Wireless Channel Charting for Massive MIMO

Maxime Guillaud[1] and Christoph Studer[2]

[1]Huawei Technologies France

[2]ETH Zurich, Switzerland

IEEE WCNC 2022 Tutorial

Thanks to Olav Tirkkonen for discussions on Channel Charting,
and to Paul Ferrand, Alexis Decurninge, and Luis Garcia Ordóñez.

Channel charting resources website


Channel Charting Resources

locating user equipment within the radio environment using channel state information

What is Channel Charting?

Channel charting learns a mapping from channel state information (CSI) to a so-called **channel chart** in which nearby datapoints indicate nearness in real space. In other words, the learned channel chart captures the spatial geometry of the transmitting user equipments (UEs), effectively encoding relative (or logical) UE locations. Channel charting is self-supervised as the mapping from CSI to the channel chart is learned only using a database of passively collected CSI information. Such a data-driven localization approach has the advantages of being scalable and avoiding reference location information, e.g., from global navigation satellite systems (GNSSs). The self-supervised nature of channel charting also avoids the need for line-of-sight (LoS) propagation conditions or (costly) measurement campaigns, while enabling the infrastructure basestations or access points to perform cognitive and predictive radio access network (RAN) tasks which are tied to UE location.

Typical Channel Charting Pipeline

- Up-to-date list of channel charting papers
- Channel charting source code
- Datasets
- More to come...

channelcharting.github.io

Outline

- **Part 1:** Prerequisites
- **Part 2:** Channel charting basics
- **Part 3:** Classical dimensionality reduction (DR) methods
- **Part 4:** Neural-network-based DR and performance metrics
- **Part 5:** Feature design and synthetic results
- **Part 6:** Distance learning and experimental results
- **Part 7:** Extensions and applications
- **Part 8:** Open problems and future directions

Part 1:

Prerequisites

MIMO digital communications basics I

MIMO baseband model used by communications engineers:

$$\mathbf{r}_n = \mathbf{H}_n\mathbf{s}_n + \mathbf{w}_n$$

- n is a discrete index (corresponding to time, frequency, etc.)
- $\mathbf{s}_n$ is the transmit vector; contains the data to be sent
- $\mathbf{H}_\mathrm{n}$ is the MIMO channel matrix; **models wireless propagation**
- $\mathbf{w}_n$ is an additive disturbance (interference, noise, etc.)
- $\mathbf{r}_n$ is the received signal vector

MIMO digital communications basics II

- We will **extensively** focus on the MIMO channel matrix $\mathbf{H}_n$, which contains **channel state information** (CSI)

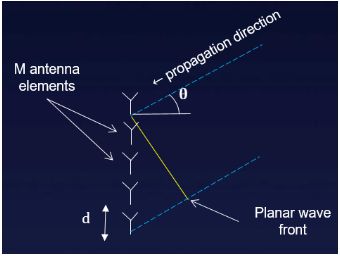
The **channel state** $\mathbf{H}_n$ (and often also the noise and signal variance) is a critical ingredient in baseband signal processing

- The receiver needs knowledge of $\mathbf{H}_n$ to estimate $\mathbf{s}_n$ from $\mathbf{r}_n$
- $\mathbf{H}_n$ shall be tracked **in real time** and, ideally, predicted

The channel matrix $\mathbf{H}_n$ is estimated during a dedicated training phase via pilots, but typically discarded after detection or precoding

The entries in $\mathbf{H}_n$ are governed by Maxwell's equations!

- **Toy example:** Simple plane-wave model with receiver equipped with uniform linear array (ULA) and $\lambda/2$ antenna spacing



- Transmitter (Tx) in far-field
- Small number L of arriving paths; e.g., line-of-sight path & scatterers
- Only one UE and one antenna at Tx $\rightarrow$ $\mathbf{H}_n$ **reduces to a column** $\mathbf{h}_n$
- Channel vector model:

$$\mathbf{h}_n = \sum_{\ell=0}^{L-1} \alpha_\ell \mathbf{a}(\omega_\ell) \quad \text{with} \quad \mathbf{a}(\omega) = [e^{j0\omega}, e^{j1\omega}, \ldots, e^{j(B-1)\omega}]^T$$

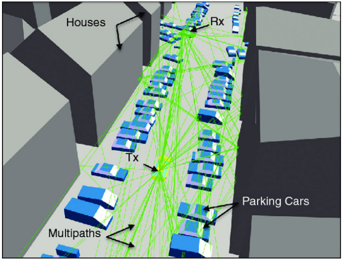
where $\omega_\ell = 2\pi \frac{d}{\lambda} \cos(\theta_\ell)$ with ℓth incident angle θ_ℓ

Key fact: Channel state $\mathbf{H}_n$ can be high-dimensional, but entries are often **described by a small set of variables**

Real-world channels are extremely complicated

In reality, geometric propagation models are intractable:

- The **number of channel parameters explodes**
- Parameters difficult/impossible to estimate; e.g., scatterers at unknown position, unknown electromagnetic properties, etc.



(image from: Reichardt, Pontes, Wiesbeck, Zwick, "Virtual Drives in Vehicular Communications," 2011.)

Analyze real-world channels using a data-driven approach
$\rightarrow$ **Channel Charting** [1]

[1] C. Studer, S. Medjkouh, E. Gönültaş, T. Goldstein, and O. Tirkkonen, "Channel Charting: Locating Users within the Radio Environment using Channel State Information," IEEE Access, 2018

Part 2:

Channel charting basics

In an ideal world...

... each position in space $\mathbf{p}_n \in \mathbb{R}^d$ with $d = 3$ would lead to a unique CSI feature $\mathbf{x}_n \in \mathbb{C}^D$ at the BS/AP side

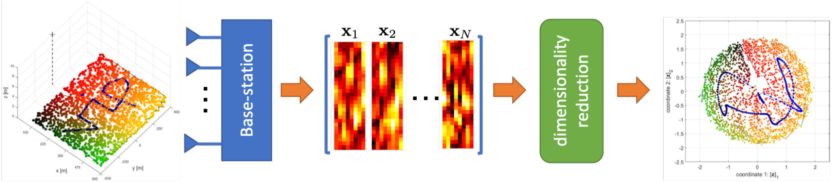
- CSI features $\{\mathbf{x}_n\}_{n=1}^N$ are high-dimensional $D \gg d = 3$
- Hope: collection of CSI features "live" on a low-dimensional manifold $\rightarrow$ **intrinsic dimensionality is** $d = 3$

Channel charting learns the low-dimensional manifold structure present in CSI features

✓ **Self-supervised** $\rightarrow$ no ground-truth location necessary

✓ Recycles CSI, which is acquired anyway at BS/AP

Channel charting: Self-supervised UE “localization”

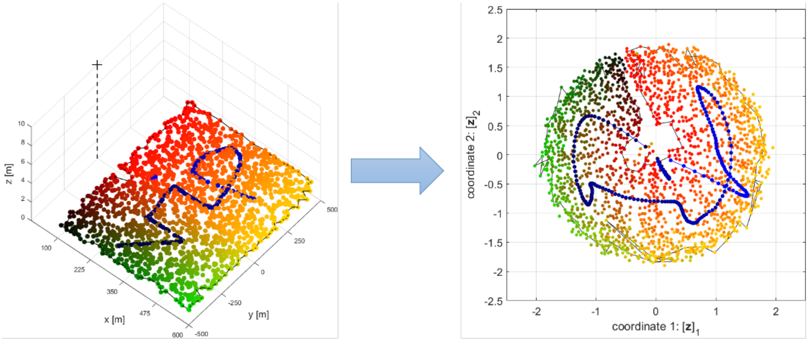


- Transmitting pilots from a location in space $\mathbf{p}_n \in \mathbb{R}^d$ generates a high-dimensional CSI vector $\mathbf{h}_n \in \mathbb{R}^D$ at the BS with $D \gg d$
- Assumption: CSI features $\{\mathbf{x}_n\}_{n=1}^N$ extracted at BS from CSI $\{\mathbf{h}_n\}_{n=1}^N$ are close to a low-dimensional manifold

Channel charting learns this low-dimensional manifold from CSI features in a data-driven and scalable manner [1]

[1] CS, S. Medjkouh, E. Gönültaş, T. Goldstein, and O. Tirkkonen, “Channel charting: Locating users within the radio environment using channel state information," IEEE Access, Vol. 6, pp. 47682-47698, Aug. 2018

An idealistic view on the capabilities of channel charting



32 BS antennas; uniform linear array with $\lambda/2$ antenna spacing; 2 GHz; narrowband; simulated at 0 dB SNR

- We can extract relative UE location in a self-supervised manner, without GNSS, antenna calibration, or LoS connectivity
- We can predict future actions of UEs that are tied to location

Unfortunately, reality is much more tricky...

Part 3:

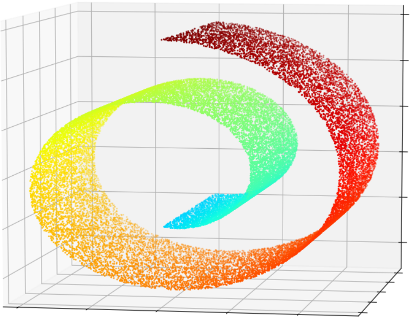
Classical dimensionality reduction approaches

Consider a Swiss roll

(photo credit: Max Pixel)

- Believed to have originated elsewhere, likely Austria...

Consider a Swiss roll



- Data-points (samples) in $\mathbb{R}^3$ (color is arbitrary)
- However, the position of any data-point on the Swiss roll manifold can be parametrized by only 2 variables...

Dimensionality reduction aims at identifying the low-dimensional parametric description of such sample distributions

What is a manifold?

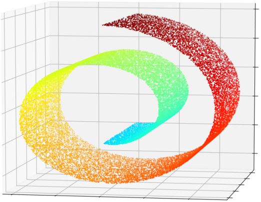
(image: Free Icons Download)

- Ex.: Earth is roughly a 3D sphere, but appears flat (2D) to us
- Manifold: a space that **locally resembles an Euclidean space**
- Each point of a D-dimensional manifold has a neighborhood that can be mapped to Euclidean space of dimension d

Dimensionality reduction I

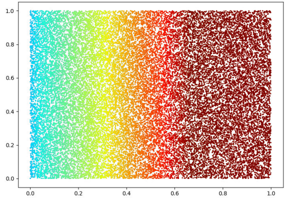
- Real-world data, such as speech signals, images, videos, MRI scans, channel measurements are usually high dimensional
- Dimensionality reduction: **mapping** of high-dimensional data into a **meaningful representation** of low dimension
- Ideally: Mapping should **preserve neighborhoods** in dataset with dimension corresponding to **intrinsic dimensionality**

Samples in $\mathcal{H}^D$ (ambient space)



$\longrightarrow$

Low-dimensional embedding in $\mathcal{H}^d$ (latent space)



Dimensionality reduction II

- Each sample maps to a point in the low-dimensional latent space ($d \ll D$):
$$\begin{array}{ccc} \mathcal{H}^D & \rightarrow & \mathcal{H}^d \\ \mathbf{x}_i & \overset{f}{\mapsto} & \mathbf{y}_i \end{array}$$

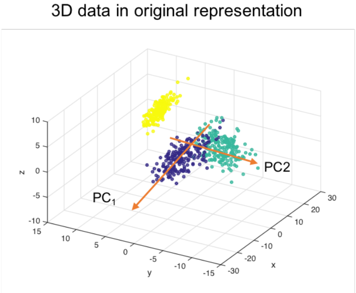
- Function f is **optimized based on dataset** such that the resulting low-dimensional **embedding** $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^N$ respects local geometry (neighborhoods) of the high-dimensional dataset

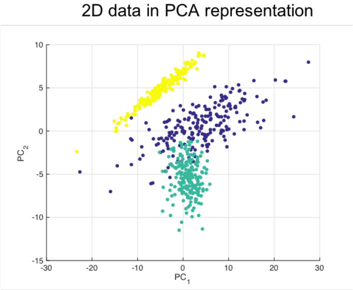
- Example: locally Euclidean manifold so that $\|\boldsymbol{y}_i - \boldsymbol{y}_j\| \approx \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ for nearby points, i.e., for small $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$

- Dimensionality reduction is **self-supervised** $\rightarrow$ label free
- Dimensionality reduction is **continuous version of clustering**
- Dimensionality reduction is ill-posed and requires assumptions on the data: **dimension d, similarity measures, etc.**

Dimensionality reduction using classical methods

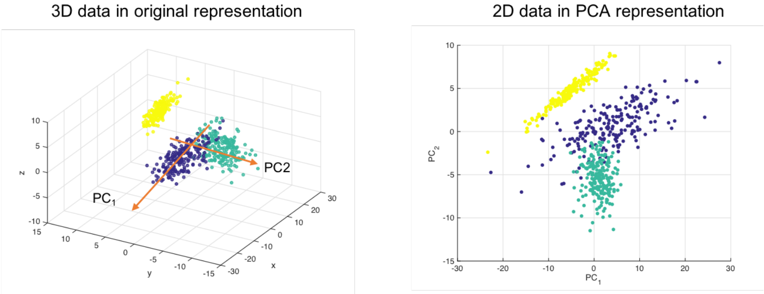
Principal component analysis (PCA) I





- PCA-based dimensionality reduction proceeds as follows:
  - Preprocess data-points $\bar{\mathbf{x}}_n = \mathbf{x}_n - \mathbf{m}$ by subtracting mean $\mathbf{m}$
  - Extract matrix $\mathbf{M}$ from dominant left singular vectors of zero-mean data set matrix $\bar{\mathbf{X}} = [\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_N]$
  - $\mathbf{y}_n = \mathbf{M}(\mathbf{x}_n - \mathbf{m})$ where $\mathbf{M} \in \mathcal{H}^{d \times D}$ and $\mathbf{M}\mathbf{M}^H = \mathbf{I}_d$

Principal component analysis (PCA) II



- PCA is **affine**, and aims at maximizing variance (entropy, if data is Gaussian) in low dimensional space
- PCA is **parametric**, i.e., learns a function $f : \mathcal{H}^D \to \mathcal{H}^d$ that can map new (unseen) points $\mathbf{x}_n$ to low dimensional space

$$\mathbf{y}_n = f_\theta(\mathbf{x}_n) = \mathbf{M}(\mathbf{x}_n - \mathbf{m})$$

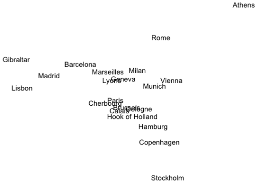
with function parameters $\theta = \{\mathbf{M}, \mathbf{m}\}$

Multi-dimensional scaling (MDS) I

Eurostat Pairwise distances (by road) data

	Athens	Barcelona	Brussels	Calais	Cherbourg	Cologne
Barcelona	3313					
Brussels	2963	1318				
Calais	3175	1326	204			
Cherbourg	3339	1294	583	460		
Cologne	2762	1498	206	409	785	
Copenhagen	3276	2218	966	1136	1545	760
Geneva	2610	803	677	747	853	1662
Gibraltar	4485	1172	2256	2224	2047	2436
Hamburg	2977	2018	597	714	1115	460

$\longrightarrow$



- MDS assigns coordinates to points for which **pairwise distances** are known
- Generate fictitious coordinates in an arbitrary (Euclidean) space that approximate the given pairwise distances
- Solution is **nonunique** up to a global translation and a unitary transform (rotation)

Nonparametric: generates embedding $\mathcal{Y}$, but no explicit mapping

Multi-dimensional scaling (MDS) II

- Goal: Find low-dimensional embedding that preserves all **pairwise distances**, i.e., $\|\mathbf{x}_i - \mathbf{x}_j\| \approx \|\mathbf{y}_i - \mathbf{y}_j\|$

- Metric MDS [1] minimizes the following cost function:

$$L(\{\mathbf{y}_n\}_{n=1}^N) = \frac{2}{N^2 - N} \sum_{i<j} (\|\mathbf{x}_u - \mathbf{x}_j\| - \|\mathbf{y}_i - \mathbf{y}_j\|)^2$$

- One typically includes a constraint that embedding is zero-mean

The MDS cost function is nonconvex but embeddings can be computed approximately and efficiently using gradient descent

[1] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," Psychometrika, Mar. 1964

Sammon mapping (SM)

Metric MDS tries to match all pairwise distances between nearby and also far away points, which is **often infeasible**

- Sammon's mapping preserves only **small pairwise distances** $\rightarrow$ manifolds should be **locally** Euclidean
- Sammon mapping [1] minimizes the following cost function:

$$L(\{\mathbf{y}_n\}_{n=1}^N) = \frac{2}{N^2 - N} \sum_{n<m} \frac{1}{\|\mathbf{x}_n - \mathbf{x}_m\|} (\|\mathbf{x}_n - \mathbf{x}_m\| - \|\mathbf{y}_n - \mathbf{y}_m\|)^2$$

- Pairwise distances for which $\|\mathbf{x}_n - \mathbf{x}_m\|$ is large are less important; other de-weighting functions are possible

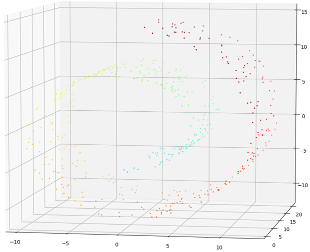
The SM cost function is nonconvex but solutions can be found approximately and efficiently using (projected) gradient descent

[1] J. W. Sammon, "A nonlinear mapping for data structure analysis," IEEE Trans. Comput., May 1969

Isomap

Isomap [1] tries to recover the manifold structure using a graph

- Construct a graph of "neighbor" samples (based on distance threshold)
- Use pairwise distances on graph to **approximate the geodesic distances** on the manifold
- Use MDS on the geodesic distances $D_{i,j}$ to assign Euclidean coordinates

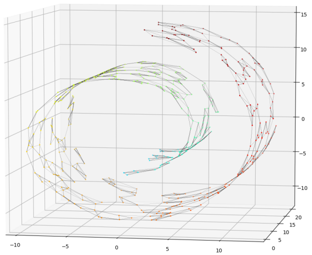


[1] J. B. Tenenbaum, V. de Silva and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," Science, 2000.

Isomap

Isomap [1] tries to recover the manifold structure using a graph

- Construct a graph of "neighbor" samples (based on distance threshold)
- Use pairwise distances on graph to **approximate the geodesic distances** on the manifold
- Use MDS on the geodesic distances $D_{i,j}$ to assign Euclidean coordinates

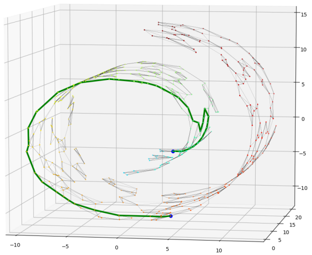


[1] J. B. Tenenbaum, V. de Silva and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," Science, 2000.

Isomap

Isomap [1] tries to recover the manifold structure using a graph

- Construct a graph of "neighbor" samples (based on distance threshold)
- Use pairwise distances on graph to **approximate the geodesic distances** on the manifold
- Use MDS on the geodesic distances $D_{i,j}$ to assign Euclidean coordinates



[1] J. B. Tenenbaum, V. de Silva and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," Science, 2000.

t-distributed stochastic neighbor embedding (t-SNE)

- t-SNE [1] is a widespread dimensionality reduction method used for data visualization (often 2D or 3D embeddings)
- Learns a low-dimensional representation by minimizing the KL divergence between the distributions of original pairwise similarities, and the low-dimensional pairwise similarities:

$$(t\text{-SNE}) \qquad \text{minimize } \textstyle\sum_i \sum_j P_{i,j} \log \frac{P_{i,j}}{Q_{i,j}},$$

$\{P_{i,j}\}$ original probabilities with Gaussian kernels

$\{Q_{i,j}\}$ representation probabilities with t-distribution kernels

Solution can approximately be found via gradient descent

[1] L. J. P. van der Maaten and G. E. Hinton, "Visualizing High-Dimensional Data Using t-SNE," J. Machine Learning Research , Nov. 2008

Other dimensionality reduction methods

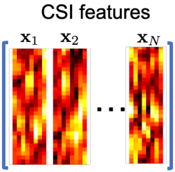
- **Laplacian eigenmaps**: Spectral DR applied to graph Laplacian based on k-NN or heat kernel
- **UMAP**: Related to t-SNE but avoids scaling issues to reduce complexity for (stochastic) gradient descent
- **Diffusion maps**: Spectral DR applied to diffusion distance graph simulating a Markov random walk between samples
- **Maximum variance unfolding**: Spectral DR applied to feature matrix from optimization problem with local isometry constraint

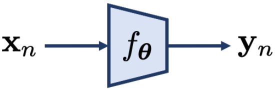
All of the methods discussed so far (except PCA)
are based on **pairwise distances** in $\mathcal{H}^D$

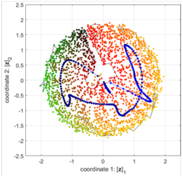
Part 4:

Neural-network-based dimensionality reduction and performance metrics

Why neural networks?

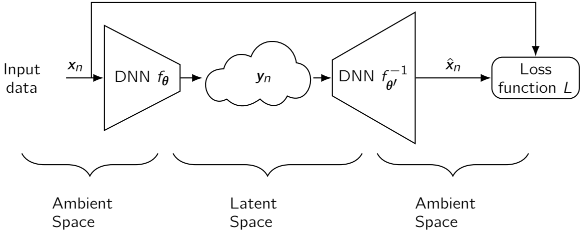






- Deep neural networks are excellent function approximators
- Deep neural networks can be trained efficiently from very large datasets using stochastic gradient descent
- **Hope 1:** We can train a neural network $\mathbf{y}_n = f_\theta(\mathbf{x}_n)$ that performs parametric dimensionality reduction
- **Hope 2:** We can train such neural networks in a self-supervised manner, without labels or ground truth location information

Autoencoders



- Autoencoders [1] are **parametric** and learn two functions:
  - Encoder function $\mathbf{y}_n = f_\theta(\mathbf{x}_n)$ from $\mathcal{H}^D \rightarrow \mathcal{H}^d$
  - Decoder function $\hat{\mathbf{x}}_n = f_{\theta'}^{-1}(\mathbf{y}_n)$ from $\mathcal{H}^d \rightarrow \mathcal{H}^D$
- Goal: learn weights $\{\boldsymbol{\theta}, \boldsymbol{\theta}'\}$ of two functions that minimize **pairwise** reconstruction error $\|\hat{\boldsymbol{x}}_n - \boldsymbol{x}_n\|$

$$L(\theta, \theta') = \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{x}_n - f_{\theta'}^{-1}(f_\theta(\mathbf{x}_n))\|^2$$

[1] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," AIChE Journal, 1991

How to constrain the geometry in the latent space?

Autoencoders do not preserve local Euclidean geometry

- Recall: Sammon's mapping preserves **small pairwise distances**

$$L(\{\mathbf{y}_n\}_{n=1}^N) = \frac{2}{N^2 - N} \sum_{n<m} \frac{1}{\|\mathbf{x}_n - \mathbf{x}_m\|} (\|\mathbf{x}_n - \mathbf{x}_m\| - \|\mathbf{y}_n - \mathbf{y}_m\|)^2$$

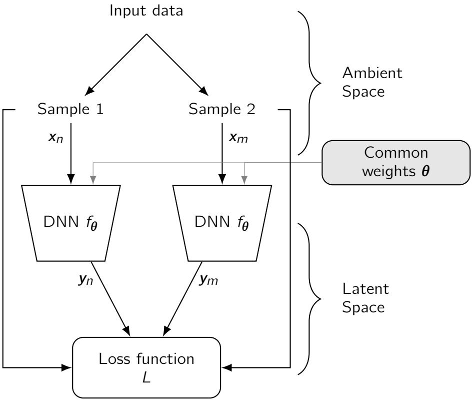
- Instead of computing nonparametric embedding $\{\mathbf{y}_n\}_{n=1}^N$, directly learn function $\mathbf{y}_n = f_\theta(\mathbf{x})$ that performs DR:

$$L(\boldsymbol{\theta}) = \frac{2}{N^2 - N} \sum_{n<m} \frac{1}{\|\mathbf{x}_n - \mathbf{x}_m\|} \big(\|\mathbf{x}_n - \mathbf{x}_m\| - \| f_{\boldsymbol{\theta}}(\mathbf{x}_n) - f_{\boldsymbol{\theta}}(\mathbf{x}_m)\|\big)^2$$

"Siamese" networks: parametric Sammon-type Mapping

- Minimizing $L(\boldsymbol{\theta})$ yields the structure of a so-called "Siamese" network [1]:

$$L(\boldsymbol{\theta}) = \frac{2}{N^2 - N} \sum_{n<m} \frac{1}{\|\mathbf{x}_n - \mathbf{x}_m\|} \left(\|\mathbf{x}_n - \mathbf{x}_m\| - \| f_{\boldsymbol{\theta}}(\mathbf{x}_n) - f_{\boldsymbol{\theta}}(\mathbf{x}_m) \| \right)^2$$



[1] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," Advances in Neural Information Processing Systems (NIPS), 1994

How can we measure the performance of an embedding?

Evaluating the performance of dimensionality reduction

- Dimensionality reduction is agnostic to labels → **classification performance cannot be used** as a performance metric
- There is typically no ground truth available for the embedded data points → **mean-square error (MSE) cannot be used**

To evaluate the performance of dimensionality reduction, we need to specify measures of **distance or dissimilarity** on both $\mathcal{X}$ and $\mathcal{Y}$

- A common distance (or dissimilarity) measure is the **Euclidean distance** $d(\mathbf{u}, \mathbf{u}') = \|\mathbf{u} - \mathbf{u}'\|$ between two points $\mathbf{u}, \mathbf{u}' \in \mathcal{H}^K$

The setup to measure performance

- Consider two abstract sets of N data-points: $\{\mathbf{u}_n\}_{n=1}^N$ in original space and $\{\mathbf{v}_n\}_{n=1}^N$ in representation space (embedding)
- Assume a relationship between pairs of points $\mathbf{u}_n \sim \mathbf{v}_n$
- Two measures based on neighborhood ranking:
  - **Continuity (CT):** how similar is ranking of nearest-K points in original space to ranking of points in representation space
  - **Trustworthiness (TW):** how similar is ranking of nearest-K points of embedding in original space
  - Both metrics in $[0, 1]$ and optimal if $CT = TW = 1$

Continuity (CT) and Trustworthiness (TW)

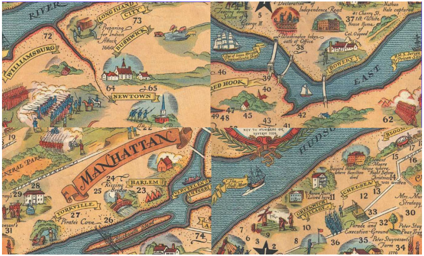
Continuity is penalized when one of the K-neighbors in original space is far away (dissimilar) in representation space

Trustworthiness is penalized when one of the K-neighbors in the representation space is far away (dissimilar) in the original space

Limitations of CT and TW

- Both metrics are based on **distances in $\mathcal{H}^D$**
- CT and TW are local metrics, **oblivious to global errors**

(image source: Brooklyn Historical Society)



Example: almost all points in the dataset have the same K neighbors in the ambient and latent space: $CT \approx 1$ and $TW \approx 1$

Kruskal stress [1]

- **Kruskal stress (KS)** compares pairwise distances:

$$KS = \min_{\beta} \sqrt{\frac{\sum_{n,m} (\delta_{n,m} - \beta d_{n,m})^2}{\sum_{n,m} \delta_{n,m}^2}}$$

with $\delta_{n,m} = \|\mathbf{u}_n - \mathbf{u}_m\|$, $d_{n,m} = \|\mathbf{v}_n - \mathbf{v}_m\|$

- Parameter β is selected to minimize KS

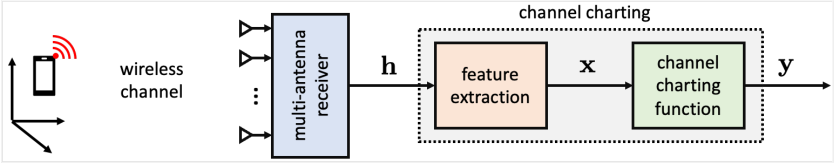
- Kruskal stress is invariant to rotation and global shifts

[1] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," Psychometrika, Mar. 1964

Part 5:

CSI feature design
and synthetic results

Outline of a CC pipeline



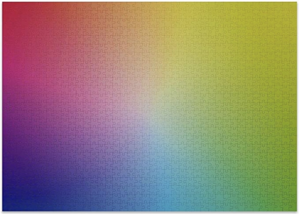
1. UEs transmit data to the multi-antenna receiver (Rx)
2. Rx estimates the wireless channel for data detection
3. Channel-state information is recycled $\rightarrow$ **channel features**
4. Perform dimensionality reduction for channel charting

Requires no true location information $\rightarrow$ **self-supervised**

CC ingredient I: Feature engineering

In principle, one could learn channel chart directly from CSI $\rightarrow$ astronomically large data set and excessive training complexity

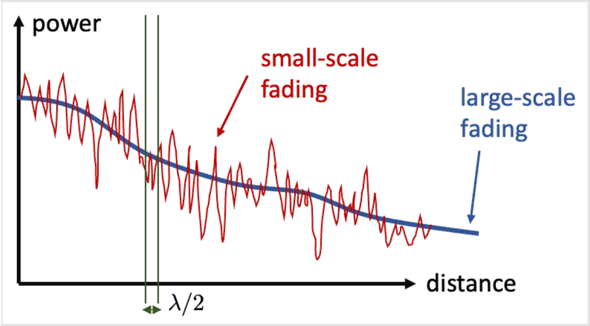
(image source: Cloudberries puzzles)



(image source: Cloudberries puzzles)

- CSI features corresponding to nearby locations must be "sufficiently similar" to recover local geometry efficiently

CC ingredient I: Feature engineering (cont'd)



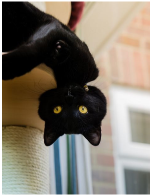
- Suitable CSI features are robust to small-scale fading, noise, system/hardware impairments, etc.
- Suitable CSI features capture large-scale fading properties, such as angle-of-arrival, receive power, etc.

Role of the features

Features are functions of the input samples which should

- **Remove (=be invariant to) spurious transformations** of the data due to sampling, noise etc.
- Preserve the relevant part of the signal

Example: rotation is a spurious transformation for the purpose of object classification

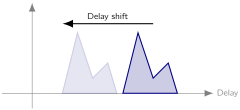


Expert-designed features have the advantage (e.g., over data augmentation) of **reducing complexity and dimensionality** of the unsupervised DR problem

Common impairments in CSI estimation

Some impairments can make samples more dissimilar than they really are:

- **multiplicative impairments** (gain and phase shift) per antenna port, due to RF components manufacturing variations;
- **clock and frequency offsets, jitter** between the Tx and Rx;
- **imperfect synchronization.**



Solution: Appropriate feature designs (see [1])

- Autocorrelation (invariant to time shifting)
- Spatial Fourier transform yields sparser angular representation

[1] P. Ferrand, A. Decurninge, and M. Guillaud, "DNN-based Localization from Channel Estimates: Feature Design and Experimental Results," IEEE Globecom 2020

CC ingredient II: Dimensionality reduction

- Simple CSI feature extraction pipeline:
  - Transform multi-antenna CSI $\mathbf{h}_n$ into angular domain $\hat{\mathbf{h}}_n = \mathbf{F}\mathbf{h}_n$
  - Remove global phase* and global scale $\mathbf{x}_n = |\hat{\mathbf{h}}_n|/\|\hat{\mathbf{h}}_n\|$

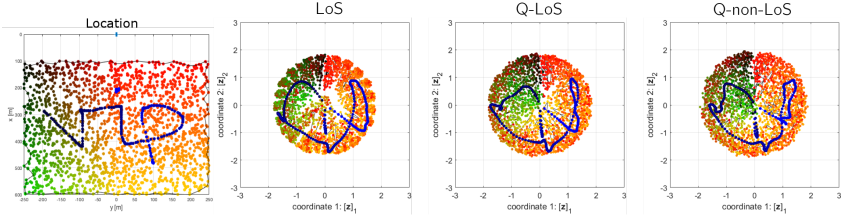
CSI features must be adapted to communication system, receiver hardware, and CSI sampling density

- **Dimensionality reduction**: Learn low-dimensional structure in set of CSI features $\{\mathbf{x}_n\}_{n=1}^N$ that preserves **pairwise similarities**
- Optimization problem yields channel chart $\{\mathbf{y}_n\}_{n=1}^N$, e.g., solve:

$$L(\{\mathbf{y}_n\}_{n=1}^N) = \sum_{n<m} \left(\|\mathbf{x}_n - \mathbf{x}_m\| - \|\mathbf{y}_n - \mathbf{y}_m\|\right)^2$$

*Notation $|\hat{\mathbf{h}}_n|$ denotes the entry-wise absolute value.

Channel charting is robust to different scenarios



32 BS antennas; uniform linear array with $\lambda/2$ antenna spacing; 2 GHz; narrowband; simulated at 0 dB SNR

- **LoS:** textbook line-of-sight channel (e.g., outdoors, rural)
- **Q-LoS:** Quadriga LoS channel (e.g., outdoors, urban)
- **Q-non-LoS:** Quadriga non-LoS channel (e.g., indoors, urban)

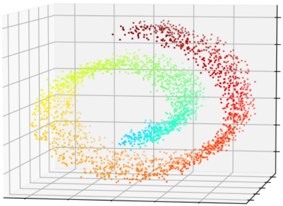
[1] S. Jaeckel, L. Raschkowski, K.Börner, and L. Thiele, "QuaDRiGa: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," IEEE Transactions on Antennas and Propagation, Mar. 2014

Part 6:

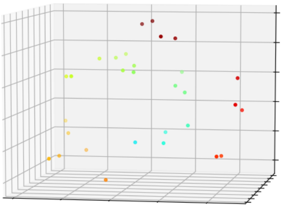
Distance learning
and experimental results

The issue with distance

- The cost functions considered so far all require the evaluation of Euclidean distances $\|\boldsymbol{x} - \boldsymbol{x}'\|$ in $\mathcal{H}^D$



3000 samples in $\mathbb{R}^3$



30 samples in $\mathbb{R}^3$

Curse of Dimensionality: distances in $\mathcal{H}^D$ are unreliable

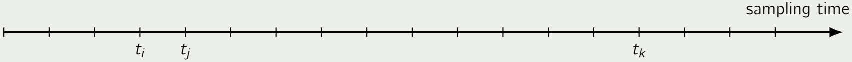
- In high dimensions, all samples are far away from each other
- **Pairwise distance is dominated by noise**, not by proximity

Idea: Learn a distance!

- In wireless communications, **CSI samples acquired over time are not independent**:
  - The speed at which the physical parameters of the system change is bounded: the **meaningful** latent parameters should change continuously in time

Distance Learning using Short-Term CSI Time Correlation

- **Almost successive samples** i and j should have smaller distances, than to a randomly picked sample k in the dataset: $d_{\mathrm{CSI}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq d_{\mathrm{CSI}}(\boldsymbol{x}_i, \boldsymbol{x}_k)$



Distance learning using triplets

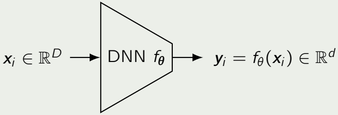
- Inspired by an approach to learn a Mahalanobis distance [1]
  - Select sample triplets $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k)$ acquired at times t_i, t_j, t_k, where $|t_i - t_j| \leq T \leq |t_i - t_k|$
  - $T \approx$ coherence time
  - The learned distance d_{CSI} should fulfill (with high probability)

$$d_{\mathrm{CSI}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq d_{\mathrm{CSI}}(\boldsymbol{x}_i, \boldsymbol{x}_k)$$

[1] C. Shen, J. Kim, L. Wang, and A. Hengel, "Positive semidefinite metric learning with boosting," Proc. Advances in Neural Information Processing Systems, 2009.

Joint distance learning and dimensionality reduction

DNN-based parametric approach with trainable parameters θ



$$d_{\mathrm{CSI}}(x, x') = \|f_{\boldsymbol{\theta}}(x) - f_{\boldsymbol{\theta}}(x')\|$$

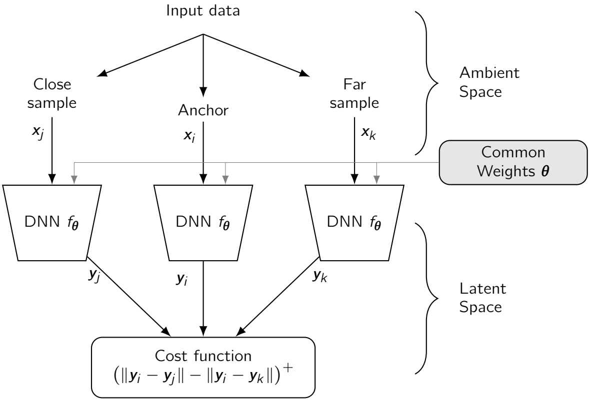
- Triplets should fulfill $\|f_{\boldsymbol{\theta}}(\mathbf{x}_i) - f_{\boldsymbol{\theta}}(\mathbf{x}_j)\| \leq \|f_{\boldsymbol{\theta}}(\mathbf{x}_i) - f_{\boldsymbol{\theta}}(\mathbf{x}_k)\|$
- Learn d_{CSI} by optimization over $\boldsymbol{\theta}$ of a **cost function that penalizes the violations of the inequality**:

$$\min_{\boldsymbol{\theta}} \sum_{(i,j,k)\in\mathcal{T}_T} \Big(\|f_{\boldsymbol{\theta}}(x_i) - f_{\boldsymbol{\theta}}(x_j)\| - \|f_{\boldsymbol{\theta}}(x_i) - f_{\boldsymbol{\theta}}(x_k)\|\Big)^+$$

where triplets in $\mathcal{T}_T$ fulfill the timestamp constraint
$|t_i - t_j| \leq T \leq |t_i - t_k|$

Triplet networks

- This defines a triplet network [1]:



- The cost function involves **only norms in $\mathcal{H}^d$**

[1] P. Ferrand, A. Decurninge, L. Garcia Ordoñez and M. Guillaud, "Triplet-Based Wireless Channel Charting: Architecture and Experiments," IEEE JSAC, Aug. 2021.

Experimental results

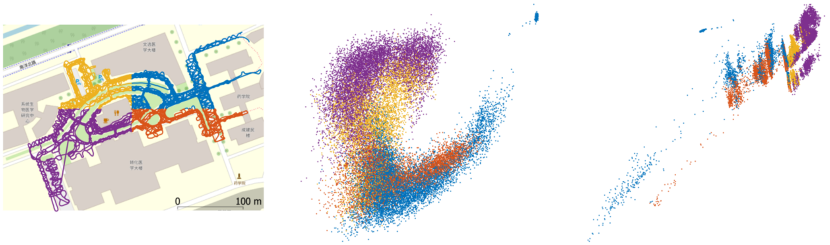
Massive MIMO data collection set-up

- Rooftop antenna array and pedestrian UE (both commercial 4G hardware)
- 64-element rectangular array ($4 \times 8 \times 2$ polarizations)
- 20 MHz channel bandwidth in the 3.5 GHz band ($\lambda = 8\,\text{cm}$)
- Uplink CSI recorded at BTS, 200 samples/second
- Timestamps t_i and GNSS locations recorded at the UE

Key dataset and neural network characteristics

- CSI data in $\mathbb{C}^{18432}$ (64 antenna elements $\times$ 288 subcarriers)
- Approximately 2.8 million samples collected over 4 hours
- We use a five-layer fully-connected neural network with batch normalization and ReLU to implement charting function $f_{\boldsymbol{\theta}}$

Experimental channel charting results ($d = 2$)



(a) Geographic position
(b) Principal Component Analysis (PCA)
(c) Autoencoder

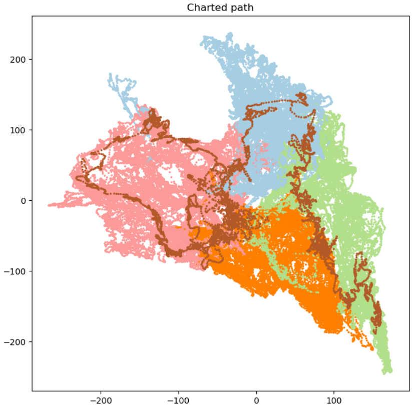
(d) Siamese network
(e) UMAP
(f) Triplets network (margin cost)

Similarity between chart and GNSS position

	Trustworthiness	Continuity	Kruskal stress
PCA ($d = 2$)	0.843	0.894	0.629
Siamese networks $d = 2$	0.846	0.910	0.494
Autoencoders $d = 2$	0.929	0.889	0.701
UMAP $d = 2$	0.951	0.923	0.526
Triplets (exp. loss) $d = 2$	0.957	0.975	0.238
Triplets (margin loss) $d = 2$	**0.967**	**0.977**	**0.206**
Siamese network $d = 5$	0.940	0.940	**0.496**
Autoencoders $d = 5$	0.972	0.968	0.877
Triplets (exp. loss) $d = 5$	0.969	0.974	0.767
Triplets (margin loss) $d = 5$	**0.973**	**0.976**	0.500
Siamese network $d = 10$	0.960	0.938	**0.500**
Autoencoders $d = 10$	0.975	0.967	0.694
Triplets (exp. loss) $d = 10$	0.978	**0.976**	0.850
Triplets (margin loss) $d = 10$	**0.979**	**0.976**	0.723

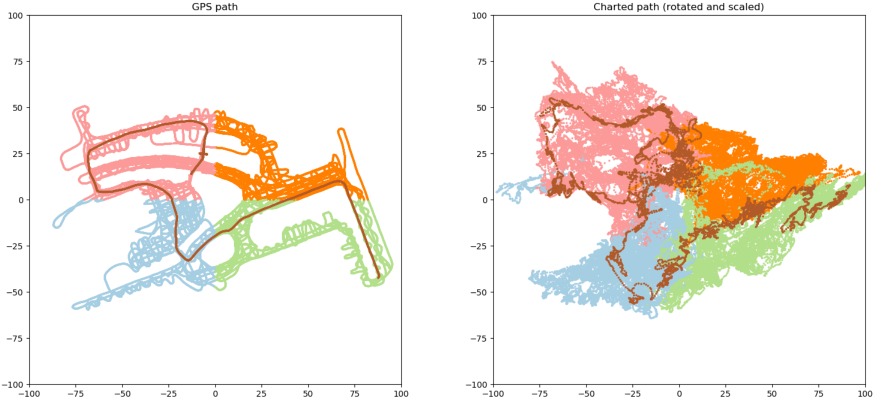
Separate datasets for chart construction and exploitation

- NN trained with $\approx 10^6$ CSI samples collected over 1.5 hour
- Brown dataset corresponds to a 5-minutes pedestrian trajectory recorded later



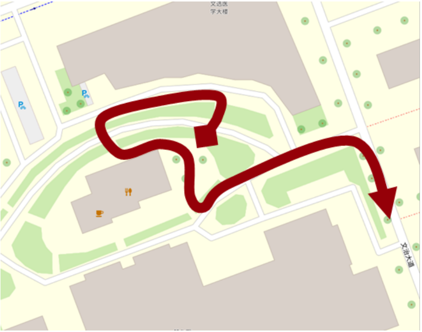
So what?

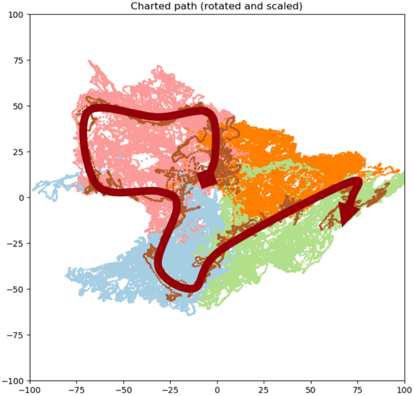
Several datasets



The output of this **self-supervised** approach shows **striking similarity to the GNSS position**!

Pseudo-position



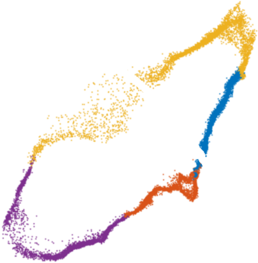


- Self-supervised pseudo-position as a replacement for (network-side) user position information, **without GNSS**

A Simple Cyclic Dataset

Ground track

Channel chart



Triplets (margin cost)

Chart topology shows strong similarity with the ground track!

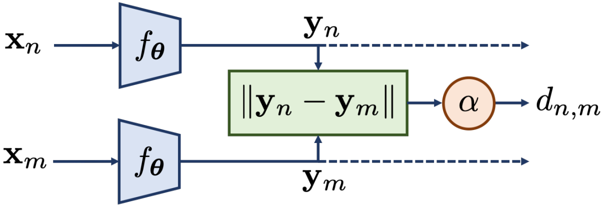
Part 7:

Extensions and applications

Extension I:
Semi-supervised channel charting

Semi-supervised channel charting [1]

Idea: Measure a small number of anchor positions $\mathbf{p}_n$,
e.g., with GNNS information, and associated CSI features $\mathbf{x}_n$

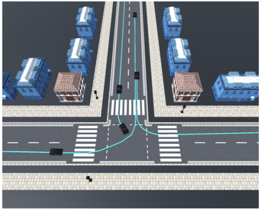


- Pass low-dimensional vectors $\mathbf{y}_n$ and $\mathbf{y}_m$ as secondary outputs
- By adding $\sum_{n\in\mathcal{N}} \|\mathbf{y}_n - \mathbf{p}_n\|^2$ to loss $L(\boldsymbol{\theta})$, we can enforce a-priori measured position information $\mathbf{p}_n$ from $n \in \mathcal{N}$

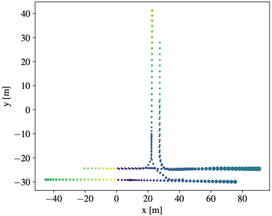
[1] E. Lei, O. Castañeda, O. Tirkkonen, T. Goldstein, and C. Studer, "Siamese Neural Networks for Wireless Positioning and Channel Charting," 57th Annual Allerton Conf., Sep. 2019

T-intersection simulation with very small dataset

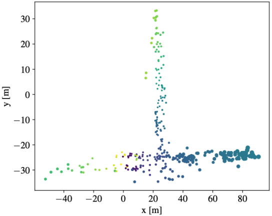
Unity traffic simulator



validation set of true tracks



validation set of predicted tracks



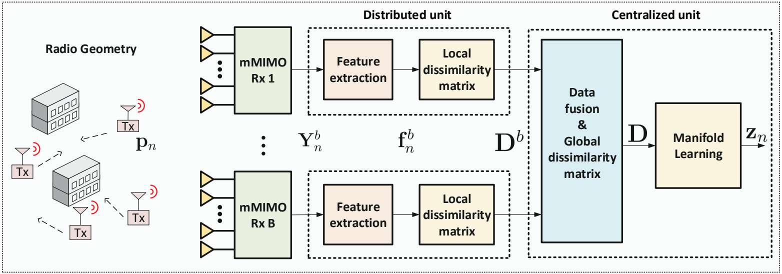
- Siamese network trained from only 20 UE traces at a T-intersection with 10% anchor point data:

	Q-LoS	**Q-NLoS**
MDE [m]	2.95	5.32
KS	0.188	0.471
TW ($K = 80$)	0.755	0.729
CT ($K = 80$)	0.974	0.944

- Performs well even with very small datasets
- Absolute positioning possible

Extension II:
Multi-point channel charting

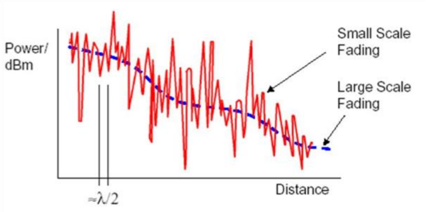
Multi-point channel charting [1]

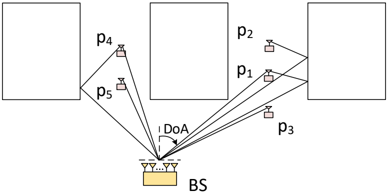


1. Several multi-antenna BSs serve all UEs distributed in area
2. Each BS estimates the CSI for a given UE
3. CSI is summarized into **channel features** at each BS
4. Perform data fusion and manifold learning for channel charting

[1] J. Deng, S. Medjkouh, N. Malm, O. Tirkkonen, and C. Studer, "Multipoint channel charting for wireless networks," 52nd Asilomar Conf. on Signals, Systems, and Computers, Oct. 2018

Channel features suitable for mmWave channels



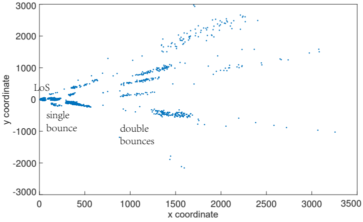


- Suitable channel features must be robust to small-scale fading effects
- Suitable channel features must capture large-scale fading properties
- We need a feature that is smoothly and slowly changing with the UE location

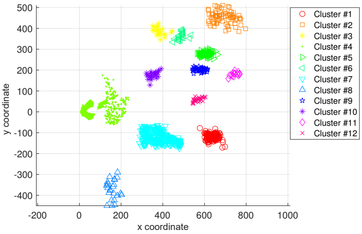
MUSIC*: directions-of-arrival $\{\phi_\ell\}_{\ell=1}^{L}$ and receive powers $\{|\beta_\ell|^2\}_{\ell=1}^{L}$ of the multi-path components $\longrightarrow$ distance between virtual sources

*multiple signal classification

Channel features II



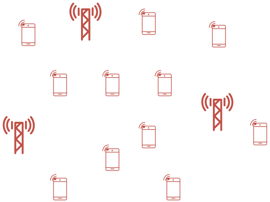
- Project DoA and associated power into **virtual sources**
- Nearby UE locations would have similar virtual sources



- Cluster the virtual sources after appropriate scaling
- **Dissimilarity is distance of virtual sources in clusters**

Multi-point channel charting

Goal is to capture global geometry with multiple BSs



- Instead of learning separate channel charts followed by fusion, we build one chart
- Build a common multi-cell chart by merging multiple BS "views"
- One can leverage concepts of multi-view machine learning and data fusion

- Combining all CSI information measured at multiple BSs mitigates distortion that occurs in single-point channel charting

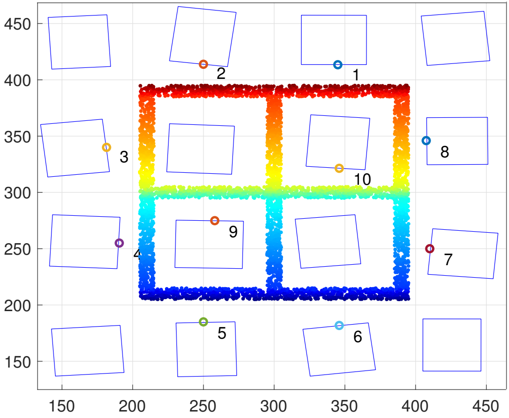
Multi-point channel charting (cont'd)

- Use a network-level weighted dissimilarity

$$D_{i,j} = \frac{1}{\sum_{b'=1}^{B} w_{b'}} \sum_{b=1}^{B} w_b d_b(\mathbf{x}_i, \mathbf{x}_j)$$

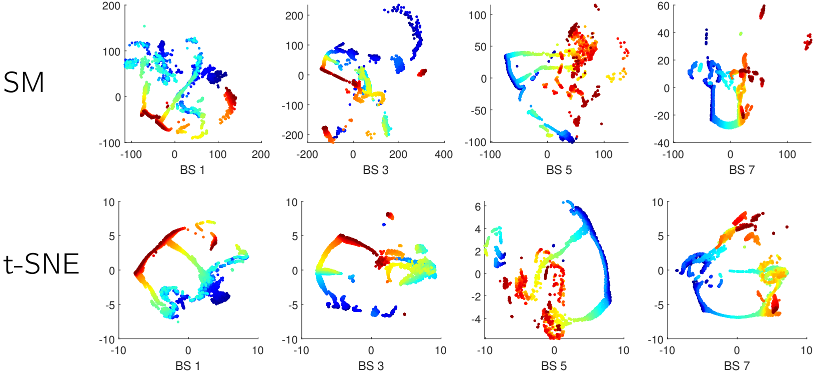
  - $d_b(\mathbf{x}_i, \mathbf{x}_j)$ dissimilarity between feature i and j at BS b
  - $w_b = \min\{\gamma_i^b, \gamma_j^b\}$ is a weight characterizing reliability of the dissimilarity generated by BS b for samples i and j
  - SNR γ_i^b is estimated using $\|\mathbf{h}_i^b\|_2^2/\sigma^2$ with noise power σ^2, and channel vector $\mathbf{h}_i^b$ between BS b and UE location $\mathbf{p}_i$

Scenario: Manhattan grid via ray-tracing



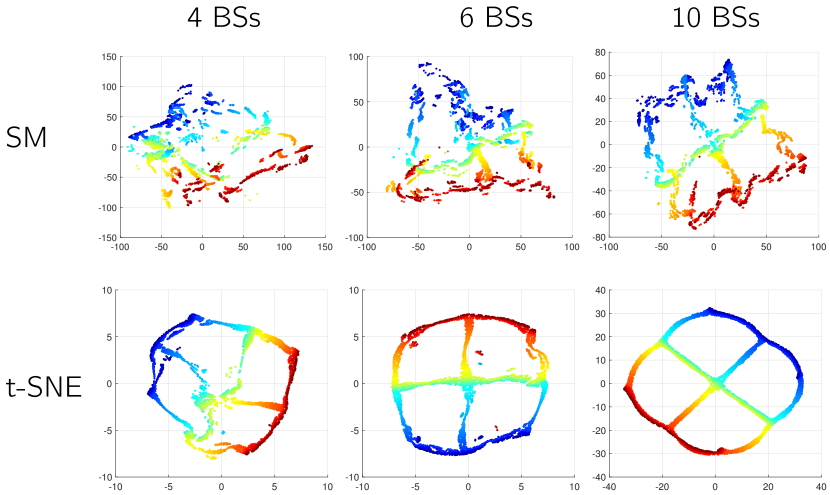
- 10 BSs samples UE Tx from positions marked by colors
- Each BS has 64 antennas with ULA $\lambda/2$ spacing
- 28 GHz carrier; 256 MHz bandwidth; 23 dBm UE-Tx power

Channel charting results: **single-point**



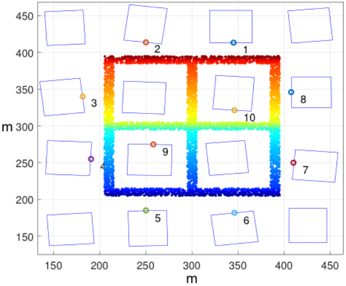
Manifold structure of sampled UE locations close to BS in question is well preserved; points far from the BS are distorted

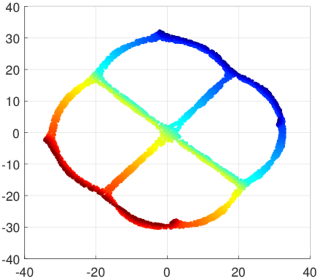
Channel charting results: **multi-point**



Multi-point channel charting captures global geometry!

Multipoint channel charting





Multipoint channel charting helps to unwrap the manifold
$\rightarrow$ **geometry is restored almost perfectly!**

Applications

Applications that can rely on pseudo-position

Advantages of pseudo-location information

- **Self-supervised:** easy to deploy, completely data driven, and adaptive to the system over time
- **Consistent** (in time, across users)

- **Handover prediction** (predict UE approaching cell edge)
- **Rate adaptation** (predict UE congestion), predictive **buffering**
- **Beam management** (for mmWave)
- **BS association** (find best BS to connect to)

More applications

- Massive MIMO **CSI compression**
- **Proximity detection** (contact tracing, advertisement)
- **Event labeling** (e.g., link failures)
- Increased **context-awareness** (e.g., pedestrian or in vehicle?)
- **Network planning** (find “difficult” locations)
- **UE grouping** in D2D networks (identify nearby UEs)

Part 8:

Open problems and future directions

Open problems

- **On-line** neural network training (dataset grows continuously)
- **Life-long** learning, improve channel charting without "forgetting" crucial old information
- **Partially supervised** version (some components of $\boldsymbol{y}$ supervised, side information from BS/AP location information)
- Transition of UEs from **outdoors-to-indoors** (and vice versa)
- Robustness to **large-scale changes** in physical channel (weather conditions, presence/absence of trains, ships, etc.)
- Dealing with UEs that perform **transmit beamforming**
- **Data fusion** with other information sources (e.g., cameras, traffic data, elevator occupancy, etc.)

Practical implementation challenges

In theory, theory and practice are the same...

- How to get CSI out of existing BS/AP architectures?
- How to reduce the amount of CSI without sacrificing accuracy?
- How to train large DNNs efficiently and quickly (unique opportunity for real-time learning and not just inference!)?
- How to properly implement DNNs with complex-valued inputs?
- How to measure performance/success of a given application?
- How to convince regulatory bodies that privacy is no issue?

Conclusions

- Application of ML to the physical layer of communications systems has **unique challenges**:
  - Real time, low latency
  - Large dimensions at fast rates
  - Lack of well-understood feature design
- Dimensionality reduction and manifold learning **relatively unexplored** in communication systems

Channel charting provides pseudo-location in a self-supervised and data-driven manner, which opens up many new application opportunities

- Large potential **impact on network efficiency** (energy consumption, bandwidth, quality-of-service, etc.)

Channel charting resources

Channel Charting Resources

locating user equipment within the radio environment using channel state information

What is Channel Charting?

Channel charting learns a mapping from channel state information (CSI) to a so-called **channel chart** in which nearby datapoints indicate nearness in real space. In other words, the learned channel chart captures the spatial geometry of the transmitting user equipments (UEs), effectively encoding relative (or logical) UE locations. Channel charting is self-supervised as the mapping from CSI to the channel chart is learned only using a database of passively collected CSI information. Such a data-driven localization approach has the advantages of being scalable and avoiding reference location information, e.g., from global navigation satellite systems (GNSSs). The self-supervised nature of channel charting also avoids the need for line-of-sight (LoS) propagation conditions or (costly) measurement campaigns, while enabling the infrastructure basestations or access points to perform cognitive and predictive radio access network (RAN) tasks which are tied to UE location.

channelcharting.github.io

- Up-to-date and complete list of channel charting papers
- Channel charting source code
- More to come...

[1] C. Studer, S. Medjkouh, E. Gönültaş, T. Goldstein, and O. Tirkkonen, "Channel Charting: Locating Users Within the Radio Environment Using Channel State Information," IEEE Access, Aug. 2018

[2] J. Deng, S. Medjkouh, N. Malm, O. Tirkkonen, and C. Studer, "Multipoint Channel Charting for Wireless Networks," 52nd Asilomar Conf. on Signals, Systems, and Computers, Oct. 2018

[3] E. Lei, O. Castañeda, O. Tirkkonen, T. Goldstein, and C. Studer, "Siamese Neural Networks for Wireless Positioning and Channel Charting," 57th Annual Allerton Conf. on Communication, Control, and Computing (Allerton), Sep. 2019

[4] P. Ferrand, A. Decurninge, L. Garcia Ordoñez and M. Guillaud, "Triplet-Based Wireless Channel Charting: Architecture and Experiments," IEEE J. on Selected Areas in Communications, Aug. 2021.