Here is a cool paper!

Albert Gu, Frederic Sala, Beliz Gunel & Christopher Re
Stanford University

Siba Smarak Panigrahi (SSP)
McGill University and Mila

Oct 21, 2022

Learning Mixed-Curvature Representations in Products of Model Spaces

Albert Gu, Frederic Sala, Beliz Gunel & Christopher Re
Stanford University

Siba Smarak Panigrahi (SSP)
McGill University and Mila

Oct 21, 2022

Outline

- Season 1: Pilot
  - Why do we need this paper?
  - What do I additionally need to know?

Outline

- Season 1: Pilot
    - Why do we need this paper?
    - What do I additionally need to know?
- Season 2: Into the Product Spaces
    - How to construct product spaces?
    - How to optimize on product spaces while learning curvatures?
    - How to estimate signature?

Outline

- Season 1: Pilot
    - Why do we need this paper?
    - What do I additionally need to know?
- Season 2: Into the Product Spaces
    - How to construct product spaces?
    - How to optimize on product spaces while learning curvatures?
    - How to estimate signature?
- Season 3: Swan Song
    - What experiments do they do?
    - What are we concluding then? (with a couple of personal thoughts)

Season 1: Pilot

Why do we need this paper?

- **Idea**: Quality of representations is better if geometry of embedding space and data match

Why do we need this paper?

- **Idea**: Quality of representations is better if geometry of embedding space and data match
  - Spherical spaces are wonderful for cyclical structure
  - Hyperbolic spaces for hierarchical structure
  - Most data is not structured (this uniformly)!

Why do we need this paper?

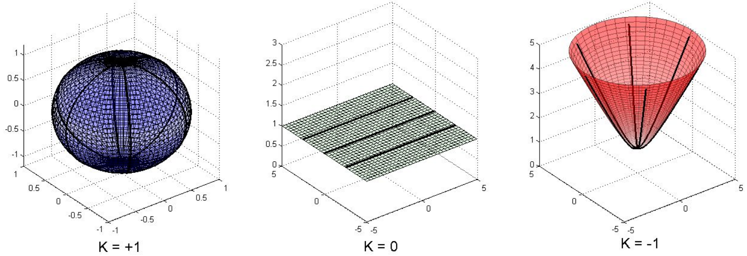
- **Idea**: Quality of representations is better if geometry of embedding space and data match
  - Spherical spaces are wonderful for cyclical structure
  - Hyperbolic spaces for hierarchical structure
  - Most data is not structured (this uniformly)!
  - To better capture the geometry of data:
    - Product of constant curvature spaces to obtain range of (non-constant) curvatures
    - Even better, we can learn the curvature and embedding simultaneously!

Why do we need this paper?

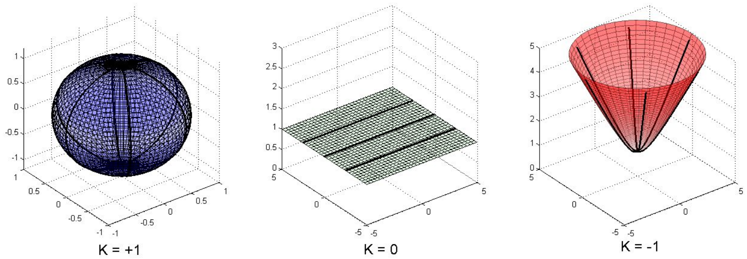
- **Idea**: Quality of representations is better if geometry of embedding space and data match
  - Spherical spaces are wonderful for cyclical structure
  - Hyperbolic spaces for hierarchical structure
  - Most data is not structured (this uniformly)!

  - To better capture the geometry of data:
    - Product of constant curvature spaces to obtain range of (non-constant) curvatures
    - Even better, we can learn the curvature and embedding simultaneously!

- Model spaces of constant curvature include:
    - Euclidean (curvature = 0)
    - Hyperbolic (curvature = negative)
    - Spherical (curvature = positive)

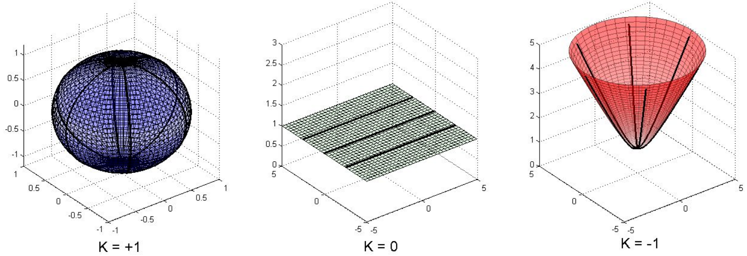


- Model spaces of constant curvature include:
    - Euclidean (curvature = 0)
    - Hyperbolic (curvature = negative)
    - Spherical (curvature = positive)



- Authors also provide a way to compute
    - Means on these mixed spaces (and provide its computational complexity)
        - $T = \{p_1, p_2, \cdots, p_n\}$ in manifold M (dimension r), mean is $\mu(T) := argmin_p \sum_i (d_M^2(p, pi))$
        - Essential for downstream applications like analogy tasks with word embeddings, for clustering, and for centering before applying PCA.

- Model spaces of constant curvature include:
    - Euclidean (curvature = 0)
    - Hyperbolic (curvature = negative)
    - Spherical (curvature = positive)



- Authors also provide a way to compute
    - Means on these mixed spaces (and provide its computational complexity)
        - $T = \{p_1, p_2, \cdots, p_n\}$ in manifold M (dimension r), mean is $\mu(T) := argmin_p \sum_i (d_M^2(p, pi))$
        - Essential for downstream applications like analogy tasks with word embeddings, for clustering, and for centering before applying PCA.

Lemma 2. *Let $\mathcal{P}$ be a product of model spaces of total dimension r, $T = \{p_1, \ldots, p_n\}$ points in $\mathcal{P}$ and $w_1, \ldots, w_n$ weights satisfying $w_i \geq 0$ and $\sum_{i=1}^{n} w_i = 1$. Moreover, let the components of the points in $\mathcal{P}$, $p_{i|\mathbb{S}^j}$ restricted to each spherical component space $\mathbb{S}^j$ fall in one hemisphere of $\mathbb{S}^j$. Then, Riemannian gradient descent recovers the mean $\mu(T)$ within distance ϵ in time $O(nr \log \epsilon^{-1})$.*

What do I additionally need to know?

- **Embeddings, distortion, and mAP**:
  - d_U, d_V are distances for metric spaces U and V
  - *f: U -> V* is an embedding from U to V
  - Distortion: between a pair of points *a, b* is defined as

$$\frac{|d_V(f(a),f(b)) - d_U(a,b)|}{d_U(a,b)}$$

    - Average over all pairs of points := average distortion (D_{avg}) (lower is better)

What do I additionally need to know?

- **Embeddings, distortion, and mAP**:
  - d_U, d_V are distances for metric spaces U and V
  - *f: U -> V* is an embedding from U to V
  - Distortion: between a pair of points *a, b* is defined as

$$\frac{|d_V(f(a),f(b)) - d_U(a,b)|}{d_U(a,b)}$$

    - Average over all pairs of points := average distortion (D_{avg}) (lower is better)
  - Mean Average Precision (mAP): for unweighted graph G = (V, E) (higher is better)

$$\mathrm{mAP}(f) = \frac{1}{|V|} \sum_{a \in V} \frac{1}{\deg(a)} \sum_{i=1}^{|\mathcal{N}_a|} |\mathcal{N}_a \cap R_{a,b_i}| / |R_{a,b_i}|$$

N_a = neighborhood of *a* = $\{b_1, \ldots, b_{\deg(a)}\}$

R_{a,b_i} = smallest set of nearest points required to retrieve i-th neighbor of a in f

- Product Manifolds
  - Smooth manifolds: $M_1, M_2, \ldots, M_K$; M = (cartesian) product manifold = $M_1 \times M_2 \times \ldots \times M_K$
  - Point p in M is represented by $p = (p_1, \ldots, p_k) : p_i \in M_i$;
  - Similarly, $v \in T_pM$ can be written $(v_1, \ldots, v_k) : v_i \in T_{p_i}M_i$
  - If g_i is metric associated with M_i then M is also Riemannian with metric g

$$g(u, v) = \sum_{i=1}^{k} g_i(u_i, v_i)$$

i.e., product metric decomposes into the sum of the constituent metrics

- Product Manifolds
  - Smooth manifolds: $M_1, M_2, \ldots, M_K$; M = (cartesian) product manifold = $M_1 \times M_2 \times \ldots \times M_K$
  - Point p in M is represented by $p = (p_1, \ldots, p_k) : p_i \in M_i$;
  - Similarly, $v \in T_pM$ can be written $(v_1, \ldots, v_k) : v_i \in T_{p_i}M_i$
  - If g_i is metric associated with M_i then M is also Riemannian with metric g

$$g(u, v) = \sum_{i=1}^{k} g_i(u_i, v_i)$$

i.e., product metric decomposes into the sum of the constituent metrics

- Distances
  - Key idea: Optimization (taking a step) on manifold can be performed in tangent space and transferred to manifold through exponential map, i.e., $\mathrm{Exp}_p : T_pM \to M$
  - Interestingly, exponential map and squared distances decompose in product space

$$\mathrm{Exp}_p(v) = (\mathrm{Exp}_{p_1}(v_1), \ldots, \mathrm{Exp}_{p_k}(v_k)), \qquad d_{\mathcal{P}}^2(x, y) = \sum_{i=1}^{k} d_i^2(x_i, y_i)$$

i.e., shortest path between points in product space is shortest path traveled in each component

- Hyperbolic Model
  - For hyperboloid $\mathbb{H}_K^d$, points in $\mathbb{R}^{d+1}$ such that $\{p \in \mathbb{R}^{d+1} : \|p\|_* = -K^{1/2}, p_0 > 0\}$
  - Minkowski product is defined as $\langle p, q\rangle_* := p^T J q = -p_0 q_0 + p_1 q_1 + \ldots + p_d q_d$
    - norm as $\|p\|_* = \langle p, p\rangle_*^{\frac{1}{2}}$
    - hyperbolic distance $\mathbb{H}^d$ is $d_H(p, q) = \operatorname{acosh}(-\langle p, q\rangle_*)$
    - Also note that $J = \begin{bmatrix} -1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \in \mathrm{R}^{(d+1)x(d+1)}$

- Hyperbolic Model
  - For hyperboloid $\mathbb{H}^d_K$, points in $\mathbb{R}^{d+1}$ such that $\{p \in \mathbb{R}^{d+1} : \|p\|_* = -K^{1/2}, p_0 > 0\}$
  - Minkowski product is defined as $\langle p, q \rangle_* := p^T J q = -p_0 q_0 + p_1 q_1 + \ldots + p_d q_d$
    - norm as $\|p\|_* = \langle p, p \rangle_*^{\frac{1}{2}}$
    - hyperbolic distance $\mathbb{H}^d$ is $d_H(p, q) = \operatorname{acosh}(-\langle p, q \rangle_*)$
    - Also note that $J = \begin{bmatrix} -1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \in \mathrm{R}^{(d+1)x(d+1)}$

- Spherical Model
  - Similar to hyperbolic model, differences being
    - $\mathbb{S}^d_K = \{p \in \mathbb{R}^{d+1} : \|p\|_2 = K^{1/2}\}$
    - Metric is same as Euclidean metric, thus, $J = \begin{bmatrix} 1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \in \mathrm{R}^{(d+1)x(d+1)}$
    - Spherical distance $d_S(p, q) = \arccos(\langle p, q \rangle)$

Season 2: Into the Product Spaces

How to construct product spaces?

- Consider product space as $\mathcal{P} = \mathbb{S}^{s_1} \times \mathbb{S}^{s_2} \times \cdots \times \mathbb{S}^{s_m} \times \mathbb{H}^{h_1} \times \mathbb{H}^{h_2} \times \cdots \times \mathbb{H}^{h_n} \times \mathbb{E}^{e}$
  - s_i, h_i and e are dimensions of spherical, hyperbolic, and euclidean spaces

How to construct product spaces?

- Consider product space as $\mathcal{P} = \mathbb{S}^{s_1} \times \mathbb{S}^{s_2} \times \cdots \times \mathbb{S}^{s_m} \times \mathbb{H}^{h_1} \times \mathbb{H}^{h_2} \times \cdots \times \mathbb{H}^{h_n} \times \mathbb{E}^{e}$
  - s_i, h_j and e are dimensions of spherical, hyperbolic, and euclidean spaces
  - has m+n+1 components, with total dimension being $\sum_i s_i + \sum_j h_j + e$
  - signature is the number of components of each type and their dimensions

How to construct product spaces?

- Consider product space as $\mathcal{P} = \mathbb{S}^{s_1} \times \mathbb{S}^{s_2} \times \cdots \times \mathbb{S}^{s_m} \times \mathbb{H}^{h_1} \times \mathbb{H}^{h_2} \times \cdots \times \mathbb{H}^{h_n} \times \mathbb{E}^{e}$
  - s_i, h_i and e are dimensions of spherical, hyperbolic, and euclidean spaces
  - has m+n+1 components, with total dimension being $\sum_i s_i + \sum_j h_j + e$
  - signature is the number of components of each type and their dimensions
    - we can thus say signature is (1, spherical, s_1), (2, spherical, s_2), ...
    - note, there is no such signature (i.e., there is only one euclidean component if exists in product), since product of $\mathbb{E}^{r_1}, \ldots, \mathbb{E}^{r_n}$ is equal to single space $\mathbb{E}^{r_1 + \ldots + r_n}$

How to construct product spaces?

- Consider product space as $\mathcal{P} = \mathbb{S}^{s_1} \times \mathbb{S}^{s_2} \times \cdots \times \mathbb{S}^{s_m} \times \mathbb{H}^{h_1} \times \mathbb{H}^{h_2} \times \cdots \times \mathbb{H}^{h_n} \times \mathbb{E}^{e}$
  - s_i, h_i and e are dimensions of spherical, hyperbolic, and euclidean spaces
  - has m+n+1 components, with total dimension being $\sum_i s_i + \sum_j h_j + e$
  - signature is the number of components of each type and their dimensions
    - we can thus say signature is (1, spherical, s_1), (2, spherical, s_2), ...
    - note, there is no such signature (i.e., there is only one euclidean component if exists in product), since product of $\mathbb{E}^{r_1}, \ldots, \mathbb{E}^{r_n}$ is equal to single space $\mathbb{E}^{r_1 + \ldots + r_n}$
- Estimating signature
  - (empirical) discrete curvature of given data
  - (theoretical) sectional curvature distribution
  - matching them both to obtain signature

How to optimize on product spaces while learning curvatures?

- Suppose we already have a signature
  - To obtain the embeddings we minimize a loss (basically a stable form of distortion)

$$\mathcal{L}(x) = \sum_{1 \le i \le j \le n} \left| \left(\frac{d_{\mathcal{P}}(x_i, x_j)}{d_G(X_i, X_j)} \right)^2 - 1 \right|$$

How to optimize on product spaces while learning curvatures?

- Suppose we already have a signature
  - To obtain the embeddings we minimize a loss (basically a stable form of distortion)

$$\mathcal{L}(x) = \sum_{1 \le i \le j \le n} \left| \left(\frac{d_{\mathcal{P}}(x_i, x_j)}{d_G(X_i, X_j)} \right)^2 - 1 \right|$$

Algorithm 1 R-SGD in products

Input: Loss function $L : \mathcal{P} \to \mathbb{R}$
Initialize $x^{(0)} \in \mathcal{P}$ randomly
for $t = 0, \ldots, T-1$ **do**
  $h \leftarrow \nabla L(x^{(t)})$
  for $i = 1, \ldots, m$ **do**
    $v_i \leftarrow \text{proj}^S_{x_i^{(t)}}(h_i)$
  for $i = m+1, \ldots, m+n$ **do**
    $v_i \leftarrow \text{proj}^H_{x_i^{(t)}}(h_i)$
    $v_i \leftarrow J v_i$
  $v_{m+n+1} \leftarrow h_{m+n+1}$
  for $i = 1, \ldots, m+n+1$ **do**
    $x_i^{(t+1)} \leftarrow \text{Exp}_{x_i^{(t)}}(v_i)$
return $x^{(T)}$

- Compute Euclidean gradient
  - w.r.t to ambient space of the embedding

How to optimize on product spaces while learning curvatures?

- Suppose we already have a signature
  - To obtain the embeddings we minimize a loss (basically a stable form of distortion)

$$\mathcal{L}(x) = \sum_{1 \le i \le j \le n} \left| \left(\frac{d_{\mathcal{P}}(x_i, x_j)}{d_G(X_i, X_j)} \right)^2 - 1 \right|$$

Algorithm 1 R-SGD in products

Input: Loss function $L : \mathcal{P} \to \mathbb{R}$
Initialize $x^{(0)} \in \mathcal{P}$ randomly
for $t = 0, \ldots, T-1$ **do**
  $h \leftarrow \nabla L(x^{(t)})$
  for $i = 1, \ldots, m$ **do**
    $v_i \leftarrow \text{proj}^{S}_{x_i^{(t)}}(h_i)$
  for $i = m+1, \ldots, m+n$ **do**
    $v_i \leftarrow \text{proj}^{H}_{x_i^{(t)}}(h_i)$
    $v_i \leftarrow J v_i$
  $v_{m+n+1} \leftarrow h_{m+n+1}$
  for $i = 1, \ldots, m+n+1$ **do**
    $x_i^{(t+1)} \leftarrow \text{Exp}_{x_i^{(t)}}(v_i)$
return $x^{(T)}$

- Compute Euclidean gradient
  - w.r.t to ambient space of the embedding
- Hyperboloid and spherical models have lower dimensions than the embedding space
  - Project gradient vector *h* into tangent spaces

How to optimize on product spaces while learning curvatures?

- Suppose we already have a signature
  - To obtain the embeddings we minimize a loss (basically a stable form of distortion)

$$\mathcal{L}(x) = \sum_{1 \le i \le j \le n} \left| \left(\frac{d_{\mathcal{P}}(x_i, x_j)}{d_G(X_i, X_j)} \right)^2 - 1 \right|$$

Algorithm 1 R-SGD in products

Input: Loss function $L : \mathcal{P} \to \mathbb{R}$
Initialize $x^{(0)} \in \mathcal{P}$ randomly
for $t = 0, \ldots, T-1$ **do**
    $h \leftarrow \nabla L(x^{(t)})$
    for $i = 1, \ldots, m$ **do**
        $v_i \leftarrow \mathrm{proj}^S_{x_i^{(t)}}(h_i)$
    for $i = m+1, \ldots, m+n$ **do**
        $v_i \leftarrow \mathrm{proj}^H_{x_i^{(t)}}(h_i)$
        $v_i \leftarrow J v_i$
    $v_{m+n+1} \leftarrow h_{m+n+1}$
    for $i = 1, \ldots, m+n+1$ **do**
        $x_i^{(t+1)} \leftarrow \mathrm{Exp}_{x_i^{(t)}}(v_i)$
return $x^{(T)}$

- Compute Euclidean gradient
  - w.r.t to ambient space of the embedding
- Hyperboloid and spherical models have lower dimensions than the embedding space
  - Project gradient vector *h* into tangent spaces
  - Take the step with these vectors in tangent space and Exponential map

How to optimize on product spaces while learning curvatures?

- Suppose we already have a signature
  - To obtain the embeddings we minimize a loss (basically a stable form of distortion)

$$\mathcal{L}(x) = \sum_{1 \le i \le j \le n} \left| \left(\frac{d_{\mathcal{P}}(x_i, x_j)}{d_G(X_i, X_j)} \right)^2 - 1 \right|$$

Algorithm 1 R-SGD in products

```
Input: Loss function L : P → ℝ
Initialize x^(0) ∈ P randomly
for t = 0, ..., T − 1 do
    h ← ∇L(x^(t))
    for i = 1, ..., m do
        v_i ← proj^S_{x_i^(t)}(h_i)
    for i = m + 1, ..., m + n do
        v_i ← proj^H_{x_i^(t)}(h_i)
        v_i ← J v_i
    v_{m+n+1} ← h_{m+n+1}
    for i = 1, ..., m + n + 1 do
        x_i^(t+1) ← Exp_{x_i^(t)}(v_i)
return x^(T)
```

- Compute Euclidean gradient
  - w.r.t to ambient space of the embedding
- Hyperboloid and spherical models have lower dimensions than the embedding space
  - Project gradient vector h into tangent spaces
  - Take the step with these vectors in tangent space and Exponential map
  - Since g_P of product space decomposes we can carry these steps independently in each component

$$\mathrm{proj}_x^S(h) = h - \langle h, x \rangle x \qquad \mathrm{proj}_x^H(h) = h + \langle h, x \rangle_* x$$

How do we simultaneously learn the curvature?

- First note that, for all values of K, there exists a hyperbolic (K < 0) or a spherical (K > 0) model
- Further, note that we can emulate all curvature (K) values on the corresponding standard models
  - For instance, given points p, q on $\mathbb{S}_{1/R^2}$ of radius R, then

$$d(p,q) \;=\; R \cdot d_{\mathbb{S}_1}(p/R, q/R)$$

  - Hence, we can work with models of curvature 1 rather than K.

How do we simultaneously learn the curvature?

- First note that, for all values of K, there exists a hyperbolic (K < 0) or a spherical (K > 0) model
- Further, note that we can emulate all curvature (K) values on the corresponding standard models
  - For instance, given points p, q on $\mathbb{S}_{1/R^2}$ of radius R, then

$$d(p,q) \;=\; R \cdot d_{\mathbb{S}_1}(p/R, q/R)$$

  - Hence, we can work with models of curvature 1 rather than K.
- Note, the loss depends only on squared distances (in turn is sum of distances in components).
  - We can consider R as parameter and optimize for the curvature!

How to estimate the signature?

- As stated earlier, we will
  - find the sectional curvature distribution and
  - empirically estimate the discrete curvature from data
  - to match them and estimate signature

How to estimate the signature?

- As stated earlier, we will
  - find the sectional curvature distribution and
  - empirically estimate the discrete curvature from data
  - to match them and estimate signature
- **Sectional curvature**: linearly independent x, y in tangent space T_pM, spanning a two-dimensional subspace U, the sectional curvature $K_p(x, y)$ is defined as the Gaussian curvature of the surface Exp(U).

$$K(x,y) := \frac{(x,y,x,y)}{\|x\|^2\|y\|^2 - \langle x,y\rangle^2}$$

How to estimate the signature?

- As stated earlier, we will
  - find the sectional curvature distribution and
  - empirically estimate the discrete curvature from data
  - to match them and estimate signature
- **Sectional curvature**: linearly independent x, y in tangent space T_pM, spanning a two-dimensional subspace U, the sectional curvature $K_p(x, y)$ is defined as the Gaussian curvature of the surface Exp(U).

$$K(x, y) := \frac{(x, y, x, y)}{\|x\|^2\|y\|^2 - \langle x, y\rangle^2}$$

Lemma 1. *Let $M = M_1 \times M_2$ where M_i has constant curvature K_i. For any $u, v \in T_pM$, K_1, K_2 are both non-negative, the sectional curvature satisfies $K(u, v) \in [0, \max\{K_1, K_2\}]$. K_1, K_2 are both non-positive, the sectional curvature satisfies $K(u, v) \in [\min\{K_1, K_2\}, 0]$. $K_i < 0$ and $K_j > 0$ for $i \neq j$, then $K(u, v) \in [K_i, K_j]$.*

The derivation of this lemma enables us to obtain a distribution of sectional curvature!

$$K((x_1, x_2), (y_1, y_2)) = \frac{\alpha_1 K_1}{\alpha_1 + \alpha_2 + \beta} + \frac{\alpha_2 K_2}{\alpha_1 + \alpha_2 + \beta}$$

$\alpha_i = \|x_i\|^2 \|y_i\|^2 - \langle x_i, y_i \rangle^2 \text{ for } i = 1, 2$ and $\beta = \|x_1\|^2 \|y_2\|^2 + \|x_2\|^2 \|y_1\|^2$

W.L.O.G and using Cauchy-Schwarz inequality, we get $0 \leq K((x_1, x_2), (y_1, y_2)) \leq \frac{\alpha_1}{\alpha_1 + \alpha_2} K_1 + \frac{\alpha_2}{\alpha_1 + \alpha_2} K_2.$

$$K((x_1,x_2),(y_1,y_2)) = \frac{\alpha_1 K_1}{\alpha_1+\alpha_2+\beta} + \frac{\alpha_2 K_2}{\alpha_1+\alpha_2+\beta}$$

$\alpha_i = \|x_i\|^2\|y_i\|^2 - \langle x_i, y_i\rangle^2 \text{ for } i = 1,2$ and $\beta = \|x_1\|^2\|y_2\|^2 + \|x_2\|^2\|y_1\|^2$

W.L.O.G and using Cauchy-Schwarz inequality, we get $0 \le K((x_1,x_2),(y_1,y_2)) \le \frac{\alpha_1}{\alpha_1+\alpha_2}K_1 + \frac{\alpha_2}{\alpha_1+\alpha_2}K_2.$

Algorithm 2 Sectional curvature distribution

1: **Input: Dimensions** d_1, d_2
2: $a_1 \leftarrow \chi^2(d_1-1)$
3: $b_1 \leftarrow \chi^2(d_1-1)$
4: $t_1 \leftarrow \mathrm{Beta}((d_1-1)/2,(d_1-1)/2)$
5: $c_1 \leftarrow a_1^{1/2} b_1^{1/2}(2t_1-1)$
6: $a_2 \leftarrow \chi^2(d_2-1)$
7: $b_2 \leftarrow \chi^2(d_2-1)$
8: $t_2 \leftarrow \mathrm{Beta}((d_2-1)/2,(d_2-1)/2)$
9: $c_2 \leftarrow a_2^{1/2} b_2^{1/2}(2t_2-1)$
10: $\alpha_1 \leftarrow a_1 b_1 - c_1^2$
11: $\alpha_2 \leftarrow a_2 b_2 - c_2^2$
12: $\beta \leftarrow a_1 b_2 + a_2 b_1$
13: **return** $\frac{\alpha_1}{\alpha_1+\alpha_2+\beta}K_1 + \frac{\alpha_2}{\alpha_1+\alpha_2+\beta}K_2$

Use Algorithm 2 to obtain sectional curvature distribution

- note the input to algorithm is dimensions of manifolds
- Also observe that Lemma 1 is for product of 2 model spaces only

How to get the discrete curvature from graph data?

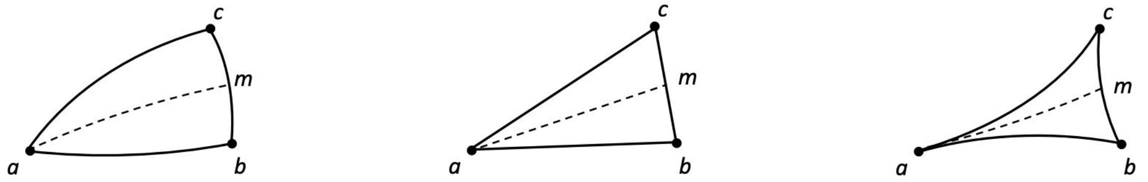


Figure 3: Geodesic triangles in differently curved spaces: compared to Euclidean geometry in which it satisfies the parallelogram law (Center), the median am is longer in cycle-like positively curved space (Left), and shorter in tree-like negatively curved space (Right). The relative length of am can be used as a heuristic to estimate discrete curvature.

How to get the discrete curvature from graph data?

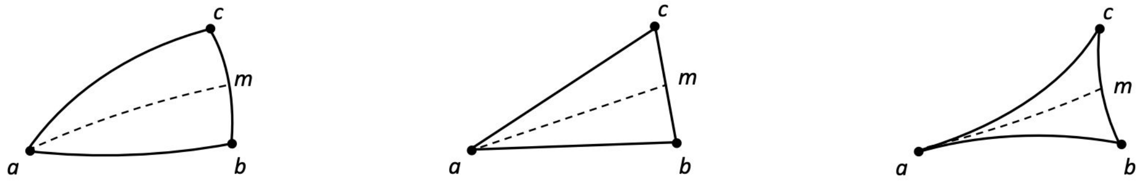


Figure 3: Geodesic triangles in differently curved spaces: compared to Euclidean geometry in which it satisfies the parallelogram law (Center), the median am is longer in cycle-like positively curved space (Left), and shorter in tree-like negatively curved space (Right). The relative length of am can be used as a heuristic to estimate discrete curvature.

Authors use the following equation to estimate the curvature

$$\xi_M(a,b,c) := d_M(a,m)^2 + d_M(b,c)^2/4 - \left(d_M(a,b)^2 + d_M(a,c)^2\right)/2$$

How to get the discrete curvature from graph data?

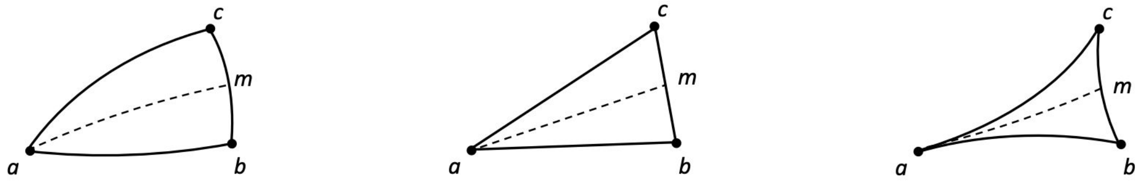


Figure 3: Geodesic triangles in differently curved spaces: compared to Euclidean geometry in which it satisfies the parallelogram law (Center), the median am is longer in cycle-like positively curved space (Left), and shorter in tree-like negatively curved space (Right). The relative length of am can be used as a heuristic to estimate discrete curvature.

Authors use the following equation to estimate the curvature

$$\xi_M(a,b,c) := d_M(a,m)^2 + d_M(b,c)^2/4 - \left(d_M(a,b)^2 + d_M(a,c)^2\right)/2$$

$$\xi_G(m;b,c;a) = \frac{1}{2d_G(a,m)}\xi_G(a,b,c)$$

$$\xi_G(m;b,c) = \frac{1}{|V|-1}\sum_{a\neq m}\xi_G(m;b,c;a)$$

The authors prove three lemmas before they give their algorithm for discrete curvature estimation:

- Lemma 3: their estimation works for lines
- Lemma 4: their estimation technique works for cycles
- Lemma 5: their estimation technique works for trees

The authors prove three lemmas before they give their algorithm for discrete curvature estimation:

- Lemma 3: their estimation works for lines
- Lemma 4: their estimation technique works for cycles
- Lemma 5: their estimation technique works for trees

Lemma 3. *Suppose a lies on the same geodesic line as b, m, c; in other words, WLOG $d_G(a, b) \leq d_G(a, c)$ and suppose $d_G(a, c) = d_G(a, b) + d_G(b, m) + d_G(m, c)$. Then $\xi(m; b, c; a) = 0$.*

Lemma 4. *Consider a cycle graph C with nodes b, m, c such that (m, b) and (m, c) are neighbors. Then for all $a \in C$, $\xi(m; b, c; a)$ is either 0 or positive.*

Lemma 5. *Consider a tree graph T with nodes b, m, c such that (m, b) and (m, c) are neighbors. Then for all $a \in T$, $\xi(m; b, c; a)$ is either 0 or negative.*

The authors prove three lemmas before they give their algorithm for discrete curvature estimation:

- Lemma 3: their estimation works for lines
- Lemma 4: their estimation technique works for cycles
- Lemma 5: their estimation technique works for trees

Lemma 3. *Suppose a lies on the same geodesic line as b, m, c; in other words, WLOG $d_G(a, b) \leq d_G(a, c)$ and suppose $d_G(a, c) = d_G(a, b) + d_G(b, m) + d_G(m, c)$. Then $\xi(m; b, c; a) = 0$.*

Lemma 4. *Consider a cycle graph C with nodes b, m, c such that (m, b) and (m, c) are neighbors. Then for all $a \in C$, $\xi(m; b, c; a)$ is either 0 or positive.*

Lemma 5. *Consider a tree graph T with nodes b, m, c such that (m, b) and (m, c) are neighbors. Then for all $a \in T$, $\xi(m; b, c; a)$ is either 0 or negative.*

Algorithm 3 Empirical estimation of sectional curvature distribution

Input: Graph $G = (V, E)$
$m \leftarrow \text{Uniform}(V)$
$b \leftarrow \text{Uniform}(\mathcal{N}(m))$ $\{\mathcal{N}(v)$ is the neighbor set of $v\}$
$c \leftarrow \text{Uniform}(\mathcal{N}(m))$
$a \leftarrow \text{Uniform}(V)$
$K \leftarrow \xi(m; b, c; a)$
return K

Use moment matching to get K_1 and K_2 i.e.,

- get first and second moments from the above distribution and
- the outputs of algorithm 3 applied to random planes (m,b,c) of graph data

Season 3: Swan Song

What experiments do they do?

	Cycle	**Tree**	**Ring of Trees**
	$\lvert V\rvert = 40, \lvert E\rvert = 40$	$\lvert V\rvert = 40, \lvert E\rvert = 39$	$\lvert V\rvert = 40, \lvert E\rvert = 40$
$(\mathbb{E}^3)^1$	0.1064	0.1483	0.0997
$(\mathbb{H}^3)^1$	0.1638	**0.0321**	0.0774
$(\mathbb{S}^3)^1$	**0.0007**	0.1605	0.1106
$(\mathbb{H}^2)^1 \times (\mathbb{S}^1)^1$	0.1108	0.0538	**0.0616**

Matching Geometries:

- Best distortion values with the geometry of embedding space matching that of data
- Authors consider a fixed total dimension of 3, and obtain results with different signatures

	Cities	CS PhDs		Power		Facebook	
	$\|V\|=312$	$\|V\|=1025, \|E\|=1043$		$\|V\|=4941, \|E\|=6594$		$\|V\|=4039, \|E\|=88234$	
	D_{avg}	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP
$\mathbb{E}^{10}$	0.0735	0.0543	0.8691	0.0917	0.8860	0.0653	0.5801
$\mathbb{H}^{10}$	0.0932	0.0502	0.9310	0.0388	0.8442	0.0596	0.7824
$\mathbb{S}^{10}$	0.0598	0.0569	0.8329	0.0500	0.7952	0.0661	0.5562
$(\mathbb{H}^5)^2$	0.0756	0.0382	0.9628	0.0365	0.8605	0.0430	0.7742
$(\mathbb{S}^5)^2$	**0.0593**	0.0579	0.7940	0.0471	0.8059	0.0658	0.5728
$\mathbb{H}^5 \times \mathbb{S}^5$	0.0622	0.0509	0.9141	**0.0323**	0.8850	**0.0402**	0.7414
$(\mathbb{H}^2)^5$	0.0687	**0.0357**	0.9694	0.0396	0.8739	0.0525	0.7519
$(\mathbb{S}^2)^5$	0.0638	0.0570	0.8334	0.0483	0.8818	0.0631	0.5808
$(\mathbb{H}^2)^2 \times \mathbb{E}^2 \times (\mathbb{S}^2)^2$	0.0765	0.0391	0.8672	0.0380	0.8152	0.0474	0.5951
Best model	$\mathbb{S}^5_{1.0} \times \mathbb{S}^5_{1.1}$	$\mathbb{H}^2_{.3} \times \mathbb{H}^2_{.6} \times \mathbb{H}^2_{1.5} \times (\mathbb{H}^2_{1.2})^2$		$\mathbb{H}^5_{3.4} \times \mathbb{S}^5_{12.6}$		$\mathbb{H}^5_{0.3} \times \mathbb{S}^5_{3.5}$	
D_{avg} **improvement over single space**	0.8%	28.89%		16.75%		32.55%	

	Cities	CS PhDs		Power		Facebook	
	$\lvert V\rvert=312$	$\lvert V\rvert=1025, \lvert E\rvert=1043$		$\lvert V\rvert=4941, \lvert E\rvert=6594$		$\lvert V\rvert=4039, \lvert E\rvert=88234$	
	D_{avg}	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP
$\mathbb{E}^{10}$	0.0735	0.0543	0.8691	0.0917	0.8860	0.0653	0.5801
$\mathbb{H}^{10}$	0.0932	0.0502	0.9310	0.0388	0.8442	0.0596	0.7824
$\mathbb{S}^{10}$	0.0598	0.0569	0.8329	0.0500	0.7952	0.0661	0.5562
$(\mathbb{H}^5)^2$	0.0756	0.0382	0.9628	0.0365	0.8605	0.0430	0.7742
$(\mathbb{S}^5)^2$	**0.0593**	0.0579	0.7940	0.0471	0.8059	0.0658	0.5728
$\mathbb{H}^5 \times \mathbb{S}^5$	0.0622	0.0509	0.9141	**0.0323**	0.8850	**0.0402**	0.7414
$(\mathbb{H}^2)^5$	0.0687	**0.0357**	0.9694	0.0396	0.8739	0.0525	0.7519
$(\mathbb{S}^2)^5$	0.0638	0.0570	0.8334	0.0483	0.8818	0.0631	0.5808
$(\mathbb{H}^2)^2 \times \mathbb{E}^2 \times (\mathbb{S}^2)^2$	0.0765	0.0391	0.8672	0.0380	0.8152	0.0474	0.5951
Best model	$\mathbb{S}^5_{1.0} \times \mathbb{S}^5_{1.1}$	$\mathbb{H}^2_{.3} \times \mathbb{H}^2_{.6} \times \mathbb{H}^2_{1.5} \times (\mathbb{H}^2_{1.2})^2$		$\mathbb{H}^5_{3.4} \times \mathbb{S}^5_{12.6}$		$\mathbb{H}^5_{0.3} \times \mathbb{S}^5_{3.5}$	
D_{avg} **improvement over single space**	0.8%	28.89%		16.75%		32.55%	

- Fix total dimension (d) = 10
- cities graph (intrinsic structure is S^2) embeds well into product with spherical component(s)
- Similarly, PhDs (tree-like structure) in product with hyperbolic component(s)
- Even with data which matches a single constant curvature space, product space does not harm the performance
- In products of identical spaces, the curvatures can be non-uniform, instead of identical

	Cities	CS PhDs		Power		Facebook	
	$\lvert V\rvert=312$	$\lvert V\rvert=1025, \lvert E\rvert=1043$		$\lvert V\rvert=4941, \lvert E\rvert=6594$		$\lvert V\rvert=4039, \lvert E\rvert=88234$	
	D_{avg}	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP
$\mathbb{E}^{10}$	0.0735	0.0543	0.8691	0.0917	0.8860	0.0653	0.5801
$\mathbb{H}^{10}$	0.0932	0.0502	0.9310	0.0388	0.8442	0.0596	0.7824
$\mathbb{S}^{10}$	0.0598	0.0569	0.8329	0.0500	0.7952	0.0661	0.5562
$(\mathbb{H}^5)^2$	0.0756	0.0382	0.9628	0.0365	0.8605	0.0430	0.7742
$(\mathbb{S}^5)^2$	**0.0593**	0.0579	0.7940	0.0471	0.8059	0.0658	0.5728
$\mathbb{H}^5 \times \mathbb{S}^5$	0.0622	0.0509	0.9141	**0.0323**	0.8850	**0.0402**	0.7414
$(\mathbb{H}^2)^5$	0.0687	**0.0357**	0.9694	0.0396	0.8739	0.0525	0.7519
$(\mathbb{S}^2)^5$	0.0638	0.0570	0.8334	0.0483	0.8818	0.0631	0.5808
$(\mathbb{H}^2)^2 \times \mathbb{E}^2 \times (\mathbb{S}^2)^2$	0.0765	0.0391	0.8672	0.0380	0.8152	0.0474	0.5951
Best model	$\mathbb{S}^5_{1.0} \times \mathbb{S}^5_{1.1}$	$\mathbb{H}^2_{.3} \times \mathbb{H}^2_{.6} \times \mathbb{H}^2_{1.5} \times (\mathbb{H}^2_{1.2})^2$		$\mathbb{H}^5_{3.4} \times \mathbb{S}^5_{12.6}$		$\mathbb{H}^5_{0.3} \times \mathbb{S}^5_{3.5}$	
D_{avg} **improvement over single space**	0.8%	28.89%		16.75%		32.55%	

	CS PhDs	Power	Facebook
Estimated Signature	$\mathbb{H}^5_{1.3} \times \mathbb{H}^5_{0.2}$	$\mathbb{H}^5_{1.8} \times \mathbb{S}^5_{1.7}$	$\mathbb{H}^5_{0.9} \times \mathbb{S}^5_{1.6}$

- Fix total dimension (d) = 10
- cities graph (intrinsic structure is S^2) embeds well into product with spherical component(s)
- Similarly, PhDs (tree-like structure) in product with hyperbolic component(s)
- Even with data which matches a single constant curvature space, product space does not harm the performance
- In products of identical spaces, the curvatures can be non-uniform, instead of identical

	Cities	CS PhDs		Power		Facebook	
	$\lvert V\rvert=312$	$\lvert V\rvert=1025, \lvert E\rvert=1043$		$\lvert V\rvert=4941, \lvert E\rvert=6594$		$\lvert V\rvert=4039, \lvert E\rvert=88234$	
	D_{avg}	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP
$\mathbb{E}^{10}$	0.0735	0.0543	0.8691	0.0917	0.8860	0.0653	0.5801
$\mathbb{H}^{10}$	0.0932	0.0502	0.9310	0.0388	0.8442	0.0596	0.7824
$\mathbb{S}^{10}$	0.0598	0.0569	0.8329	0.0500	0.7952	0.0661	0.5562
$(\mathbb{H}^5)^2$	0.0756	0.0382	0.9628	0.0365	0.8605	0.0430	0.7742
$(\mathbb{S}^5)^2$	**0.0593**	0.0579	0.7940	0.0471	0.8059	0.0658	0.5728
$\mathbb{H}^5 \times \mathbb{S}^5$	0.0622	0.0509	0.9141	**0.0323**	0.8850	**0.0402**	0.7414
$(\mathbb{H}^2)^5$	0.0687	**0.0357**	0.9694	0.0396	0.8739	0.0525	0.7519
$(\mathbb{S}^2)^5$	0.0638	0.0570	0.8334	0.0483	0.8818	0.0631	0.5808
$(\mathbb{H}^2)^2 \times \mathbb{E}^2 \times (\mathbb{S}^2)^2$	0.0765	0.0391	0.8672	0.0380	0.8152	0.0474	0.5951
Best model	$\mathbb{S}^5_{1.0} \times \mathbb{S}^5_{1.1}$	$\mathbb{H}^2_{.3} \times \mathbb{H}^2_{.6} \times \mathbb{H}^2_{1.5} \times (\mathbb{H}^2_{1.2})^2$		$\mathbb{H}^5_{3.4} \times \mathbb{S}^5_{12.6}$		$\mathbb{H}^5_{0.3} \times \mathbb{S}^5_{3.5}$	
D_{avg} improvement over single space	0.8%	28.89%		16.75%		32.55%	

- Fix total dimension (d) = 10
- cities graph (intrinsic structure is S^2) embeds well into product with spherical component(s)
- Similarly, PhDs (tree-like structure) in product with hyperbolic component(s)
- Even with data which matches a single constant curvature space, product space does not harm the performance
- In products of identical spaces, the curvatures can be non-uniform, instead of identical

	CS PhDs	Power	Facebook
Estimated Signature	$\mathbb{H}^5_{1.3} \times \mathbb{H}^5_{0.2}$	$\mathbb{H}^5_{1.8} \times \mathbb{S}^5_{1.7}$	$\mathbb{H}^5_{0.9} \times \mathbb{S}^5_{1.6}$

Heuristic allocation of signature and curvature:

- Given d_1, d_2, find K_1, K_2; report corresponding space with min. D_{avg}
- The curvature signs match that with the products of two models spaces with min. D_{avg}

	Dim 50			Dim 100		
	WS-353	Simlex	MEN	WS-353	Simlex	MEN
Euclidean	0.6628	0.2738	0.7217	0.6986	0.2923	0.7473
Hyperbolic	0.6787	0.2784	0.7117	0.6846	0.2832	0.7217
2 Hyperbolics	0.6955	**0.2870**	0.7246	0.7297	0.3168	0.7450
5 Hyperbolics	**0.7048**	0.2837	**0.7270**	**0.7379**	**0.3212**	**0.7530**

- Spearman rank correlation between obtained scores and annotated ratings on the word similarity dataset
- Hyperbolic embedding learnt with $P(y|w,u) = \sigma\left((-1)^{1-y}(-\cosh(d(\alpha_u, \gamma_w)) + \theta)\right)$

	Dim 50			Dim 100		
	WS-353	Simlex	MEN	WS-353	Simlex	MEN
Euclidean	0.6628	0.2738	0.7217	0.6986	0.2923	0.7473
Hyperbolic	0.6787	0.2784	0.7117	0.6846	0.2832	0.7217
2 Hyperbolics	0.6955	**0.2870**	0.7246	0.7297	0.3168	0.7450
5 Hyperbolics	**0.7048**	0.2837	**0.7270**	**0.7379**	**0.3212**	**0.7530**

- Spearman rank correlation between obtained scores and annotated ratings on the word similarity dataset
- Hyperbolic embedding learnt with $P(y|w,u) = \sigma\left((-1)^{1-y}(-\cosh(d(\alpha_u, \gamma_w)) + \theta)\right)$

- In literature, hyperbolic embeddings perform favorably against Euclidean word vectors in low dimensions (d = 5, 20), but less so in higher dimensions (d = 50, 100).
- Hypothesis is that in high dimensions, a product of multiple smaller-dimension hyperbolic spaces will substantially improve performance (as shown in the table)

	Dim 50			Dim 100		
	WS-353	Simlex	MEN	WS-353	Simlex	MEN
Euclidean	0.6628	0.2738	0.7217	0.6986	0.2923	0.7473
Hyperbolic	0.6787	0.2784	0.7117	0.6846	0.2832	0.7217
2 Hyperbolics	0.6955	**0.2870**	0.7246	0.7297	0.3168	0.7450
5 Hyperbolics	**0.7048**	0.2837	**0.7270**	**0.7379**	**0.3212**	**0.7530**

- Spearman rank correlation between obtained scores and annotated ratings on the word similarity dataset
- Hyperbolic embedding learnt with $P(y|w,u) = \sigma\left((-1)^{1-y}(-\cosh(d(\alpha_u, \gamma_w)) + \theta)\right)$

- In literature, hyperbolic embeddings perform favorably against Euclidean word vectors in low dimensions (d = 5, 20), but less so in higher dimensions (d = 50, 100).
- Hypothesis is that in high dimensions, a product of multiple smaller-dimension hyperbolic spaces will substantially improve performance (as shown in the table)

Total Dim d / Model	$\mathbb{R}^d$	$(\mathbb{H}^d)^1$	$(\mathbb{H}^{d/2})^2$	$(\mathbb{H}^{d/5})^5$	$(\mathbb{H}^2)^{d/2}$
50	0.3866	0.3424	0.3928	0.4181	**0.4209**
100	**0.5513**	0.3738	0.4310	0.4731	0.5216

- Analogy tasks: again observe product of multiple smaller-dimension hyperbolic spaces improve performance
- Design (somewhat like a) parallelogram in the product space with $d^2(a,b) = d^2(c,d)$ and $d^2(a,c) = d^2(b,d)$
  - Pair- a:b :: c:d
  - Simply reflect a w.r.t b-c geodesic (at mean m of b-c)

What are we concluding then?

- Product of model spaces improve representations
- We saw how to learn embeddings and curvatures, estimate signatures, and compute mean
- Experiments which validate above claims along with importance of products of smaller hyperbolic spaces than a single space

What are we concluding then?

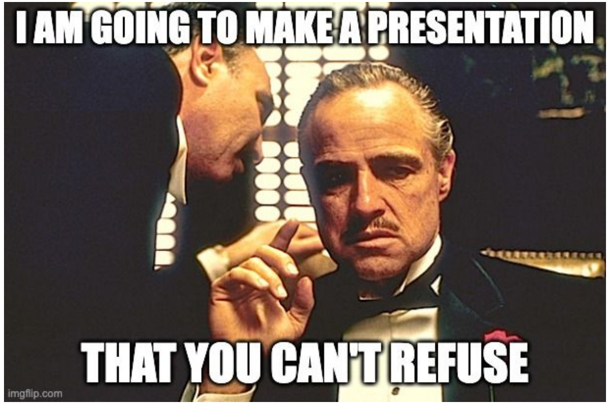
- Product of model spaces improve representations
- We saw how to learn embeddings and curvatures, estimate signatures, and compute mean
- Experiments which validate above claims along with importance of products of smaller hyperbolic spaces than a single space

A couple of personal thoughts

- The signature estimation can be carried out for product of two model spaces
- In the table supporting their claim "signature estimation matches curvature"
    - no exact numbers corresponding to distortion values
    - although we simply need the curvature signs, but it would be be good to observe difference between distortion values estimated curvatures and actual least distortion values
- Is there another way to identify dimensions (rather than doubling until reach the final total dimension)?
    - We are missing several combinations of product spaces (can they be discarded?)

Thank You!

Thank You!


I AM GOING TO MAKE A PRESENTATION
THAT YOU CAN'T REFUSE
imgflip.com