such polytope can be turned into a descriptor separating the points in the set from the exterior. Instead of computing actual distance, we will use the substitute of scaling distance.
(a piecewise-linear scaling of euclidean distance) from a well defined central point of the polytope.

3.1 An idealized setting

We consider \( N \) distinct sets \( X_1, X_2, \ldots, X_N \subset \mathbb{R}^d \), where \( X_i \) is the set of samples of class \( i \), considered in feature space \( \mathbb{R}^d \). Associated with these classes are \( N \) convex sets \( P_1, P_2, \ldots, P_N \), where

\[
P_i = \text{conv}(X_i) = \left\{ \sum_{k=1}^{n} \lambda_k x_k \mid x_1, \ldots, x_n \in X_i, 0 \leq \lambda_1, \ldots, \lambda_n, \sum \lambda_k = 1 \right\}.
\]

If all pairwise intersections \( P_i \cap P_j \) are disjoint, for \( i \neq j \), then these convex sets provide perfect descriptors, provided that we have some oracle which can test whether a new sample \( z \in \mathbb{R}^d \) is contained in either of the sets \( P_1, P_2, \ldots, P_N \) or their complement.

In the applications below each class \( X_i \) will be finite, whence the sets \( P_i \) form convex polytopes; cf. [56] for general background. We will return to the more general setting later; but for now we will stick to finite sets and polytopes. In this case an oracle for checking \( z \in P_i \) is given by linear programming. While linear programs can be solved in (weakly) polynomial time, they are still somewhat expensive in high dimensions. If many samples need to be checked it is thus desirable to convert each set \( X_i \) into a description in terms of linear inequalities. A minimal encoding of this kind is given by the facets of the polytope \( P_i \).

Then the containment \( z \in P_i \) can be decided in \( O(m_i d) \) time, where \( m_i \) is the number of facets of \( P_i \). We assume that each set \( X_i \) affinely spans \( \mathbb{R}^d \), whence \( P_i \) is full-dimensional, and thus its facet description is unique.

Converting \( X_i \) into the facets of \( P_i = \text{conv}(X_i) \) is the convex hull problem. It is worth noting that the dual problem of computing the vertices of a polytope given in terms of finitely many linear inequalities (assuming that it is bounded) is equivalent to the convex hull problem by means of cone polarity. If we let \( n_i \) be the cardinality of \( X_i \), then McMullen’s upper bound theorem [37] says that

\[
m_i \in O \left( n_i^\left\lfloor d/2 \right\rfloor \right), \quad \text{if } d \text{ is considered constant.} \tag{1}
\]

That bound is actually tight as can be seen, e.g., from the cyclic polytopes. In view of cone polarity the same (tight) estimate holds for the number of vertices if the number of inequalities is given. This means that the number of vertices and the number of facets of a polytope may differ by several orders of magnitude.

This sketch of a classification algorithm is too naive to be useful in practice. Yet it is instructive to keep this in mind as a guiding principle. We identify three issues.

1. The idealized setup does not deal with outliers, i.e. is generally not robust.
2. Computing the convex hull of each class can be prohibitively expensive.
3. The \( N \) classes may not be well represented by their convex hulls.

Our main algorithmic contribution is a concept for addressing all three issues simultaneously. For empirical data on convex hull computations see [4].

3.2 Random polytope descriptors

As our main contribution we propose a certain convex polytope as an approximate description of some learned representation. Its algorithmic efficiency and its robustness to the presence of outliers can be tuned by a pair of parameters. Again we assume that \( X \subset \mathbb{R}^d \) is finite, and \( n = |X| \).
Definition 1. Let \( Y \subset \mathbb{S}^{d-1} \) be a set of \( m \) unit vectors chosen uniformly at random, and let \( \ell \) be a positive integer. The random polytope descriptor is the polyhedron

\[
\text{RPD}_{m,\ell}(X) := \{ v \in \mathbb{R}^d \mid \langle v, y \rangle \leq \ell \cdot \max_{x \in X} \{ \langle x, y \rangle \}, \ y \in Y \} \tag{2}
\]

where \( \ell \cdot \text{max} \) denotes the \( \ell \)-th largest element of the set.

Tacitly we will assume that \( \text{RPD}_{m,\ell}(X) \) is bounded. This is the case if and only if \( Y \) is positively spanning. It is worth noting that the random polytope descriptors form a variation of dual bounding bodies; cf. Section A.

We briefly sketch what it means to compute \( \text{RPD}_{m,\ell}(X) \), and which cost this incurs. First we need \( m \) random unit vectors, which can be found in \( O(m) \); cf. \cite[3.4.1.E.6]{20}. There is no reason to be overly exact here; so it is adequate to assume that the coefficients of the random vectors are constantly bounded. Given \( X \), we then evaluate \( mn \) scalar products to obtain the H-description \( (2) \) of \( \text{RPD}_{m,\ell}(X) \), resulting in a total complexity of \( O(mn) \). Throughout we take the number \( d \) of latent dimensions as a constant.

In the context of unsupervised anomaly detection (see Section 4) it is often beneficial to discard a fixed amount of data from \( X \), i.e., label them as anomalous. The \( \ell \cdot \text{max} \) function serves as a “threshold” to eliminate the most extreme points from the training set. For (semi-)supervised learning there may be natural candidate functions to replace \( \ell \cdot \text{max} \) to define variants of RPD, e.g., based on density estimation, weights, or simply the label. However, for simplicity we stick to unsupervised anomaly detection.

Figure 1: Convex hull \( \text{conv}(X) \) (red) and random polytope descriptor \( \text{RPD}_{80,3}(X) \) (green). Here \( d = 3 \), and \( X \sim \mathcal{N}(0, 1) \) is a random Gaussian sample of 150 points with mean 0 and variance 1. In spite of 3% outliers, the descriptor closely approximates the convex hull of the normal class (i.e., the inner cluster).

3.3 Scaling distance and anomaly scores

Let \( P \subset \mathbb{R}^d \) be a full-dimensional convex polytope with a distinguished interior point \( c \). The scaling distance of \( z \in \mathbb{R}^d \) to \( P \) with respect to \( c \) is the quantity

\[
\text{sd}_c(z, P) := \min \{ \alpha \geq 0 \mid z \in \alpha(P - c) + c \} \tag{3}
\]

This is the smallest number \( \alpha \geq 0 \) such that \( P \) inflated by a factor of \( \alpha \) from the central point \( c \) contains \( x \). The point \( z \) is contained in \( P \) if and only if \( \text{sd}_c(z, P) \leq 1 \). If \( P \) is a descriptor, e.g., as in Definition 1 this may serve as a dissimilarity (anomaly) score, where the points further from \( P \) are assigned higher scores. Given \( z, c \) and an H-description of \( P \)
in terms of \( m \) linear inequalities, we can compute \( \text{sd}_c(z, P) \) in \( O(m) \) time, for \( d \) constant, by evaluating scalar products.

There are several natural candidates for the central point of an arbitrary polytope \( P \), which differ with respect to computational complexity, when \( P \) is given in terms of inequalities. For instance, the \textit{centroid} is the center of gravity of \( P \); this is hard even to approximate, cf. [39]. The \textit{vertex barycenter} is the average of the vertices of \( P \); this is hard to compute exactly, cf. [17]; further results in loc. cit. make it unlikely that there is an efficient approximation procedure either. The \textit{Chebyshev central point} is the center of the largest sphere inscribed to \( P \); this can be computed in \( O(\sqrt{m}) \) time by linear programming, cf. [16] and [40]. Note that the latter worst case complexity assumes that the bit representation of each inequality in the H-description of \( P \) is constantly bounded. For the special case of \( P = \text{RPD}_{m,d}(X) \), where \( X \) is a normally distributed random sample, all the three central points discussed are arbitrarily close to the origin, provided that \( m \) is sufficiently large; cf. Fig. 1.

Let \( X = \bigcup_i X_i \) be the training data partitioned into classes. For the \textit{estimated distance} of a point \( z \in \mathbb{R}^d \) to the combined support of all classes we take the minimum

\[
\delta(z) := \min_i \{\text{sd}_{c_i}(z, P_i)\}
\]

of the scaling distances, where \( P_i = \text{RPD}_{m,\ell}(X_i) \), and \( c_i \) is the corresponding Chebyshev central point. Note that \( \delta(z) \) is a random variable as the construction of polytopes \( P_i \) is not deterministic. If the estimated distance to the combined support is larger than one, then evaluating (4) also assigns a class which is closest to \( z \), namely the index \( i \) at which the minimum is attained. For generic \( z \) this is unique. It follows from our complexity analysis above that estimating the distance to the combined support takes \( O(m) \) time, for \( d \) and \( N \) fixed.

### 3.4 Convex separation

Since we do not use any a priori knowledge on the latent representation that we work with, it may happen that the random polytope descriptors of two (or more) classes intersect nontrivially. Again let \( X = \bigcup_i X_i \) be partitioned into classes and \( P_i = \text{RPD}_{m,\ell}(X_i) \) are the random polytope descriptors, for some fixed values of \( m \) and \( \ell \). Then we can define the \textit{confusion coefficient} of classes \( i \) and \( j \) as

\[
\gamma_{ij} := \frac{|X_i \cap P_j| + |X_j \cap P_i|}{|X_i \cup X_j|};
\]

i.e., a low value indicates good separation. This may be seen as a Monte Carlo style approximation of the integral of the probability density function of the test data over the intersection \( P_i \cap P_j \). Again by evaluating scalar products, computing one confusion coefficient is in \( O(mn) \).

### 4 Experiments

We conduct experiments on the MNIST and FMNIST datasets, to assess the feasibility of our method. Both datasets, MNIST and FMNIST, comprise ten classes each, i.e., \( N = 10 \). It should be stressed that we are primarily interested in the overall reliability of the classification process rather than individual performance scores.

We train standard AE and VAE networks to embed the MNIST and FMNIST datasets into a \( d \)-dimensional latent space for various choices of \( d \). The autoencoder architectures are of LeNet-type, having an encoder that has two \( 5 \times 5 \) convolutional layers with max-pooling followed by two fully connected layers that map to an encoding of \( d \) dimensions. The decoder is constructed symmetrically where we replace convolution and max-pooling with deconvolution and upsampling respectively. We use batch normalization [23] and leaky ReLU activation functions these networks. For training, we take the entire training datasets,
minimizing the reconstruction error (AE) and the reconstruction error regularized by the volume of the support of latent distribution (VAE). None on the networks are given any explicit convexity prior.

It is worth noting that constructing the H-representation of the random polytope descriptor is very fast. On modern hardware, it takes less than 3 seconds to create an RPD\(_{640,2}(X)\) from a set \(X\) of 6000 points in \(\mathbb{R}^{20}\). This includes computing its Chebyshev central point already. Evaluating the scaling distance takes less than 0.001s per sample in this scenario.

4.1 Out-of-distribution detection

Our method suggests to consider the random polytope descriptors as a model for the supports of the individual class distributions in the training data. Here we check how well this approximation works to detect out-of-distributions samples.

Our experimental setup is as follows. For each of the ten classes of FMNIST training data \(X\) we constructed a random polytope \(\text{RPD}_{m,1}(X_i)\). Since this is a supervised learning scenario it’s natural to chose \(\ell = 1\). In further experiments (see Section 4.2 and specifically Fig. 9) increasing \(m\) beyond 20\(d\) (where \(d\) is the dimension of the latent representation) had little to no influence on the performance, thus we settled on choosing \(m = \max(320, 20d)\). Then we evaluated the estimated distance to the support for test data from the FMNIST and MNIST datasets. The results are depicted in Fig. 3.

**Comparison of AE and VAE networks.** The minimal scaling distances for samples from FMNIST stay well below 1 for both networks; recall from [3] that this signals containment in one of the classes. So this behavior is expected and a first hint at a reasonable performance of random polytope descriptors for anomaly detection; cf. Section 4.2 for details.

However, we observe a very clear distinction between the AE and VAE networks with respect to out-of-distribution samples. Using AE network trained on FMNIST dataset, out of distribution MNIST test data are given significantly larger dissimilarity score. In sharp contrast the VAE network embedding of MNIST test set exhibits almost complete overlap with the support of the FMNIST training distribution. This means that the regularization step, which distinguishes VAE from AE, destroys information which may be exploited to...
perform out-of-distribution attacks. We will come back to this when we will look at anomaly scores in Section 4.2.

It is worth noting that the FMNIST scores are not concentrated around 0 (as one would observe, e.g., in the case of a Gaussian distribution). Instead they seem to be distributed near a sphere of some radius increasing with the dimension. For instance, for the VAE networks mean and variance range from (0.68, 0.03) to (0.85, 0.006)).

Convex separation in classical networks. We picked ten different classes from the CIFAR-100 dataset, which to intuitive human understanding should be relatively close to each other. We evaluated convex separation of pre-trained, state-of-the-art image recognition networks: AlexNet, ImageNet and Resnet18 to form “confusion matrix” present below. Note that high confusion means that no linear separating hyperplane will provide good distinction between the classes.

4.2 Anomaly detection

As a second benchmark for the disentanglement of the learned representation we used an anomaly detection test. For each MNIST and FMNIST class an unsupervised anomaly detection scenario was created as follows. The training data \(X_i\) consisted of all training data points from class \(i\) “enriched” by \(p|X_i|\) data points chosen uniformly at random from the other classes, for some constant \(0 < p < 1\). This noisy data is then used to create a random polytope descriptor \(P_i = \text{RPD}_{m,l}(X_i)\), and its accuracy against the entire test data is evaluated in the AUC metric (area under receiver operating characteristic curve). This experiment may be understood as a stochastic approximation of the intersection \(P_i \cap \bigcup_{j \neq i} P_j\). Indeed, if two random polytope descriptors exhibit an intersection witnessed in the test data, the data points in the intersection will be counted as false negatives in the anomaly detection task.

All experiments have been conducted with the following choice of the parameters: \(p = 0.02\), \(\ell = 2\), and a grid of parameters \((m, d)\) in the range \(80 \leq m \leq 640\) and \(d \in \{5, \ldots, 20\}\). The results are shown in Figs. 6 to 9.

Comparison of AE and VAE networks. When fixing \(m\), the number of hyperplanes, in Figs. 6 and 7 we can see the monotonous increase of scores with the dimension of the latent space. This aligns with the common understanding that higher the dimension, the better is reconstruction, hence more faithful representation of data.

When fixing \(d\), the dimension, we can observe the dependence of the AUC score on the number of hyperplanes \(m\) defining an \(\text{RPD}_{m}\). The scores increase monotonously with \(m\) both for MNIST Fig. 8 and FMNIST Fig. 9 datasets. This agrees with our intuition, as with a larger number of hyperplanes the approximate bounding body tightens its approximation of the true convex hull. It seems that setting the number of hyperplanes \(m\) above 20\(d\) does not lead to significant improvement in performance.

For the MNIST dataset it seems that combining our descriptors with VAE is able separate the clusters in dimension \(d = 9\) with improvement up to \(d = 12\). Provided that the assumption on the convexity of clusters is satisfied, the deterioration of the performance for larger \(ds\) (for a fixed \(m\)) is expected, as in higher dimensions a larger number of hyperplanes is needed for similar quality of approximation. On the other hand, AE in general performs much poorer in this scenario but its performance does not deteriorate. The difference in behavior of AE and VAE networks suggest that the deterioration of performance for higher dimensions may be the consequence of overfitting by VAE networks.

A similar experiment on FMNIST, arguably more complex dataset, shows (Fig. 7) no deterioration of performance while increasing the dimension, and overall better performance of VAE networks.

A possible explanation of the phenomenon we observe with VAE networks is as follows. Since VAE networks try to pack data close to the origin, all classes necessarily will be in close
proximity to each other. It is possible though that while achieving simple (convex) shapes of clusters in low dimensions, more space in higher results in complex shapes of embedded clusters, which might be an indicator of overfitting.

The sustained performance of VAE networks for FMNIST dataset might be an indicator that the intrinsic dimension of the dataset is higher than what have been studied here. On the practical side of network design, one may argue that the regularization in VAE network is too strong for MNIST dataset.

4.3 Contracting property of autoencoders

Autoencoder networks can be seen as pair of networks \( \phi : \mathbb{R}^n \to \mathbb{R}^d, \psi : \mathbb{R}^d \to \mathbb{R}^n \) with

\[
\psi \circ \phi : \mathbb{R}^n \to \mathbb{R}^d \to \mathbb{R}^n
\]

optimized to be close to the identity (on the support of the training distribution). One may ask for the image of the reversed composition

\[
\phi \circ \psi : \mathbb{R}^d \to \mathbb{R}^n \to \mathbb{R}^d,
\]

which follows the path of adversarial learning.

In particular we are interested in measuring how far from the identity is the composition \( \Phi = \phi \circ \psi \), and whether the image of the map is closer to the support of the training data. In the second case we may say that the network has contracting properties. Since the RPD’s provide a natural “calibration” of the distance, we can sample vertices of an RPD at random (which are all of scaling distance 1) and observe the distribution of scaling distance of the image of \( \Phi \) of those vertices.

Here we compare the behavior of RPDs with a classical \( k \)-means descriptor (based on Voronoi centers); see Fig. 3. We start with a sample of vertices \( V \) of a RPD for a fixed class (and hence of scaling distance to that class equal to 1) and apply map \( \Phi \) to them. For comparison we also produce the histograms for (scaled) Euclidean distances to Voronoi center of a class. Such experiments are depicted in Fig. 5. Notice that while both networks exhibit strongly contractive behavior, the VAE network it is especially pronounced. The simplest explanation is networks nearly constant response on the entire Voronoi cell.

Acknowledgments

We thank Klaus-Robert Müller for fruitful discussions. MJ received partial support by Deutsche Forschungsgemeinschaft (EXC 2046: “MATH++”, SFB-TRR 195: “Symbolic Tools in Mathematics and their Application”, and GRK 2434: “Facets of Complexity”). MK was supported by Deutsche Forschungsgemeinschaft (EXC 2046: “MATH++”, Project EF1-3) and National Science Centre, Poland grant 2017/26/D/ST1/00103. LR acknowledges support by the German Federal Ministry of Education and Research (BMBF) in the project ALICE III (01IS18049B).

References

[1] B. Assarf, E. Gawrilow, K. Herr, M. Joswig, B. Lorenz, A. Paffenholz, and T. Rehn. Computing convex hulls and counting integer points with polymake. Math. Program. Comput., 9(1):1–38, 2017. doi: 10.1007/s12532-016-0104-z. URL http://dx.doi.org/10.1007/s12532-016-0104-z

[2] F. R. Bach and M. I. Jordan. Kernel independent component analysis. Journal of Machine Learning Research, 3(Jul):1–48, 2002.

[3] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. Neural Computation, 7(6):1129–1159, 1995.
[4] Y. Bengio, Y. LeCun, et al. Scaling learning algorithms towards ai. Large-Scale Kernel Machines, 34(5):1–41, 2007.

[5] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8):1798–1828, 2013.

[6] B. Biggio and F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. Pattern Recognition, 84:317–331, 2018.

[7] K.-H. Borgwardt. The simplex method, volume 1 of Algorithms and Combinatorics. Springer-Verlag, Berlin, 1987. ISBN 3-540-17096-0. doi: 10.1007/978-3-642-61578-8. URL https://doi.org/10.1007/978-3-642-61578-8 A probabilistic analysis.

[8] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In IEEE Symposium on Security and Privacy, pages 39–57. IEEE, 2017.

[9] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin. On evaluating adversarial robustness. arXiv preprint arXiv:1902.06705, 2019.

[10] T. Q. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud. Isolating sources of disentanglement in variational autoencoders. In Advances in Neural Information Processing Systems, pages 2610–2620, 2018.

[11] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Advances in Neural Information Processing Systems, pages 2172–2180, 2016.

[12] P. Comon. Independent component analysis, a new concept? Signal processing, 36(3):287–314, 1994.

[13] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. In International Conference on Learning Representations, 2017.

[14] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville. Adversarially learned inference. In International Conference on Learning Representations, 2017.

[15] C. Eastwood and C. K. Williams. A framework for the quantitative evaluation of disentangled representations. In International Conference on Learning Representations, 2018.

[16] B. C. Eaves and R. M. Freund. Optimal scaling of balls and polyhedra. Math. Programming, 23(2):138–147, 1982. ISSN 0025-5610. doi: 10.1007/BF01583784. URL https://doi.org/10.1007/BF01583784

[17] K. Elbassioni and H. R. Tiwary. Complexity of approximating the vertex centroid of a polyhedron. In Algorithms and computation, volume 5878 of Lecture Notes in Comput. Sci., pages 413–422. Springer, Berlin, 2009. doi: 10.1007/978-3-642-10631-6_43. URL https://doi.org/10.1007/978-3-642-10631-6_43

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems, pages 2672–2680, 2014.

[19] I. Goodfellow, Y. Bengio, and A. Courville. Deep Learning. MIT press, 2016.
[20] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

[21] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. $\beta$-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

[22] A. Hyvärinen and P. Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999.

[23] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[24] I. Khemakhem, D. P. Kingma, R. P. Monti, and A. Hyvärinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, 2020.

[25] H. Kim and A. Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658, 2018.

[26] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018.

[27] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[28] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751, 2016.

[29] D. E. Knuth. *The art of computer programming. Vol. 2*. Addison-Wesley, Reading, MA, 1998. ISBN 0-201-89684-2. Seminumerical algorithms, Third edition [of MR0286318].

[30] A. Kumar, P. Sattigeri, and A. Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2017.

[31] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

[32] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller. Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications*, 10:1096, 2019.

[33] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[34] K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177, 2018.

[35] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pages 4114–4124, 2019.
A Dual Bounding Bodies

A first key idea is to replace convex hulls by suitable dual bounding bodies, which form outer approximations. To explain the concept it suffices to consider a single class. Let \( X \) be a set of points in \( \mathbb{R}^d \), and let \( Y \subset S^{d-1} \) be a finite set of directions.

**Definition 2.** The dual bounding body of \( X \) with respect to \( Y \) is the polyhedron

\[
D_Y(X) = \left\{ v \in \mathbb{R}^d \left| \langle v, y \rangle \leq \sup_{x \in X} \langle x, y \rangle \text{ for } y \in Y \right. \right\}.
\]  

By construction \( D_Y(X) \) is convex, and it contains \( X \); so it contains \( P := \text{conv}(X) \), too. If \( X \) is finite then the supremum in (6) is a maximum, \( \mu_y \), that is attained for some \( x \in X \), which only depends on \( y \). In that case \( D_Y(X) = P \) if and only if \( Y \) contains all facet normals of \( P \). Checking if \( z \in \mathbb{R}^d \) is contained in \( D_Y(X) \) takes \( O(md) \) time for \( m = |Y| \).

In the sequel we will assume that both, \( X \) and \( Y \), are finite, and \( Y \) is positively spanning, i.e., \( \text{conv}(Y) \) contains the origin in its interior. As \( X \) is finite the latter property is satisfied if and only if the polyhedron \( D_Y(X) \) is bounded, i.e., a polytope.

**Remark 1.** If \( X \) is finite, positively spanning and contained in \( S^{d-1} \), then \( D_X(X) \) is the polar dual of the polytope \( P = \text{conv}(X) \). The facets of \( D_X(X) \) are tangent to \( S^{d-1} \), and \( \mu_x = 1 \) for all \( x \in X \).

B Random Polytopes

The following observation will turn out to be practically relevant. It says that our random polytope descriptors are combinatorially benign, in contrast to the worst case scenario described by the upper bound theorem [1].

**Theorem 3.** Let \( X \sim N(0, 1) \) be normally distributed with mean zero (and constant variance). For arbitrary \( m \) and \( \ell \) the number of vertices of \( \text{RPD}_{m,\ell}(X) \) is in the order of \( \Theta(m) \), for \( d \) considered constant.

**Proof.** The normal distribution is rotationally invariant, and so is the uniform choice of directions \( Y \subset S^{d-1} \). Consequently, the construction of \( \text{RPD}_{m,\ell}(X) \) follows the Rotation-Symmetry Model of Borgwardt [7], and this yields the claim.

We use the stochastic approach of [7] of random polytope. There, a random model of polytopes is defined as follows: we say that

\[
P \sim P(d, n) \iff P = \text{conv}(X)
\]

where \( X \) is chosen uniformly over \( S^{d-1} \subset \mathbb{R}^d \).
Let \( X \sim P(d, n) \) and assume that \( d_H(X, S^{d-1}) < h \), where \( d_H \) denotes the Haussdorff distance. This may be read that the height of the largest spherical cap cut from \( S^{d-1} \) by a facet of \( X \) is at most \( h \). Then given \( 0 < p < 1 \) with probability \( 1 - p \)

\[
d_H(D_Y(X), S^{d-1}) < h(d, m, p),
\]

where \( Y \subset S^{d-1} \) is a set of \( m \) points chosen uniformly at random and

\[
h(d, m, p) = \inf \left\{ h \mid \binom{m}{d} (1 - \text{NCA}_d(h))^{m-d} \leq \frac{p}{2} \right\}.
\]

Here \( \text{NCA}_d(h) \) is the area of a spherical cap of height \( h \) normalized so that \( \text{NCA}_d(1) = 1 \) regardless of dimension.

**Theorem 4.** Let \( P \sim P(d, m) \) and let \( \varepsilon > 0 \) and \( 0 < p < 1 \) be given and let \( X \) denote the set of its vertices. Denote by \( Y \) a set of \( m \) directions chosen uniformly at random and by \( h = \sup_v \|v\| - 1 \) the maximal distance over all vertices of \( D_Y(X) \) to the unit sphere. Note that all facets of \( D_Y(X) \) are contained in a \((1, 1 + h)\) \( d \)-dimensional annulus around the origin. The mean of \( k \) randomly chosen vertices of \( D_Y(X) \) is at most \( \varepsilon \) from the barycenter of \( D_X(X) \) with probability at least \( 1 - p \) if

\[
k > \left( 1 + \frac{2}{d} \log \left( \frac{2}{p} \right) \frac{e}{e-1} \cdot \frac{1}{\varepsilon^2(1-h_0)^2} \right),
\]

where \( h_0 \) is the Haussdorff distance of \( P \) to the sphere \( S^{d-1} \).
Figure 3: Results of the out-of-distribution detection experiment. AE and VAE networks were trained on FMNIST data and used to embed MNIST test sets. The plot depicts the distribution of minimal scaling distance to one of the ten FMNIST random polytope descriptors in various dimensions up to 20 for fixed $(m, \ell) = (320, 1)$. Five distinct AE and VAE networks were trained and independently five random polytopes were drawn for each of the dimensions. The lower plot shows a similar experiment with a simple descriptor based k-means, which can be understand as Voronoi counterpart to polytopal descriptors. The vertical axis is the dimension normalized Euclidean distance defined as $d(x, y) = \sqrt{\frac{1}{n} \sum_i (x_i - y_i)^2}$ for points $x, y \in \mathbb{R}^n$. 
Figure 4: Similar experiments to Fig. 3 performed with for CIFAR10 and SVHN datasets for $(m, \ell) = (20n, 1)$, where $n$ is the dimension of the embedding.
Figure 5: The distribution of scaling distance to points in $\Phi(V)$ (in blue). The sample $V$ is of scaling distance 1 (left column) and 8 (right column). The histograms of scaled Euclidean distances to the Voronoi center of the class are in green (the set of vertices $V$) and in red ($\Phi(V)$). Notice that both networks are strongly contractive.
Figure 6: Results of the anomaly detection experiment on embedded MNIST dataset. Each curve represents performance of RPDs for one class (mean and joint variance for 3 separately trained networks and 5 RPD constructions). On vertical axis is the AUC score; on horizontal axis the dimensionality of embeddings.
Figure 7: Results of the anomaly detection experiment on embedded FMNIST dataset. Each curve represents performance of RPDs for one class (mean and joint variance for 5 separately trained networks and 5 RPD constructions). On vertical axis is the AUC score; on horizontal axis the dimensionality of embeddings.
Figure 8: Results of the anomaly detection experiment on embedded MNIST dataset. Each curve represents performance of RPDs for one class (mean and joint variance for 5 separately trained networks and 5 RPD constructions). On vertical axis is the AUC score; on horizontal axis the number of hyperplanes used to define RPD.
Figure 9: Results of the anomaly detection experiment on embedded FMNIST dataset. Each curve represents performance of RPDs for one class (mean and joint variance for 3 separately trained networks and 5 RPD constructions). On vertical axis is the AUC score; on horizontal axis the number of hyperplanes used to define RPD.