Geometric Disentanglement by Random Convex Polytopes

Marek Kaluba
Karlsruher Institut für Technologie, Karlsruhe, Germany

joint work with

Michael Joswig Technische Universität Berlin, Chair of Discrete Mathematics/Geometry & MPI for Mathematics in the Sciences

Lukas Ruffs Technische Universität Berlin, Department of Electrical Engineering and Computer Science, Machine Learning Group

June 2021

What is this talk about?

- In ML **convexity** is a common (and often implicit) assumption (e.g. SVM)

What is this talk about?

- In ML **convexity** is a common (and often implicit) assumption (e.g. SVM)
- Generalization property of neural network corresponds to convexity in the feature space

What is this talk about?

- In ML **convexity** is a common (and often implicit) assumption (e.g. SVM)
- Generalization property of neural network corresponds to convexity in the feature space
- Measuring convexity is hard

What is this talk about?

- In ML **convexity** is a common (and often implicit) assumption (e.g. SVM)
- Generalization property of neural network corresponds to convexity in the feature space
- Measuring convexity is hard
- We propose **random polytope descriptor** (RPD) as a relaxation of the convex hull which is easy to compute and robust with respect to outliers.

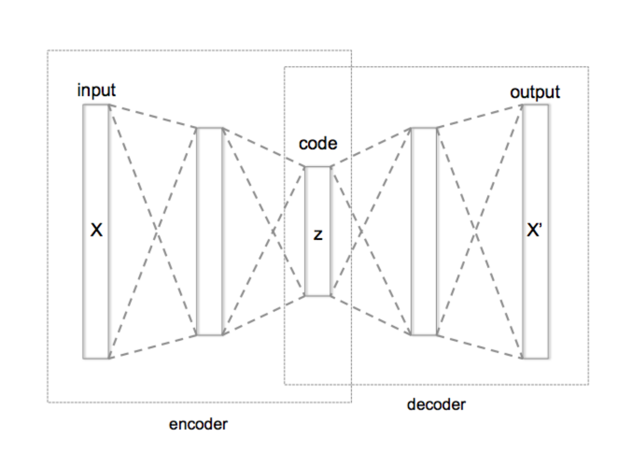
What is this talk about?

- In ML **convexity** is a common (and often implicit) assumption (e.g. SVM)
- Generalization property of neural network corresponds to convexity in the feature space
- Measuring convexity is hard
- We propose **random polytope descriptor** (RPD) as a relaxation of the convex hull which is easy to compute and robust with respect to outliers.
- We evaluate the convexity of **autoencoded data** to assess networks generalization and robustness to out-of-distribution attacks.

Autoencoders (AE)

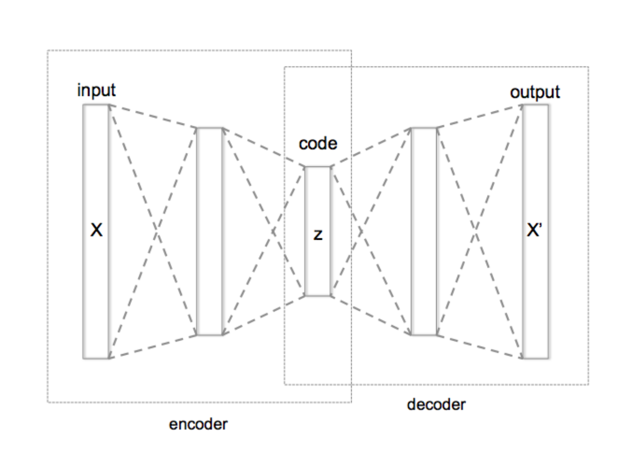
Autoencoders (AE)

- Feature "auto-selection": forcing neural network to go through a bottleneck (i.e. compress/encode the input)



Autoencoders (AE)

- Feature "auto-selection": forcing neural network to go through a bottleneck (i.e. compress/encode the input)



- What is our objective function?

Autoencoders (AE)

- Let $\Phi\colon \mathbb{R}^N \to \mathbb{R}^n$ and $\Psi\colon \mathbb{R}^n \to \mathbb{R}^N$ be a pair (encoder, decoder) of NN.

Autoencoders (AE)

- Let $\Phi\colon \mathbb{R}^N \to \mathbb{R}^n$ and $\Psi\colon \mathbb{R}^n \to \mathbb{R}^N$ be a pair (encoder, decoder) of NN.
- The standard objective is to minimize *reconstruction error*

$$\sum_x \|\Psi(\Phi(x)) - x\|$$

Autoencoders (AE)

- Let $\Phi\colon \mathbb{R}^N \to \mathbb{R}^n$ and $\Psi\colon \mathbb{R}^n \to \mathbb{R}^N$ be a pair (encoder, decoder) of NN.
- The standard objective is to minimize *reconstruction error*

$$\sum_x \|\Psi(\Phi(x)) - x\|$$

- Variational objective is to minimize a function based on what happens in the latent space, **regularized** by *reconstruction error*, e.g.

$$\underbrace{\sum_x |\|\Phi(x)\| - 1|}_{\text{objective}} + \underbrace{\sum_x \|\Psi(\Phi(x)) - x\|}_{\text{regularizer}}$$

Autoencoders (AE)

- Let $\Phi\colon \mathbb{R}^N \to \mathbb{R}^n$ and $\Psi\colon \mathbb{R}^n \to \mathbb{R}^N$ be a pair (encoder, decoder) of NN.
- The standard objective is to minimize *reconstruction error*

$$\sum_x \|\Psi(\Phi(x)) - x\|$$

- Variational objective is to minimize a function based on what happens in the latent space, **regularized** by *reconstruction error*, e.g.

$$\underbrace{\sum_x |\|\Phi(x)\| - 1|}_{\text{objective}} + \underbrace{\sum_x \|\Psi(\Phi(x)) - x\|}_{\text{regularizer}}$$

- Network with such an objective is called **Variational autoencoder** (VAE).

Where are the polytopes?

- What are the features learned by an encoder?

Where are the polytopes?

- What are the features learned by an encoder?
- What happened in the latent space to our data?

Where are the polytopes?

- What are the features learned by an encoder?
- What happened in the latent space to our data?
- Are natural clusters in the data well preserved in the latent space?

Where are the polytopes?

- What are the features learned by an encoder?
- What happened in the latent space to our data?
- Are natural clusters in the data well preserved in the latent space?
- Is the learned representation of the data robust?

Solution: assess the convexity.

Where are the polytopes?

- What are the features learned by an encoder?
- What happened in the latent space to our data?
- Are natural clusters in the data well preserved in the latent space?
- Is the learned representation of the data robust?

Solution: assess the convexity.

- Create the convex-hull of the points from a given class **in the latent space**.

Where are the polytopes?

- What are the features learned by an encoder?
- What happened in the latent space to our data?
- Are natural clusters in the data well preserved in the latent space?
- Is the learned representation of the data robust?

Solution: assess the convexity.

- Create the convex-hull of the points from a given class **in the latent space**.
- Use the proximity to the convex hull to **explain the networks decisions**.

Where are the polytopes?

- What are the features learned by an encoder?
- What happened in the latent space to our data?
- Are natural clusters in the data well preserved in the latent space?
- Is the learned representation of the data robust?

Solution: assess the convexity.

- Create the convex-hull of the points from a given class **in the latent space**.
- Use the proximity to the convex hull to **explain the networks decisions**.
- Compute the intersections of convex hulls to quantify the **entanglement** of encoded classes.

De-idealizing the setup: take one

- Convex hull computations are infeasible in reasonable dimensions

De-idealizing the setup: take one

- Convex hull computations are infeasible in reasonable dimensions
- Computing distance to a polytope is costly

De-idealizing the setup: take one

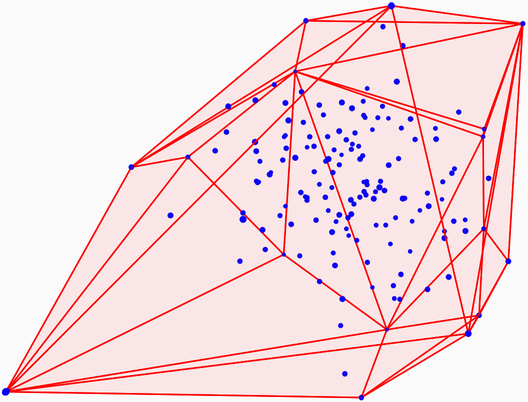
- Convex hull computations are infeasible in reasonable dimensions
- Computing distance to a polytope is costly

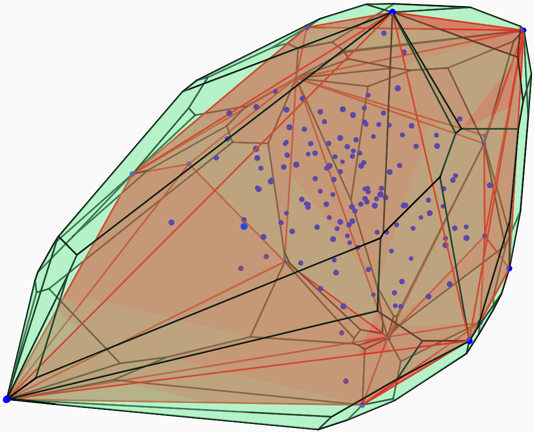
Solution:

Definition

The **dual bounding body** of X with respect to (a set of directions) Y is the polyhedron

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





De-idealizing the setup: take two

- The input is inprecise and often noisy (soft boundaries) while polytopes are very rigid

De-idealizing the setup: take two

- The input is inprecise and often noisy (soft boundaries) while polytopes are very rigid

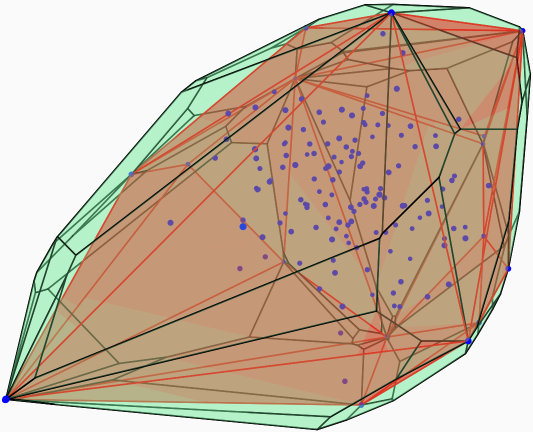
Definition

Let $\ell \in [0,1]$. The **random polytope descriptor** of X with respect to Y (=a set of m directions chosen uniformly at random) is the polyhedron

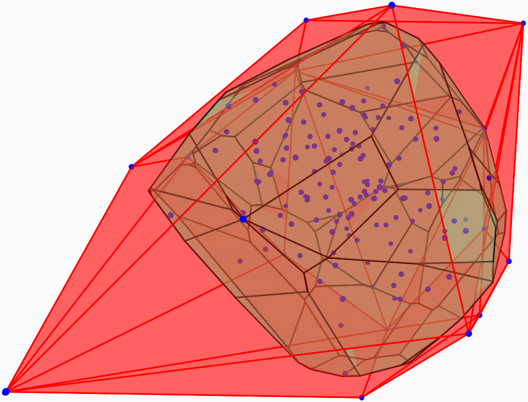
$$\mathrm{RPD}_{m,\ell}(X) := \left\{ v \in \mathbb{R}^d \;\middle|\; \langle v, y\rangle \leq \mu_{\ell,y} \sup_{x \in X} \{\langle x, y\rangle\},\ y \in Y \right\}$$

where $\mu_{\ell,y}\,\mathrm{sup}$ denotes the ℓ-th percentile of probability measure μ on X projected onto direction y.

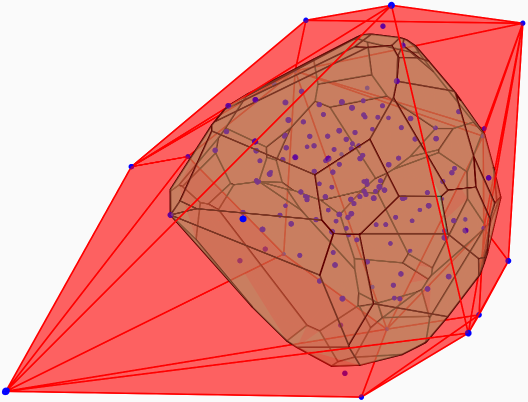
Random Polytope Descriptor



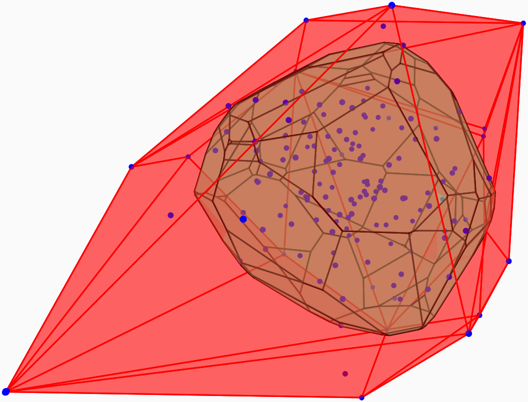
Random Polytope Descriptor



Random Polytope Descriptor



Random Polytope Descriptor



Experimental Results

Disentanglement/convexity

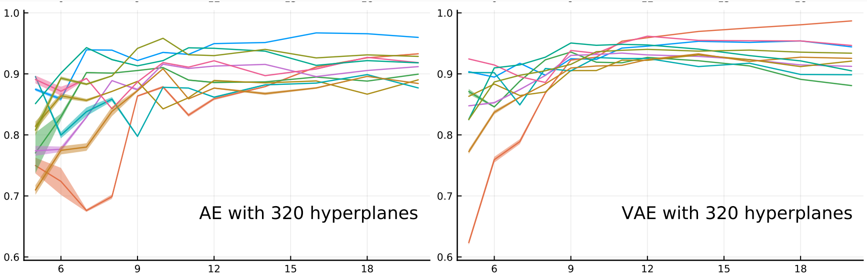
- How tight and convex are the clusters encoded by (variational) autoencoders?

Disentanglement/convexity

- How tight and convex are the clusters encoded by (variational) autoencoders?
- We trained autoencoder networks to embed the MNIST dataset in different dimensions.

Disentanglement/convexity

- How tight and convex are the clusters encoded by (variational) autoencoders?
- We trained autoencoder networks to embed the MNIST dataset in different dimensions.
- Assess the convexity/disentanglement of clusers by measuring the performance of RPDs as classifiers.

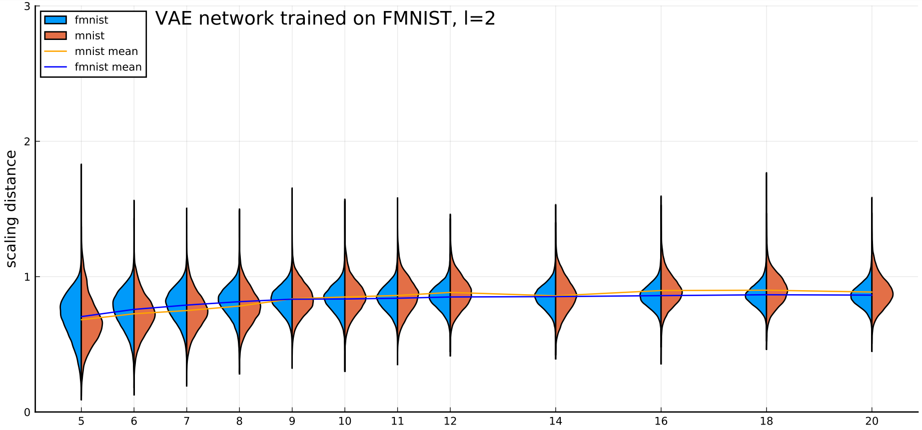


Out of distribution attack

- check how well a neural network recognizes out-of-distribution samples.

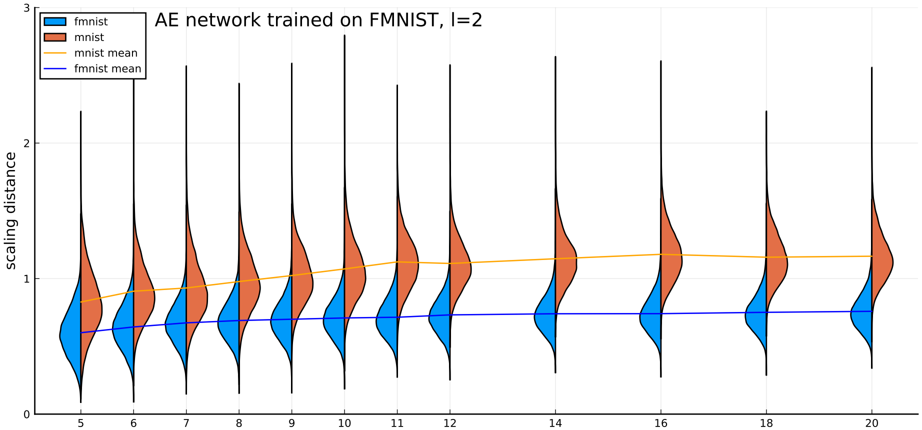
Out of distribution attack

- check how well a neural network recognizes out-of-distribution samples.
- network trained on FMNIST (more complex) is fed MNIST (less complex)



Out of distribution attack

- check how well a neural network recognizes out-of-distribution samples.
- network trained on FMNIST (more complex) is fed MNIST (less complex)



Recap

- Generalization corresponds to convexity in the latent space.
- Random Polytope Descriptor is a computable and flexible relaxation of the convex hull.
- Convexity in the latent space can be assessed by the performance of RPD and **scaling distance** as classifier.
- RPD can be used to evaluate robustness of a NN to out-of-distribution attacks.

Recap

- Generalization corresponds to convexity in the latent space.
- Random Polytope Descriptor is a computable and flexible relaxation of the convex hull.
- Convexity in the latent space can be assessed by the performance of RPD and **scaling distance** as classifier.
- RPD can be used to evaluate robustness of a NN to out-of-distribution attacks.

For further information see `https://arxiv.org/abs/2009.13987`