Multi-Microphone Speaker Localization on Manifolds

Bracha Laufer-Goldshtein[1] **Ronen Talmon**[2] **Sharon Gannot**[1]

[1]Bar-Ilan University, Ramat-Gan, Israel

[2]Technion – Israel Institute of Technology, Haifa, Israel

EUSIPCO, A Coruña, Spain, September 2, 2019

Bracha Laufer-Goldshtein

Ronen Talmon

Sharon Gannot

Slides available at:

www.eng.biu.ac.il/gannot/tutorials-and-keynote-addresses

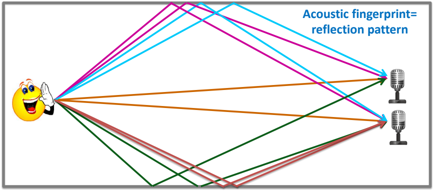
Acoustic Source Localization & Tracking

Goal

Locate/track a sound source(s) given a set of microphone signals in acoustic environment

Environment-aware data-driven acoustic source localization

- Based on fingerprints in acoustic enclosures
- Exploiting the availability of multiple microphones in ad hoc networks of low-end devices
- Utilizing the power of modern data-driven paradigms



Applications

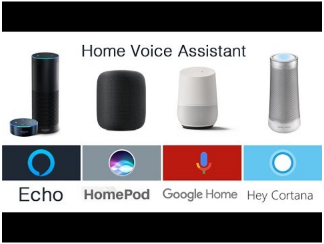
An Essential component in Speech Processing Applications

1. Hands-free voice communication
2. Human-car communication
3. Camera steering
4. Robot audition
5. Smart homes and smart conference call systems
6. Assistive devices for the elderly ("Aging in Place")
7. Smart speakers, e.g. Amazon Echo, Google Home and Apple HomePod
8. Personal assistant, e.g. Apple Siri, Cortana Microsoft and Google Assistant
9. Hearing aids
10. Hearables (wireless earbuds, augmented hearing)

Why Localization?

Smart speakers as an example

- Construct a direct-path steering vector for speech enhancement
- Determine the speakers in the scene and their role
- Carry out location specific tasks (switch the lights on, steer a camera, etc.)



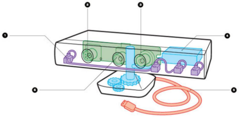
Many Microphones are Available

Devices equipped with multiple microphones

1. Cellular phones
2. Laptops and tablets
3. Hearing devices
4. Smart watches
5. Smart glasses
6. Smart homes & cars







Speaker Localization and Tracking

Basics and Prior Art

- The target of localization (or tracking) algorithms can be either the coordinates of the speaker, or the time difference of arrival (TDOA) between microphone signals
- The mathematical relations between the coordinates of the speakers (or the respective TDOAs) and the observed signals is nonlinear and non-injective
- Localization approaches can be roughly split into two groups:
  - Single-step approaches: The location of the source is estimated directly from the microphone signals
  - Dual-step approaches: TDOAs between pairs of microphone are first estimated, and are subsequently merged to obtain the source coordinates by intersecting geometric surfaces

Speaker Localization and Tracking (cont.)

Basics and Prior Art

- Dynamic scenarios further complicates the problem, as smoothness of the speaker trajectory should be kept
- Multiple concurrent speakers scenarios are even more challenging, due to mixing between the reflections of all speakers (in this tutorial, results of an ongoing research in this domain will not be presented)
- Classical localization methods are usually ignoring the richness of the acoustic propagation path
- In this tutorial, we will present a family of localization and tracking methods that
  - Directly utilize the properties of the acoustic propagation of sound in a given environment
  - Harness data-driven paradigms to extract relevant information from the large amount of available data

Speaker Localization and Tracking (cont.)

Basics and Prior Art

Single-step

- MUSIC [Schmidt, 1986]; used as a baseline for LOCATA challenge [Löllmann et al., 2018]
- ESPRIT [Roy and Kailath, 1989]; applied to speech signals (e.g. [Teutsch and Kellermann, 2005]) or as features for subsequent spatial processing (e.g. [Thiergart et al., 2014])
- Steered-response beamformer phase transform (SRP-PHAT) [DiBiase et al., 2001, Do et al., 2007]; can also be used as features for subsequent spatial processing (e.g. [Madhu and Martin, 2018, Hadad and Gannot, 2018])
- Maximum-Likelihood (e.g. [Yao et al., 2002])

Speaker Localization and Tracking (cont.)

Basics and Prior Art

TDOA estimation and tracking

- Generalized cross-correlation (GCC) [Knapp and Carter, 1976]
- Subspace methods [Benesty, 2000, Doclo and Moonen, 2003]
- Relative transfer function (RTF)-based [Dvorkind and Gannot, 2005]

Geometric intersections

- Linear intersections [Brandstein et al., 1997]
- Spherical intersections [Schau and Robinson, 1987]
- Spherical interpolation [Smith and Abel, 1987]
- One-step least squares (OSLS) [Huang et al., 2000]
- Linear-correction least-squares [Huang et al., 2001]

Speaker Localization and Tracking (cont.)

Basics and Prior Art

Bayesian

- Extended, Unscented and Iterated-Extended Kalman filter
  [Gannot and Dvorkind, 2006, Faubel et al., 2009, Klee et al., 2006]
- Particle filters (PF), Rao-Blackwellised Monte-Carlo
  [Ward et al., 2003, Lehmann and Williamson, 2006, Zhong and Hopgood, 2008, Levy et al., 2011]
- Variational Bayes [Ban et al., 2019, Soussana and Gannot, 2019]
- Probability hypothesis density (PHD) filters [Evers and Naylor, 2017]
- Viterbi algorithm for Hidden Markov model (HMM) [Roman et al., 2003]

Speaker Localization and Tracking (cont.)

Basics and Prior Art

Non-Bayesian

- Mixture of Gaussians (MoG) clustering of SRP outputs with expectation-maximization (EM) [Madhu et al., 2008]; using binaural cues and MoG clustering with predefined grid positions as Gaussian centroids [Mandel et al., 2007, Mandel et al., 2010]; using mixture of von Mises distribution [Brendel et al., 2018]
- RANdom SAmple Consensus (RANSAC) and EM [Traa and Smaragdis, 2014]
- Recursive [Schwartz and Gannot, 2013] and distributed [Dorfan and Gannot, 2015, Dorfan et al., 2018] EM MoG clustering with predefined grid positions as Gaussian centroids
- EM with spectrogram clustering [Dorfan et al., 2016, Schwartz et al., 2017, Weisberg et al., 2019]

Speaker Localization and Tracking (cont.)

Basics and Prior Art

Learning-based methods

- Probabilistic piecewise affine mapping based on smooth binaural manifolds of low dimensions

  [Deleforge and Horaud, 2012, Deleforge et al., 2013, Deleforge et al., 2015]
- MoG clustering of binaural cues using multi-condition training

  [May et al., 2011]
- Gaussian processes inference to map coherent-to-diffuse power ratio and source distance [Brendel and Kellermann, 2019]
- Deep learning for classifying feature vectors to candidate positions: Fully connected [Xiao et al., 2015]; convolutional neural networks (CNN) [Takeda and Komatani, 2016, Chakrabarty and Habets, 2019], convolutional recurrent neural network (CRNN) [Adavanne et al., 2018, Perotin et al., 2019]
- Deep ranking using triplet loss [Opochinsky et al., 2019]

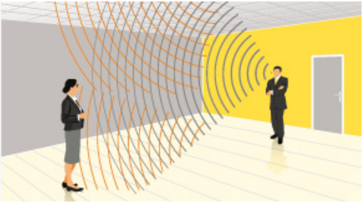
Our Proposed Methodology

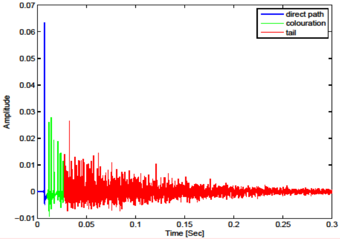
- Utilizes the reflection pattern of the acoustic propagation
- Harnesses the power of machine learning (specifically, manifold learning) to deal with the complexity of the acoustic propagation
- Is suitable for both coordinate localizing and TDOA estimation, depending on the number of nodes used
- Can be also used in dynamic scenarios

Room Acoustics Essentials

Acoustic propagation models

- When sound propagates in an enclosure it undergoes reflections from its surfaces
- Reflections can be modeled as images beyond room walls and hence impinging the microphones from many directions [Allen and Berkley, 1979, Peterson, 1986]
- Statistical models for late reflections [Polack, 1993, Schroeder, 1996, Jot et al., 1997]
- Late reflections tend to be diffused, hence do not exhibit directionality [Dal Degan and Prati, 1988, Habets and Gannot, 2007]





Describing the wave propagation of an audio source in an arbitrary acoustic environment is a cumbersome task

How to Utilize the Intricate Reflection Pattern?

- Classical multi-microphone speech processing algorithms, and specifically acoustic source localization, model the acoustic propagation as time difference of arrival (TDOA)-only, while ignoring sound reflections and focusing only on the-direct path
- It was shown [Gannot et al., 2001, Markovich et al., 2009] that utilizing the entire acoustic propagation path, manifested by the acoustic impulse response (AIR), may significantly improve the performance of speech processing algorithms
- We will show that the intricate acoustic reflection patterns define a fingerprint, uniquely characterizing the source location in the enclosure

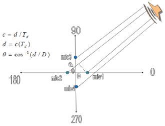
How to Model the Acoustic Environment?

How to model the acoustic path?

Classical speech processing

TDOA-only

Too simplified



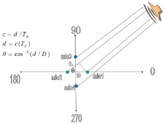
How to Model the Acoustic Environment?

How to model the acoustic path?

Classical speech processing

TDOA-only

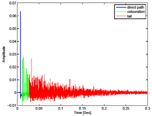
Too simplified



To cope with reverberations

Entire Impulse Response

Perfect but Complicated: No simple Model



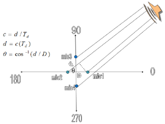
How to Model the Acoustic Environment?

How to model the acoustic path?

Classical speech processing

TDOA-only

Too simplified



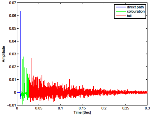
Manifold Learning

Simple to describe and to implement

To cope with reverberations

Entire Impulse Response

Perfect but Complicated: No simple Model



How to Harness Manifold Learning to Infer Source Location from Acoustic Relection Pattern?

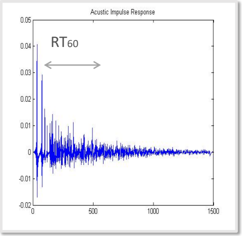
- As shown above, describing the wave propagation of an audio source in an arbitrary acoustic environment is, a cumbersome task, since:
  - No simple mathematical models exist
  - The estimation of the vast number of parameters used to describe the wave propagation suffers from large errors
- We will show that the collection of acoustic fingerprints pertain to a low-dimensional acoustic manifold:
  - The intrinsic degrees of freedom (DoF) in acoustic responses are limited to a small number of variables (e.g. room dimensions, source and microphone positions, and refection coefficients)
  - In a fixed environment and microphone constellation, the acoustic responses intrinsically differ only by the source position

How to Harness Manifold Learning to Infer Source Location from Acoustic Relection Pattern? (cont.)

Manifold learning: A data-driven approach

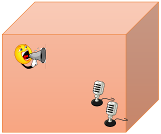
→ Extracts the geometrical structure of the acoustic fingerprints

→ Can reveal the controlling DoFs and hence improve localization ability

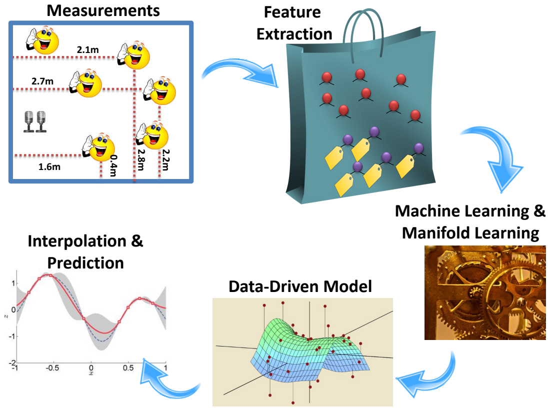


Controlling Parameters

- Room dimensions
- Reverberation time
- Microphone position
- …
- Source position



The Data Processing Pipeline



- Data pre-processing and feature extraction
- Analyzing the geometric structure of the data (manifold learning)
- Deriving data-driven algorithms and inference methodologies to perform a certain task (in our case, localizing the source)

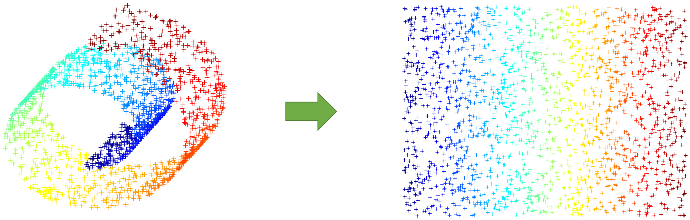
Outline

Outline

Data Representation

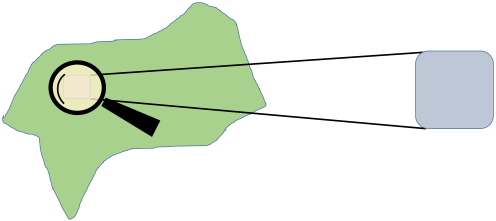
- Measured data often exhibit highly redundant representations
- Often controlled by a small set of parameters
- Lie on a low dimensional manifold

- Consider n high-dimensional features $\mathbf{h}_i \in \mathbb{R}^D$ extracted from the data
- Construct a low-dimensional representation $\mathbf{y}_i \in \mathbb{R}^d$ of $\mathbf{h}_i$, $d < D$, respecting the manifold geometric structure



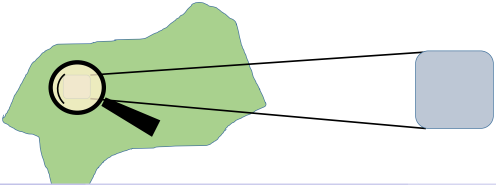
What is a manifold?

- A topological space in which every local region is isomorphic to a Euclidean space
- Differential manifold: a manifold that is locally similar to a linear space
- Riemannian manifold: a differential manifold equipped with an inner product (metric) defined on the tangent plane to the manifold at every point



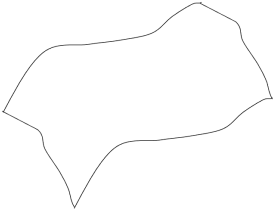
Laplacian

- The Laplacian Δ is an operator defined by the divergence of the gradient of a function in a Euclidean space: $\Delta = \nabla \cdot \nabla$
- The Laplace–Beltrami operator $\mathcal{L}$ is the extension to Riemannian manifolds
- It was shown [Bérard et al., 1994] that a local coordinate system can be built using the Laplacian of the manifold
  $\Rightarrow$ The Laplacian contains all the information about the manifold geometry
- The Laplacian describes the evolution in time of a diffusion process (heat equation)



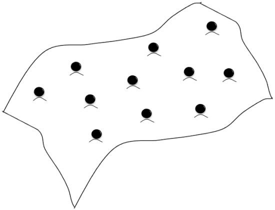
Discretization of the Manifold

- The Laplacian is an infinite-dimension operator defined on continuous spaces



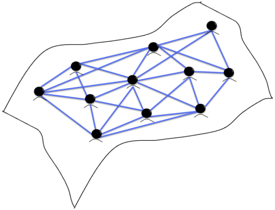
Discretization of the Manifold

- The Laplacian is an infinite-dimension operator defined on continuous spaces
  - We are typically given a finite set of observations in discrete spaces
  - What is the finite-dimension counterpart of the Laplacian?



Discretization of the Manifold

- The Laplacian is an infinite-dimension operator defined on continuous spaces
  - We are typically given a finite set of observations in discrete spaces
  - What is the finite-dimension counterpart of the Laplacian?
- The manifold can be empirically represented by a graph
  - The observations are the graph nodes
  - Define a finite operator (matrix) – the graph Laplacian



Manifold Learning Paradigms

Why learning?

- Given high-dimensional point clouds
- Recall: assume they lie on a manifold, but no other prior knowledge
- The goal is to recover the manifold from the data

Classical methods

- The foundations of manifold learning were laid in 2000:
  - Locally linear embedding (LLE) [Roweis and Saul, 2000]
  - Isometric feature mapping (ISOMAP) [Tenenbaum et al., 2000]
- We will focus on diffusion maps due to the notion of diffusion distance [Coifman and Lafon, 2006]

Locally-Linear Embedding [Roweis and Saul, 2000]

- Determine the neighbours $\mathcal{N}_i$ of each point $\mathbf{h}_i$
- Compute the weights that best reconstruct each point from its neighbors by minimizing:

$$E(\mathbf{W}) = \sum_i \|\mathbf{h}_i - \sum_{j \in \mathcal{N}_i} W_{ij}\mathbf{h}_j\|^2$$

  such that $\sum_{j \in \mathcal{N}_i} W_{ij} = 1$

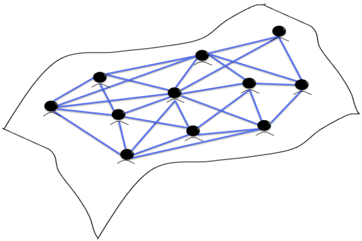
- Compute a low-dimensional embedding $\mathbf{y}_i \in \mathbb{R}^d$ of $\mathbf{h}_i \in \mathbb{R}^D$, $d < D$:

$$\underset{\mathbf{y}_i}{\operatorname{argmin}} \sum_i \|\mathbf{y}_i - \sum_j W_{ij}\mathbf{y}_j\|^2$$

  - $\mathbf{W}$ is an $n \times n$ sparse matrix
  - The embedding can be obtained by solving a sparse eigenvalue problem

ISOMAP [Tenenbaum et al., 2000]

- Determine the neighbours $\mathcal{N}_i$ of each point $\mathbf{h}_i$
- Construct a neighborhood graph:
  - Each point $\mathbf{h}_i$ is a graph node (vertex)
  - Node $\mathbf{h}_i$ is connected by an edge to each neighbor $\mathbf{h}_j \in \mathcal{N}_i$

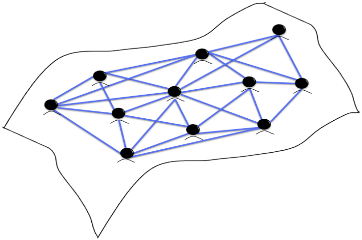


ISOMAP [Tenenbaum et al., 2000]

- Compute the shortest path between any two nodes d_{ij} (number of edges)
- Compute a low-dimensional embedding with multidimensional scaling (MDS) [Kruskal, 1964] by:

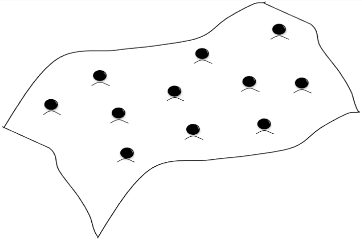
$$\operatorname{argmin}_{\mathbf{y}_1,\ldots,\mathbf{y}_n \in \mathbb{R}^d} \sum_{i<j} (\|\mathbf{y}_i - \mathbf{y}_j\| - d_{ij})^2$$

- Can be solved by eigenvalue decomposition (EVD) of a matrix computed from the pairwise distances $d_{i,j}$



Diffusion Maps [Coifman and Lafon, 2006]

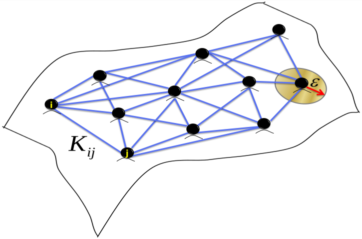
- Samples are the **graph nodes**



Diffusion Maps [Coifman and Lafon, 2006]

- Samples are the graph nodes
- The weights of the edges are defined using a kernel function:

$$K_{ij} = k(\mathbf{h}_i, \mathbf{h}_j) = \exp\left\{-\frac{\|\mathbf{h}_i - \mathbf{h}_j\|^2}{\varepsilon}\right\}$$



Diffusion Maps [Coifman and Lafon, 2006]

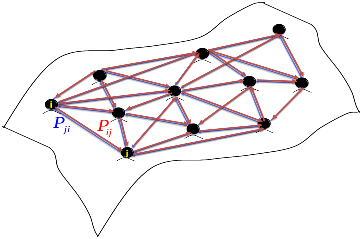
- Samples are the graph nodes
- The weights of the edges are defined using a kernel function:

$$K_{ij} = k(\mathbf{h}_i, \mathbf{h}_j) = \exp\left\{-\frac{\|\mathbf{h}_i - \mathbf{h}_j\|^2}{\varepsilon}\right\}$$

- Define a Markov process on the graph by the transition matrix:

$$P_{ij} = p(\mathbf{h}_i, \mathbf{h}_j) = K_{ij} / \sum_{r=1}^{N} K_{ir}$$

which is a discretization of a diffusion process on the manifold



Diffusion Maps [Coifman and Lafon, 2006]

- Samples are the graph nodes
- The weights of the edges are defined using a kernel function:
$$K_{ij} = k(\mathbf{h}_i, \mathbf{h}_j) = \exp\left\{-\frac{\|\mathbf{h}_i - \mathbf{h}_j\|^2}{\varepsilon}\right\}$$
- Define a Markov process on the graph by the transition matrix:
$$P_{ij} = p(\mathbf{h}_i, \mathbf{h}_j) = K_{ij} / \sum_{r=1}^{N} K_{ir}$$
which is a discretization of a diffusion process on the manifold

- In matrix form: $\mathbf{P} = \mathbf{D}^{-1}\mathbf{K} \in \mathbb{R}^{n \times n}$
where $\mathbf{D}$ is diagonal with:
$$D_{ii} = \sum_{r=1}^{n} K_{ir}$$
- $\mathbf{P}$ is similar to a symmetric matrix $\mathbf{S} = \mathbf{D}^{-1/2}\mathbf{K}\mathbf{D}^{-1/2}$ by
$$\mathbf{P} = \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{1/2}$$
so $\mathbf{P}$ has a real spectrum

Diffusion Maps [Coifman and Lafon, 2006]

- The (normalized) graph Laplacian is defined by

$$\mathbf{N} = \mathbf{I} - \mathbf{P}$$

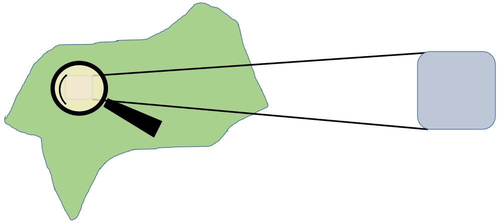
- It was shown that **N** asymptotically ($\varepsilon \to 0$ $n \to \infty$) converges to the Laplacian $\mathcal{L}$
  $\Rightarrow$ The normalized graph Laplacian **N** (and **P**) contains the information about the manifold geometry

Diffusion Maps [Coifman and Lafon, 2006]

- Apply eigenvalue decomposition (EVD) to the matrix $\mathbf{P} \in \mathbb{R}^{n \times n}$ and obtain n eigenvalues $\{\lambda_j\}$ and n right eigenvectors $\{\varphi_j\}$ in $\mathbb{R}^n$
- A nonlinear mapping into a new d-dimensional Euclidean space:

$$\mathbf{\Phi}_d : \mathbf{h}_i \mapsto \left[\lambda_1 \varphi_1(i), \ldots, \lambda_d \varphi_d(i)\right]^T$$

where $d < n$ is typically set by prior knowledge or according to a "spectral gap"



Q: In what sense the space is Euclidean?

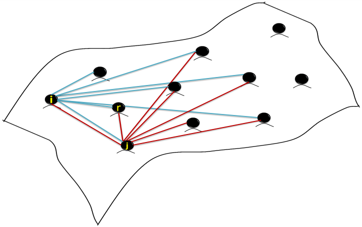
Diffusion Distance

The distance along the manifold is approximated by the diffusion distance:

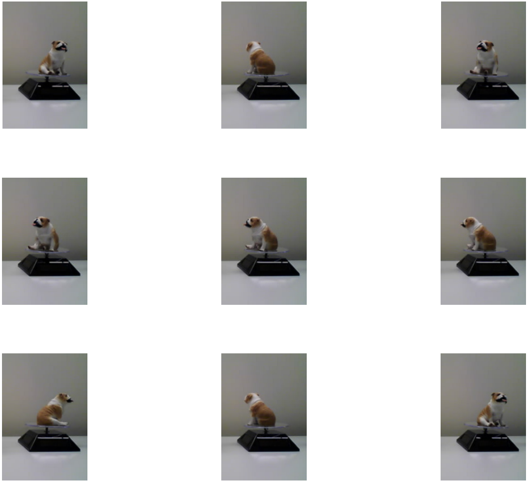
$$D^2_{\mathrm{Diff}}(\mathbf{h}_i, \mathbf{h}_j) = \sum_{r=1}^{n} \left(p\left(\mathbf{h}_i, \mathbf{h}_r\right) - p\left(\mathbf{h}_j, \mathbf{h}_r\right)\right)^2 / \phi_0^{(r)}$$

- Two points are close if they are highly connected in the graph
- The diffusion distance can be well approximated by the Euclidean distance in the embedded domain:

$$D_{\mathrm{Diff}}(\mathbf{h}_i, \mathbf{h}_j) \cong \|\boldsymbol{\Phi}_d(\mathbf{h}_i) - \boldsymbol{\Phi}_d(\mathbf{h}_j)\|$$



Toy Example [Lederman and Talmon, 2018]



Building the Embedding

Diffusion maps

- Compute $\mathbf{P}$ (or equivalently $\mathbf{N}$) from the images $\mathbf{h}_i$
- Apply EVD to $\mathbf{P}$ (or $\mathbf{N}$) and obtain eigenvalues $\{\lambda_j\}$ and eigenvectors $\{\varphi_j\}$
- Build the map:

$$\mathbf{h}_i \mapsto [\lambda_1 \varphi_1(i),\ \lambda_2 \varphi_2(i)]$$

Geometry of Data

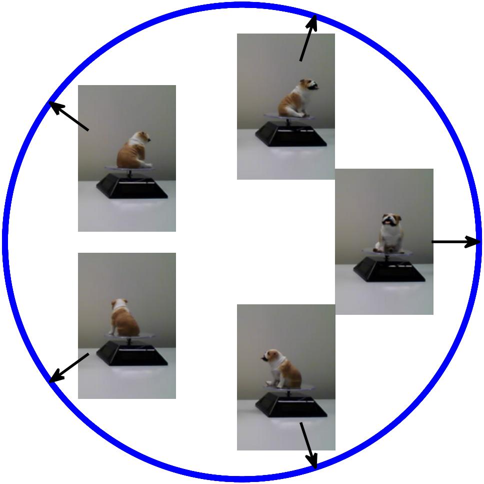


Figure: Each sample (snapshot) is a point on the circle (the rotation angle)

Geometry of Data

Video: One variable.

Geometry of Data

Video: One variable.

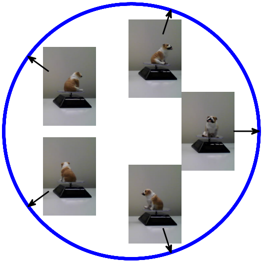
Q: why a circle?

Diffusion

Analogy to the toy example

- The manifold $\mathcal{M}$ is a 1-dimensional sphere in $\mathbb{R}$
- Can be parametrized by $x_i \in [0, 2\pi]$ representing the hidden angle (with periodic boundary conditions)
- We have access to the images $\mathbf{h}_i$, which can be viewed as functions of the hidden angle

$$\mathbf{h}_i := h(x_i)$$



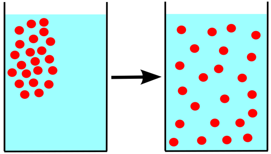
Diffusion

Diffusion process

- The Laplace-Beltrami operator defines a diffusion process on the manifold:

$$u_t = \mathcal{L}u$$

  for a function $u(x, t)$ defined on the manifold, $x \in \mathcal{M}$ and $t \geq 0$
- Suppose $u(x, 0) = u_0(x)$
  $\Rightarrow u(x, t)$ is the propagation of $u_0(x)$ by the application of $\mathcal{L}$



Diffusion

The 1D case

$$u_t = \mathcal{L}u = u_{xx}$$
$$u(x,0) = u_0(x), \forall x \in [0,1]$$
$$u(0,t) = u(1,t), u_x(0,t) = u_x(1,t), \forall t > 0$$

Diffusion

The 1D case

$$u_t = \mathcal{L}u = u_{xx}$$
$$u(x,0) = u_0(x), \forall x \in [0,1]$$
$$u(0,t) = u(1,t), u_x(0,t) = u_x(1,t), \forall t > 0$$

Solution I: separation of variables

$$u(x,t) = X(x)T(t)$$
$$\frac{\dot{T}(t)}{T(t)} = \frac{X''(x)}{X(x)} = -\lambda$$
$$X''(x) = -\lambda X(x)$$

Diffusion

The 1D case

$$u_t = \mathcal{L}u = u_{xx}$$
$$u(x,0) = u_0(x), \forall x \in [0,1]$$
$$u(0,t) = u(1,t), u_x(0,t) = u_x(1,t), \forall t > 0$$

Solution I: separation of variables

$$u(x,t) = X(x)T(t)$$
$$\frac{\dot{T}(t)}{T(t)} = \frac{X''(x)}{X(x)} = -\lambda$$
$$X''(x) = -\lambda X(x)$$

$$X_k(x) = \sin(\sqrt{\lambda_k}x), \cos(\sqrt{\lambda_k}x)$$
$$\lambda_k = 4k^2\pi^2;\ k = 1, 2, \ldots$$

Diffusion

The 1D case

$$u_t = \mathcal{L}u = u_{xx}$$
$$u(x,0) = u_0(x), \forall x \in [0,1]$$
$$u(0,t) = u(1,t), u_x(0,t) = u_x(1,t), \forall t > 0$$

Solution I: separation of variables

$$u(x,t) = X(x)T(t)$$
$$\frac{\dot{T}(t)}{T(t)} = \frac{X''(x)}{X(x)} = -\lambda$$
$$X''(x) = -\lambda X(x)$$

$$X_k(x) = \sin(\sqrt{\lambda_k}x), \cos(\sqrt{\lambda_k}x)$$
$$\lambda_k = 4k^2\pi^2;\; k = 1, 2, \ldots$$

Solution II: EVD [Fourier, 1822]

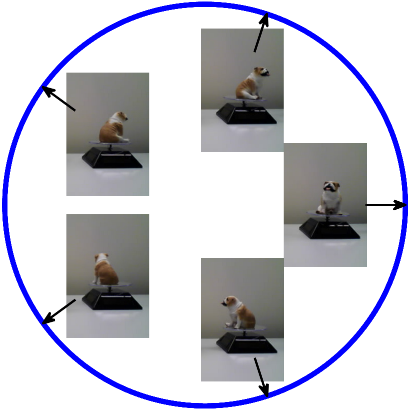
$$\mathcal{L}X(x) = X''(x) = -\lambda X(x)$$

- The eigenvalues and eigenfunctions of $\mathcal{L}$ are λ_k and $X_k(x)$
- $X_k(x)$ describe diffusion and are used for embedding
- Diffusion interprets the embedding

Diffusion

The embedding

$$\mathbf{h}_i \mapsto [4\pi^2 \cos(2\pi x_i), 4\pi^2 \sin(2\pi x_i)]$$



Smoothness on the Manifold

Measuring smoothness over $\mathcal{M}$:

- Let $\mathbf{h} \in \mathcal{M}$ and $f : \mathcal{M} \to \mathbb{R}$
- The gradient $\nabla f(\mathbf{h})$ represents amplitude and direction of variation of f around $\mathbf{h}$
- A global measure of smoothness of f on $\mathcal{M}$:

$$\|f\|_{\mathcal{M}}^2 = \int_{\mathcal{M}} \|\nabla f(\mathbf{h})\|^2 d\mu(\mathbf{h})$$

where $\mu(\mathbf{h})$ is the probability measure of $\mathbf{h}$ on $\mathcal{M}$

Smoothness on the Manifold

Measuring smoothness on $\mathcal{M}$:

- Stokes' theorem links gradient and Laplacian:

$$\int_{\mathcal{M}} \|\nabla f(\mathbf{h})\|^2 d\mu(\mathbf{h}) = \int_{\mathcal{M}} f(\mathbf{h}) \mathcal{L} f(\mathbf{h}) d\mu(\mathbf{h}) = \langle f(\mathbf{h}), \mathcal{L} f(\mathbf{h}) \rangle$$

where $\mathcal{L} = \nabla \cdot \nabla$ is the Laplace-Beltrami ("Laplacian") operator

Smoothness on the Manifold

Measuring smoothness on $\mathcal{M}$:

- Stokes' theorem links gradient and Laplacian:

$$\int_{\mathcal{M}} \|\nabla f(\mathbf{h})\|^2 d\mu(\mathbf{h}) = \int_{\mathcal{M}} f(\mathbf{h}) \mathcal{L} f(\mathbf{h}) d\mu(\mathbf{h}) = \langle f(\mathbf{h}), \mathcal{L} f(\mathbf{h}) \rangle$$

where $\mathcal{L} = \nabla \cdot \nabla$ is the Laplace-Beltrami ("Laplacian") operator

Smoothness on the manifold: Discretization

- Define the graph Laplacian: $\mathbf{L} \triangleq \mathbf{D} - \mathbf{K}$
- $\mathbf{P} = \mathbf{D}^{-1}\mathbf{K}$ and $\mathbf{N} = \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{P}$
- Smoothness of $\mathbf{f} = [f(\mathbf{h}_1), ..., f(\mathbf{h}_n)]$ on the graph: $\mathbf{f}^T\mathbf{L}\mathbf{f} = \langle \mathbf{f}, \mathbf{L}\mathbf{f} \rangle$
- Small $\mathbf{f}^T\mathbf{L}\mathbf{f}$ implies smooth $\mathbf{f}$ on the graph

Smoothness on the Manifold: Discretization

- Further insight can be obtained by:

$$\begin{aligned}
\mathbf{f}^T \mathbf{L} \mathbf{f} &= \sum_{i,j=1}^{n} f(\mathbf{h}_i) L_{ij} f(\mathbf{h}_j) \\
&= \sum_{i=1}^{n} \left(\sum_{j=1}^{n} K_{ij} - K_{ii} \right) f^2(\mathbf{h}_i) - \sum_{\substack{i,j=1 \\ i \neq j}}^{n} K_{ij} f(\mathbf{h}_i) f(\mathbf{h}_j) \\
&= \sum_{i,j=1}^{n} K_{ij} f^2(\mathbf{h}_i) - \sum_{i,j=1}^{n} K_{ij} f(\mathbf{h}_i) f(\mathbf{h}_j) \\
&= \frac{1}{2} \sum_{i,j=1}^{n} K_{ij} \left(f(\mathbf{h}_i) - f(\mathbf{h}_j) \right)^2
\end{aligned}$$

- When K_{ij} is large, the mappings $f(\mathbf{h}_i)$ and $f(\mathbf{h}_j)$ are "encouraged" to be close

Further Insight

Eigenvalue decomposition of the Laplacian

- Recall: $\mathbf{L}$ is the symmetric graph Laplacian with eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ and corresponding eigenvectors $\varphi_1, \ldots, \varphi_n$
- By the Courant-Fischer Theorem:

$$\lambda_k = \min_{\mathbf{f} \perp \varphi_1, \ldots, \varphi_{k-1}} \frac{\mathbf{f}^T \mathbf{L} \mathbf{f}}{\mathbf{f}^T \mathbf{D} \mathbf{f}}$$

$$\varphi_k = \operatorname*{argmin}_{\mathbf{f} \perp \varphi_1, \ldots, \varphi_{k-1}} \frac{\mathbf{f}^T \mathbf{L} \mathbf{f}}{\mathbf{f}^T \mathbf{D} \mathbf{f}}$$

Analogy to the Fourier transform:

- Small eigenvalues correspond to eigenvectors that change slowly on the manifold ("low frequencies")
- Large eigenvalues correspond to eigenvectors that change rapidly on the manifold ("high frequencies")

Laplacian Eigenmaps [Belkin and Niyogi, 2003]

Building low-dimensional embedding

- Similarly to Diffusion Maps:

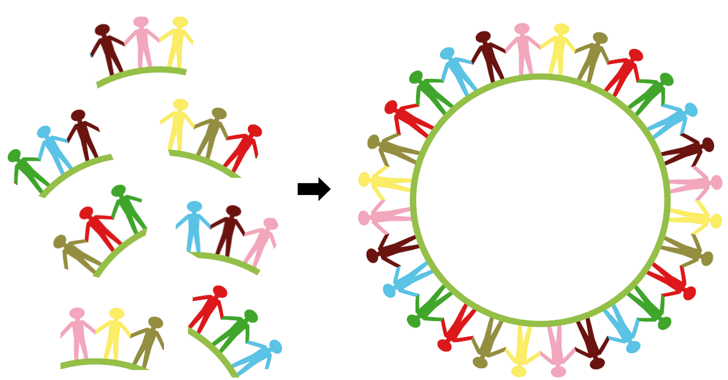
$$\mathbf{h}_i \mapsto [\varphi_1(i), \ldots, \varphi_d(i)]^T$$

- As shown above, the Euclidean distance between embedded points respects the similarity defined by the kernel
  - High kernel affinity leads to nearby embedded points

Manifold Learning – Summary

"Tell me who your friends are and I will tell you who you are"

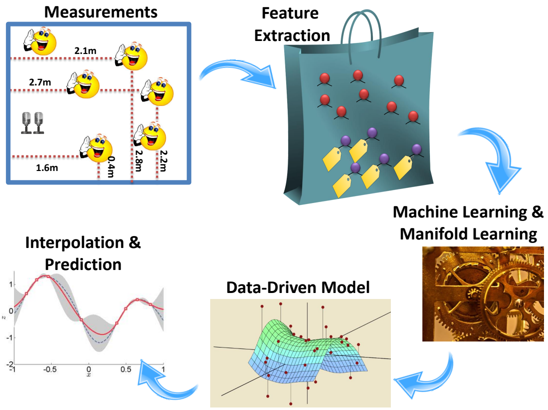
- In high-dimensional space only local relations are meaningful
- Find a global fit that preserves local relations:
  - Local relations by kernel function similarity
  - Global fit by spectral decomposition



Outline

The Data Processing Pipeline

Back to Speaker Localization



- Data pre-processing and feature extraction
- Analyzing the geometric structure of the data (manifold learning)
- Deriving data-driven algorithms and inference methodologies to perform a certain task (in our case, localizing the source)

Data Model: The Two Microphone Case

Microphone signals:

The measured signals in the two microphones (an extension to multiple microphone pairs will be discussed later):

$$y_1(n) = a_1(n) * s(n) + u_1(n)$$
$$y_2(n) = a_2(n) * s(n) + u_2(n)$$

- $s(n)$ - the source signal
- $a_i(n),\ i = \{1, 2\}$ - the acoustic impulse responses relating the source and each of the microphones
- $u_i(n),\ i = \{1, 2\}$ - noise signals, independent of the source

Data Model: The Two Microphone Case

Microphone signals:

The measured signals in the two microphones:

$$y_1(n) = a_1(n) * s(n) + u_1(n)$$
$$y_2(n) = a_2(n) * s(n) + u_2(n)$$

- $s(n)$ - the source signal
- $a_i(n),\ i = \{1, 2\}$ - the acoustic impulse responses relating the source and each of the microphones
- $u_i(n),\ i = \{1, 2\}$ - noise signals, independent of the source

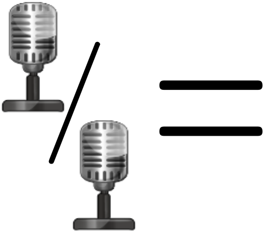
Find a feature vector representing the characteristics of the acoustic path and independent of the source signal

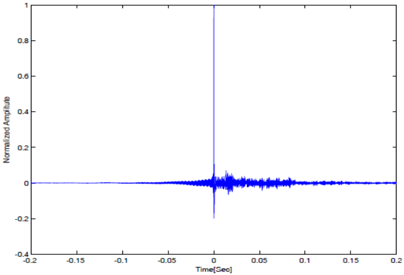
The Features

Alternatives

- The relative transfer function (RTF) for pairs of microphones
  [Gannot et al., 2001]
- Power ratios of directional microphone (using a microphone quartet)
  [Laufer-Goldshtein et al., 2018a]
- Relative harmonic coefficients (using spherical microphone array)
  [Hu et al., 2019]

Relative Transfer Function (RTF) [Gannot et al., 2001]





RTF:

- Defined as the ratio between the transfer functions of the two mics:

$$H_{12}(k) = \frac{A_2(k)}{A_1(k)} \overset{\text{low-noise}}{\simeq} \frac{\hat{S}_{y_2 y_1}(k)}{\hat{S}_{y_1 y_1}(k)}$$

  estimated based on PSD and cross-PSD
- Define the feature vector: $\mathbf{h} = [\hat{H}_{12}(k_1), \ldots, \hat{H}_{12}(k_D)]^T$

Relative Transfer Function (RTF) [Gannot et al., 2001]

$= f\left(\right.$

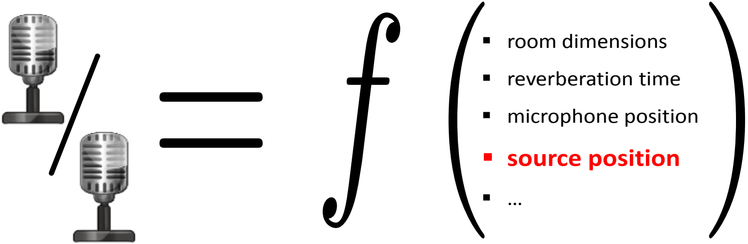
- room dimensions
- reverberation time
- microphone position
- source position
- ...

$\left.\right)$

RTF:

- Represents the acoustic paths and is independent of the source signal
- Generalizes the TDOA
- Depends on a small set of parameters related to the physical characteristics of the environment

Relative Transfer Function (RTF) [Gannot et al., 2001]



RTF:

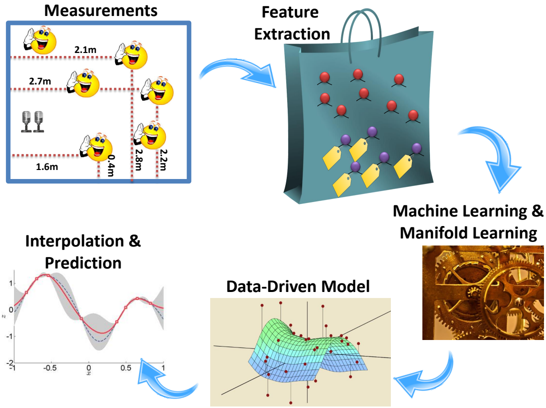
- Represents the acoustic paths and is independent of the source signal
- Generalizes the TDOA
- Depends on a small set of parameters related to the physical characteristics of the environment
- In a static environment the source position is the only varying degree of freedom

A plethora of methods for RTF Estimation

- Utilizing speech non-stationarity and noise stationarity [Shalvi and Weinstein, 1996]; [Gannot et al., 2001]
- Extension to two nonstationary sources in stationary noise [Reuven et al., 2008]
- Subspace tracking for single speaker [Affes and Grenier, 1997]
- GEVD analysis for multiple speakers [Markovich et al., 2009]
- Subspace tracking for multiple speakers [Markovich-Golan et al., 2010]
- Utilizing RIR Sparseness [Koldovský et al., 2015]
- Utilizing BSS methods [Reindl et al., 2013]
- Applying covariance whitening or covariance subtraction [Markovich-Golan et al., 2018]
- Utilizing speech sparsity in the STFT domain (w-disjoint orthogonality [Yilmaz and Rickard, 2004]) and Simplex analysis [Laufer-Goldshtein et al., 2018c]

Outline

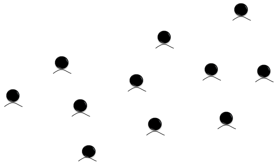
The Data Processing Pipeline



- Data pre-processing and feature extraction
- Analyzing the geometric structure of the data (manifold learning)
- Deriving data-driven algorithms and inference methodologies to perform a certain task (in our case, localizing the source)

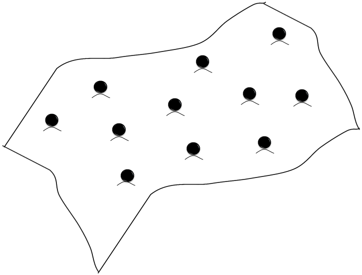
How to Measure the Affinity between Two RTF Samples? [Laufer-Goldshtein et al., 2015]

The RTFs are represented as points in a high dimensional space



How to Measure the Affinity between Two RTF Samples? [Laufer-Goldshtein et al., 2015]

The RTFs are represented as points in a high dimensional space

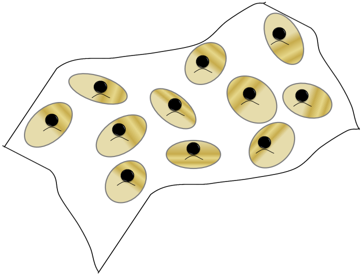


Acoustic manifold

- They lie on a low dimensional nonlinear manifold $\mathcal{M}$

How to Measure the Affinity between Two RTF Samples? [Laufer-Goldshtein et al., 2015]

The RTFs are represented as points in a high dimensional space

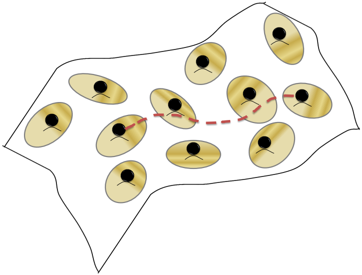


Acoustic manifold

- They lie on a low dimensional nonlinear manifold $\mathcal{M}$
- Linearity is preserved in small neighbourhoods

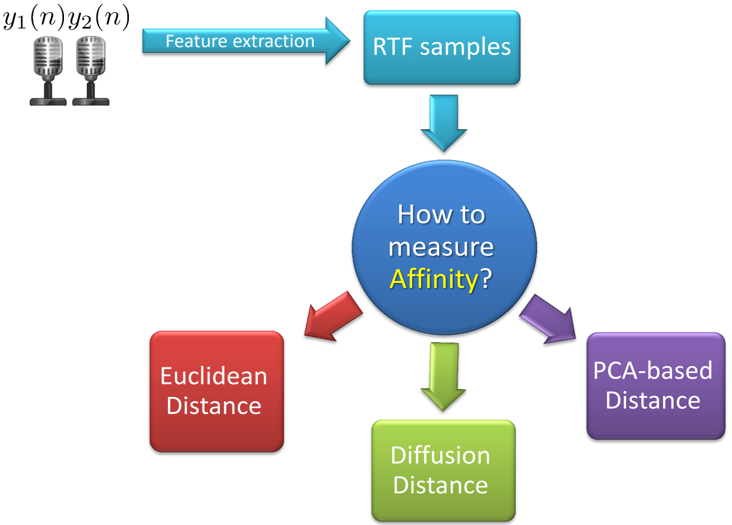
How to Measure the Affinity between Two RTF Samples? [Laufer-Goldshtein et al., 2015]

The RTFs are represented as points in a high dimensional space



Acoustic manifold

- They lie on a low dimensional nonlinear manifold $\mathcal{M}$
- Linearity is preserved in small neighbourhoods
- Distances between RTFs should be measured along the manifold



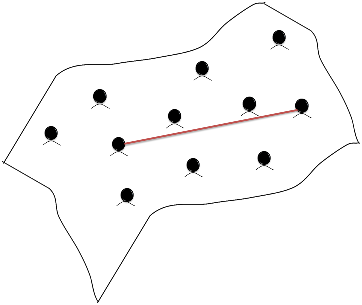
Each distance measure relies on a different hidden assumption about the underlying structure of the RTF samples

Euclidean Distance

The Euclidean distance between RTFs

$$D_{\mathrm{Euc}}(\mathbf{h}_i, \mathbf{h}_j) = \|\mathbf{h}_i - \mathbf{h}_j\|$$

- Compares two RTFs in their original space
- Does not assume an existence of a manifold
- Respects flat manifolds



A good affinity measure only when the RTFs are uniformly scattered all over the space, or when they lie on a flat manifold

Principal component analysis (PCA) [Pearson, 1901]

PCA algorithm

- Find the vectors that maximize the variance of the data:

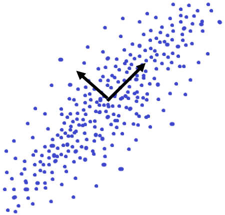
$$\underset{||\mathbf{y}||^2=1}{\operatorname{argmax}} \mathbf{y}^T \hat{\mathbf{R}} \mathbf{y}$$

where $\hat{\mathbf{R}}$ is the sample covariance matrix of the data

- The above maximization problem is solved the EVD of of $\hat{\mathbf{R}}$

Linear vs. nonliner

- PCA - smoothness over sample covariance
- Laplacain Eigenmaps - smoothness over graph Laplacain



PCA-Based Distance

PCA algorithm

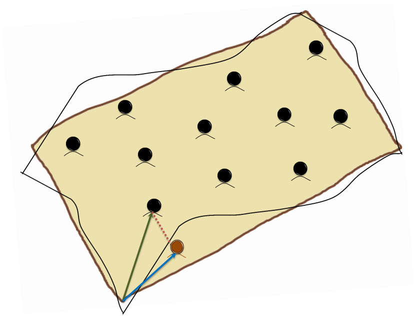
- The principal components - the d dominant eigenvectors $\{\mathbf{v}_i\}_{i=1}^{d}$ of the covariance matrix of the data
- The RTFs are linearly projected onto the principal components:

$$\boldsymbol{\nu}(\mathbf{h}_i) = [\mathbf{v}_1, \ldots \mathbf{v}_d]^T (\mathbf{h}_i - \boldsymbol{\mu})$$

PCA-based distance between RTFs

$$D_{\mathrm{PCA}}(\mathbf{h}_i, \mathbf{h}_j) = \|\boldsymbol{\nu}(\mathbf{h}_i) - \boldsymbol{\nu}(\mathbf{h}_j)\|$$

- A global approach - extracts principal directions of the entire set
- Linear projections - the manifold is assumed to be linear/flat



PCA-Based Distance

PCA algorithm

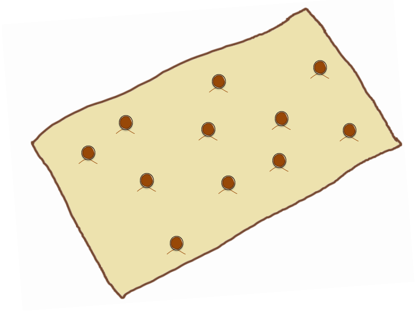
- The principal components - the d dominant eigenvectors $\{\mathbf{v}_i\}_{i=1}^{d}$ of the covariance matrix of the data
- The RTFs are linearly projected onto the principal components:

$$\boldsymbol{\nu}\left(\mathbf{h}_i\right) = \left[\mathbf{v}_1, \ldots \mathbf{v}_d\right]^T \left(\mathbf{h}_i - \boldsymbol{\mu}\right)$$

PCA-based distance between RTFs

$$D_{\mathrm{PCA}}(\mathbf{h}_i, \mathbf{h}_j) = \|\boldsymbol{\nu}(\mathbf{h}_i) - \boldsymbol{\nu}(\mathbf{h}_j)\|$$

- A global approach - extracts principal directions of the entire set
- Linear projections - the manifold is assumed to be linear/flat



PCA-Based Distance

PCA algorithm

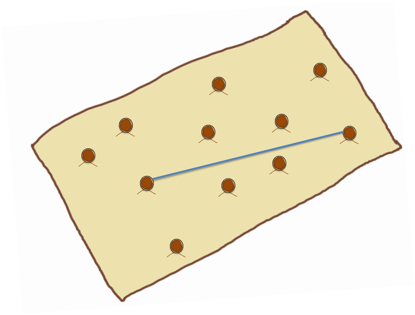
- The principal components - the d dominant eigenvectors $\{\mathbf{v}_i\}_{i=1}^{d}$ of the covariance matrix of the data
- The RTFs are linearly projected onto the principal components:

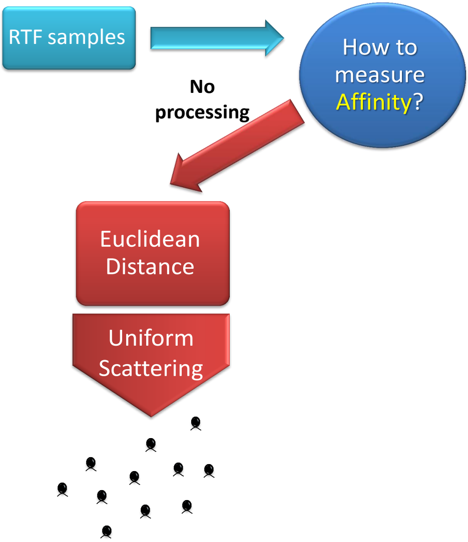
$$\boldsymbol{\nu}(\mathbf{h}_i) = [\mathbf{v}_1, \ldots \mathbf{v}_d]^T (\mathbf{h}_i - \boldsymbol{\mu})$$

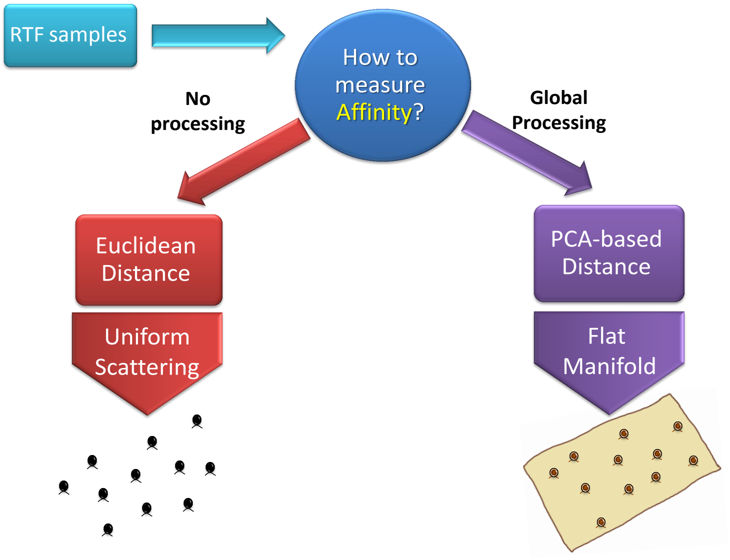
PCA-based distance between RTFs

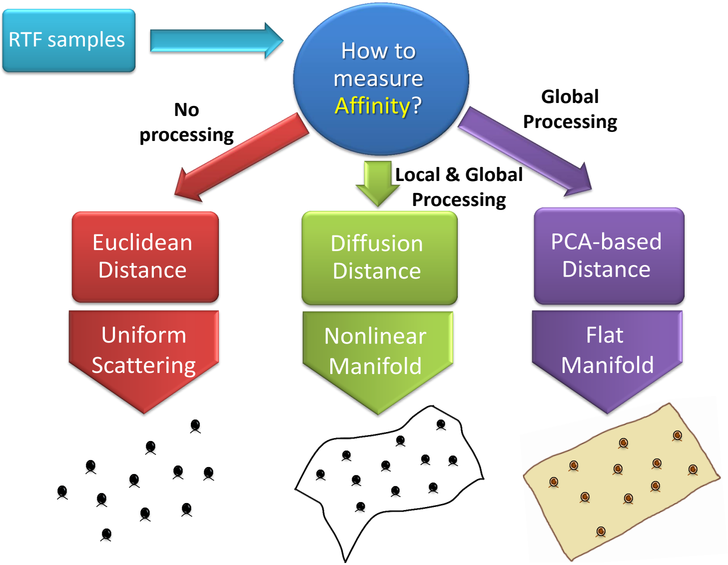
$$D_{\mathrm{PCA}}(\mathbf{h}_i, \mathbf{h}_j) = \|\boldsymbol{\nu}(\mathbf{h}_i) - \boldsymbol{\nu}(\mathbf{h}_j)\|$$

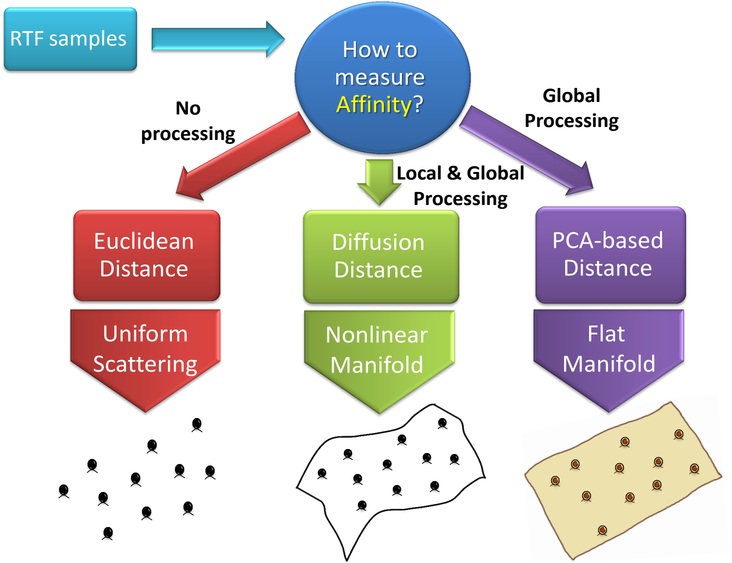
- A global approach - extracts principal directions of the entire set
- Linear projections - the manifold is assumed to be linear/flat




RTF samples
How to measure Affinity?
No processing
Euclidean Distance
Uniform Scattering


RTF samples
How to measure Affinity?
No processing
Global Processing
Euclidean Distance
Uniform Scattering
PCA-based Distance
Flat Manifold


RTF samples
How to measure Affinity?
No processing
Local & Global Processing
Global Processing
Euclidean Distance
Diffusion Distance
PCA-based Distance
Uniform Scattering
Nonlinear Manifold
Flat Manifold



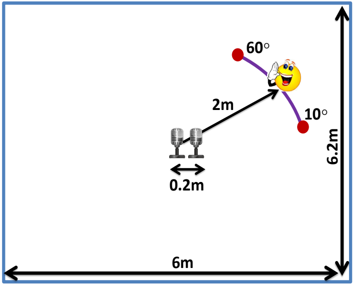
Which of the distance measures is proper?
What is the true underlying structure of the RTFs?

Simulation Results

Room setup

Simulate a reverberant room using the image method [Allen and Berkley, 1979]:

- Room dimension $6 \times 6.2 \times 3$m
- Microphones at: $[3, 3, 1]$ and $[3.2, 3, 1]$
- The source is positioned at 2m from the mics, the azimuth angle in $10^\circ \div 60^\circ$
- $T_{60} = 150/300/500$ ms
- SNR$= 20$ dB

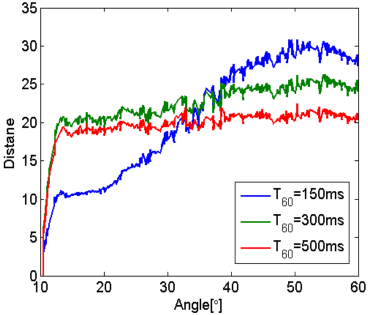


Test

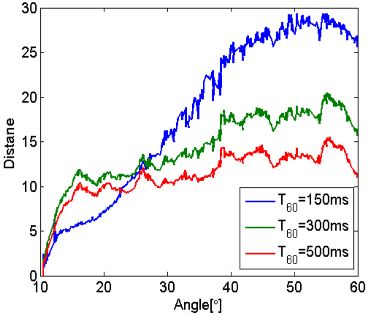
Measure the distance between each of the RTFs and the RTF corresponding to 10°:

- If monotonic with respect to the angle - proper distance
- If not monotonic with respect to the angle - improper distance

Euclidean Distance & PCA-based Distance [Laufer-Goldshtein et al., 2015]



(a) Euclidean Distance

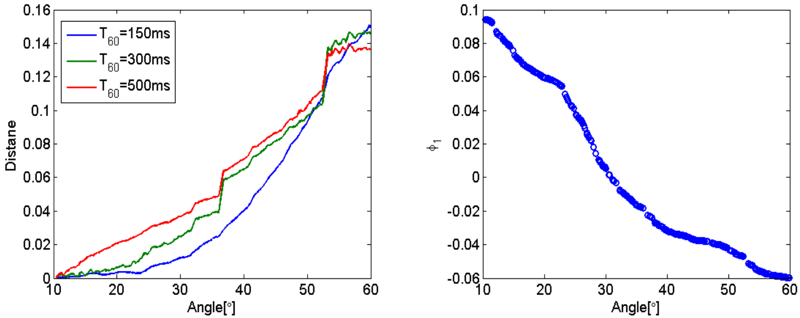


(b) PCA-based Distance

For both distance measures:

- Monotonic with respect to the angle only in a limited region
- This region becomes smaller as the reverberation time increases
- They are inappropriate for measuring angles' proximity

Diffusion Maps



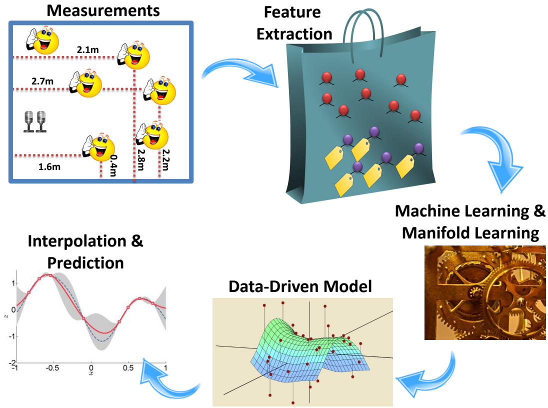
(c) Diffusion Distance (d) Diffusion Mapping

The diffusion distance:

- Monotonic with respect to the angle for almost the entire range
- It is an appropriate distance measure in terms of the source DOA
- Mapping corresponds well with angles - recovers the latent parameter

Outline

The Data Processing Pipeline



- Data pre-processing and feature extraction
- Analyzing the geometric structure of the data (manifold learning)
- Deriving data-driven algorithms and inference methodologies to perform a certain task (in our case, localizing the source)

Semi-Supervised Approaches for Localization

Intermediate summary

- We have established the existence of an acoustic manifold in a specific environment
- The RTF was shown to be a proper feature vector that can capture the acoustic variability as a function of the source position (alternative feature vectors can be used)
- We have briefly introduced the manifold learning - a systematic methodology to infer the low-dimensional intrinsic controlling parameters of the data

Semi-Supervised Approaches for Localization (cont.)

What's next?

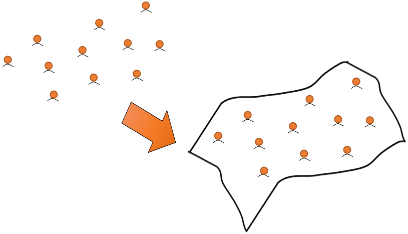
- Learning paradigms:
  1. Unsupervised localization ⇒ array constellation required (microphones positions or microphone inter-distance for DOA-only)
  2. Supervised localization ⇒ many labels
  3. Semi-supervised ⇒ utilizes a small number of labelled data and a large number of unlabelled data; array constellation not required
- Utilize the acoustic manifold to derive two data-driven approaches for speaker localization:
  1. Diffusion Distance Search (DDS) [Talmon et al., 2011, Laufer-Goldshtein et al., 2013]
  2. Manifold Regularization for Localization (MRL) [Laufer-Goldshtein et al., 2016b]

Goal: Recover the function f which transforms an RTF to position

Semi-Supervised Approaches for Localization (cont.)

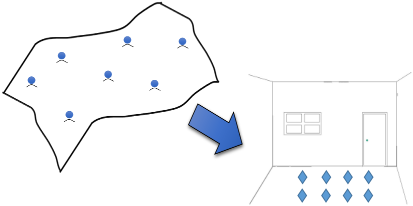
Unlabelled Samples

Recover the Manifold Structure



Labelled Samples

Anchor Points – Translate RTFs to Positions



Semi-Supervised Learning

Mixed of supervised (attached with known locations as anchors) and unsupervised (unknown locations) learning

Semi-Supervised Learning

Mixed of supervised (attached with known locations as anchors) and unsupervised (unknown locations) learning

Why using unlabeled data?

1. Localization - training should fit the specific environment of interest:
   - Cannot generate a general database for all possible acoustic scenarios
   - Generating a large amount of labelled data is cumbersome/impractical
   - Unlabelled data is freely available - whenever someone is speaking

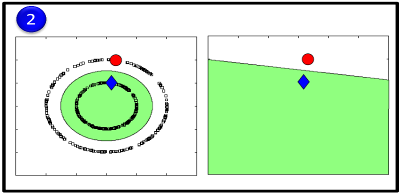
1

Semi-Supervised Learning

Mixed of supervised (attached with known locations as anchors) and unsupervised (unknown locations) learning

Why using unlabeled data?

1. Localization - training should fit the specific environment of interest:
   - Cannot generate a general database for all possible acoustic scenarios
   - Generating a large amount of labelled data is cumbersome/impractical
   - Unlabelled data is freely available - whenever someone is speaking
2. Unlabelled data can be utilize to recover the manifold structure

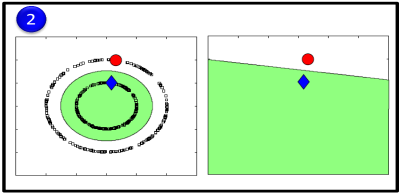


Semi-Supervised Learning

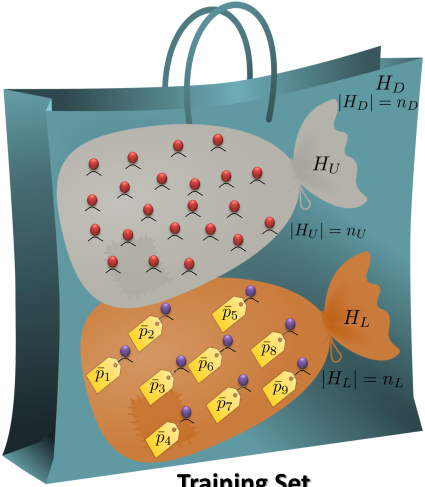
Mixed of supervised (attached with known locations as anchors) and unsupervised (unknown locations) learning

Why using unlabeled data?

1. Localization - training should fit the specific environment of interest:
   - Cannot generate a general database for all possible acoustic scenarios
   - Generating a large amount of labelled data is cumbersome/impractical
   - Unlabelled data is freely available - whenever someone is speaking
2. Unlabelled data can be utilize to recover the manifold structure
3. Semi-supervised learning is the natural setting for human learning

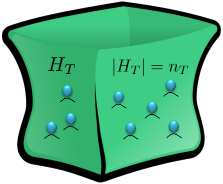


Datasets



Training Set

- $H_L = \{\mathbf{h}_i\}_{i=1}^{n_L}$ - n_L labelled samples
- $P_L = \{\bar{p}_i\}_{i=1}^{n_L}$ - labels/positions
- $H_U = \{\mathbf{h}_i\}_{i=n_L+1}^{n_D}$ - n_U unlabelled samples
- $H_D = H_L \cup H_U$ - entire training set
- $H_T = \{\mathbf{h}_i\}_{i=n_D+1}^{n}$ - n_T test samples



Test Set

Diffusion Distance Search (DDS) [Talmon et al., 2011, Laufer-Goldshtein et al., 2013]

Diffusion mapping: Reminder

- Construct $\mathbf{K}$, and normalize to obtain $\mathbf{P}$
- Employ EVD to obtain $\{\lambda_j, \varphi_j\}$
- Construct the map $\mathbf{\Phi}_d$:

$$\mathbf{\Phi}_d : \mathbf{h}_i \mapsto \left[\lambda_1 \varphi_1^{(i)}, \ldots, \lambda_d \varphi_d^{(i)}\right]^T$$

- Define diffusion distance: $D_{\mathrm{Diff}}\left(\mathbf{h}_{\mathrm{l}}, \mathbf{h}_i\right) = \|\mathbf{\Phi}_d(\mathbf{h}_l) - \mathbf{\Phi}_d(\mathbf{h}_i)\|_2$

What is the diffusion map of a new test point $\mathbf{h}_{\mathrm{t}}$?

- Either recompute the EVD of an $(n_D + 1) \times (n_D + 1)$ matrix $\mathbf{P}$
- Or apply the Nyström extension

Diffusion Distance Search (DDS)

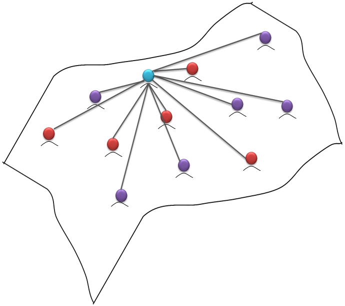
Nyström extension [Press et al., 2007]

- $\lambda_j \boldsymbol{\varphi}_j = \mathbf{P}\boldsymbol{\varphi}_j$, $j \in \{1, \ldots, d\}$
- For $i = 1, \ldots, n_D$:

$$\varphi_j^{(i)} = \frac{1}{\lambda_j} \sum_{l=1}^{n_D} p(\mathbf{h}_i, \mathbf{h}_l) \varphi_j^{(l)}$$

- For a new test point $\mathbf{h}_{\mathrm{t}}$:

$$\varphi_j^{t} = \frac{1}{\lambda_j} \sum_{l=1}^{n_D} p(\mathbf{h}_{\mathrm{t}}, \mathbf{h}_l) \varphi_j^{(l)}$$



Extension of the model for new $\mathbf{h}_{\mathrm{t}}$ (summary):

- Construct a nonsymmetric affinity vector $\mathbf{b}$: $b^{(l)} = p(\mathbf{h}_t, \mathbf{h}_l)$
- Apply Nyström extension:

$$\varphi_j^{t} = \frac{1}{\lambda_j} \mathbf{b}^T \boldsymbol{\varphi}_j \quad j \in \{1, \ldots, d\}$$

Diffusion Distance Search (DDS)

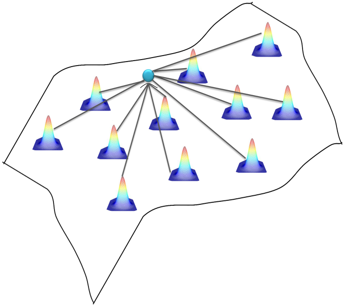
Nyström extension [Press et al., 2007]

- $\lambda_j \boldsymbol{\varphi}_j = \mathbf{P}\boldsymbol{\varphi}_j$, $j \in \{1, \ldots, d\}$
- For $i = 1, \ldots, n_D$:

$$\varphi_j^{(i)} = \frac{1}{\lambda_j} \sum_{l=1}^{n_D} p(\mathbf{h}_i, \mathbf{h}_l) \varphi_j^{(l)}$$

- For a new test point $\mathbf{h}_\mathrm{t}$:

$$\varphi_j^{t} = \frac{1}{\lambda_j} \sum_{l=1}^{n_D} p(\mathbf{h}_\mathrm{t}, \mathbf{h}_l) \varphi_j^{(l)}$$



Extension of the model for new $\mathbf{h}_\mathrm{t}$ (summary):

- Construct a nonsymmetric affinity vector $\mathbf{b}$: $b^{(l)} = p(\mathbf{h}_t, \mathbf{h}_l)$
- Apply Nyström extension:

$$\varphi_j^{t} = \frac{1}{\lambda_j} \mathbf{b}^T \boldsymbol{\varphi}_j \quad j \in \{1, \ldots, d\}$$

Diffusion Distance Search (DDS)

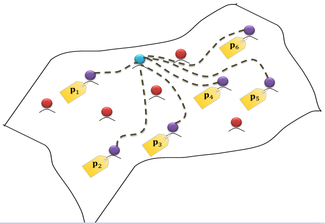
Localization:

Heuristic estimation: a linear combination of the labelled set positions according to kernelized diffusion distances:

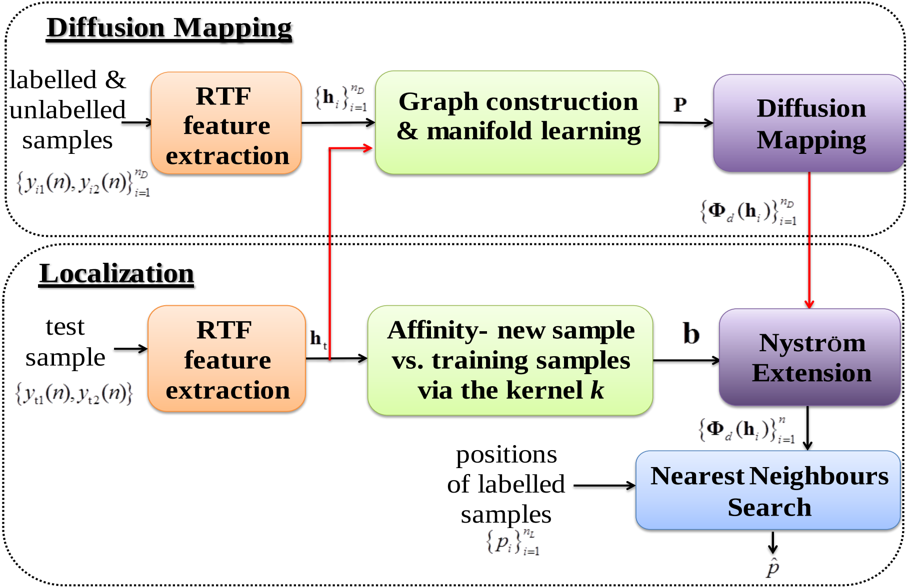
$$\hat{p}(\mathbf{h}_t) = \sum_{i=1}^{n_L} \gamma\left(\mathbf{h}_i\right) p_i$$

where the weights $\gamma\left(\mathbf{h}_i\right)$ are given by:

$$\gamma\left(\mathbf{h}_i\right) = \frac{\exp\left\{-D_{\mathrm{Diff}}\left(\mathbf{h}_{\mathrm{t}}, \mathbf{h}_i\right)/\varepsilon_\gamma\right\}}{\sum_{j=1}^{l} \exp\left\{-D_{\mathrm{Diff}}\left(\mathbf{h}_{\mathrm{t}}, \mathbf{h}_j\right)/\varepsilon_\gamma\right\}}$$

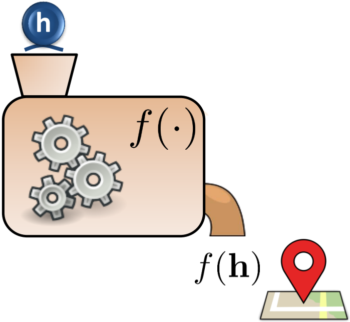


Diffusion Distance Search (DDS)



Manifold Regularization for Localization [Laufer-Goldshtein et al., 2016b]

Goal: Recover the function f which transforms an RTF to position



Manifold Regularization for Localization [Laufer-Goldshtein et al., 2016b]

Goal: Recover the function f which transforms an RTF to position

Complex nonlinear relation between RTFs and positions

Infinite search space

How to prevent overfitting?

How to utilize unlabelled data?

Manifold Regularization for Localization [Laufer-Goldshtein et al., 2016b]

Goal: Recover the function f which transforms an RTF to position

Complex nonlinear relation between RTFs and positions

- Learn a data-driven model from training data

Infinite search space

- Work in a reproducing kernel Hilbert space (RKHS)

How to prevent overfitting?

- Add regularizations to control smoothness

How to utilize unlabelled data?

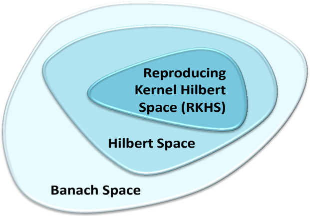
- Use manifold regularization

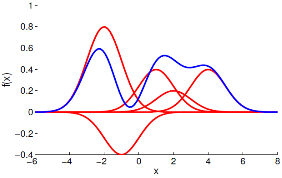
Reproducing Kernel Hilbert Space (RKHS) [Berlinet and Thomas-Agnan, 2011]

Moore-Aronszajn theorem: [Aronszajn, 1950]

For a positive definite kernel k on $\mathcal{M}$, there is a Hilbert space $\mathcal{H}_k$ (reproducing kernel Hilbert space, (RKHS)) that consists of functions on $\mathcal{M}$, satisfying:

- $k(\mathbf{h}, \cdot) \in \mathcal{H}_k, \forall \mathbf{h} \in \mathcal{M}$;
- $\text{span}\{k(\mathbf{h}, \cdot); \mathbf{h} \in \mathcal{M}\}$ is dense in $\mathcal{H}_k$;
- **The reproducing property:** $\langle f(\cdot), k(\mathbf{h}, \cdot) \rangle = f(\mathbf{h}), \forall f \in \mathcal{H}_k, \mathbf{h} \in \mathcal{M}$.





Optimization and Manifold Regularization

Optimization in a reproducing kernel Hilbert space (RKHS) [Belkin et al., 2006]:

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}_k} \frac{1}{n_L} \sum_{i=1}^{n_L} (\bar{p}_i - f(\mathbf{h}_i))^2 + \gamma_k \|f\|_{\mathcal{H}_k}^2 + \gamma_M \|f\|_{\mathcal{M}}^2$$

Optimization and Manifold Regularization

Optimization in a reproducing kernel Hilbert space (RKHS) [Belkin et al., 2006]:

$$f^* = \underset{f \in \mathcal{H}_k}{\operatorname{argmin}} \frac{1}{n_L} \sum_{i=1}^{n_L} (\bar{p}_i - f(\mathbf{h}_i))^2 + \gamma_k \|f\|^2_{\mathcal{H}_k} + \gamma_M \|f\|^2_{\mathcal{M}}$$

Cost function

$\frac{1}{n_L} \sum_{i=1}^{n_L} (\bar{p}_i - f(\mathbf{h}_i))^2$

correspondence between function values and labels

Optimization and Manifold Regularization

Optimization in a reproducing kernel Hilbert space (RKHS) [Belkin et al., 2006]:

$$f^* = \underset{f \in \mathcal{H}_k}{\operatorname{argmin}} \frac{1}{n_L} \sum_{i=1}^{n_L} (\bar{p}_i - f(\mathbf{h}_i))^2 + \gamma_k \|f\|_{\mathcal{H}_k}^2 + \gamma_M \|f\|_{\mathcal{M}}^2$$

Cost function

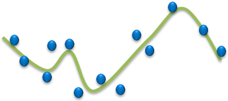
$\frac{1}{n_L} \sum_{i=1}^{n_L} (\bar{p}_i - f(\mathbf{h}_i))^2$

Tikhonov Regularization

$\|f\|_{\mathcal{H}_k}^2$

correspondence between function values and labels

smoothness condition in the RKHS

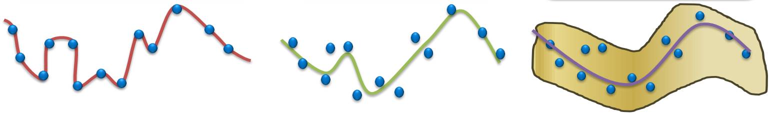


Optimization and Manifold Regularization

Optimization in a reproducing kernel Hilbert space (RKHS) [Belkin et al., 2006]:

$$f^* = \underset{f \in \mathcal{H}_k}{\operatorname{argmin}} \frac{1}{n_L} \sum_{i=1}^{n_L} (\bar{p}_i - f(\mathbf{h}_i))^2 + \gamma_k \|f\|^2_{\mathcal{H}_k} + \gamma_M \|f\|^2_{\mathcal{M}}$$

Cost function	Tikhonov Regularization	Manifold Regularization
$\frac{1}{n_L}\sum_{i=1}^{n_L}(\bar{p}_i - f(\mathbf{h}_i))^2$	$\|f\|^2_{\mathcal{H}_k}$	$\|f\|^2_{\mathcal{M}}$
correspondence between function values and labels	**smoothness condition in the RKHS**	**smoothness penalty with respect to the manifold**



Manifold Regularization

Smoothness on the manifold: A reminder

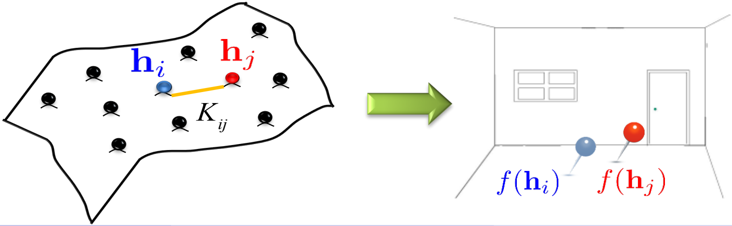
- The graph Laplacian:

$$\mathbf{L} = \mathbf{D} - \mathbf{K}$$

- Define the manifold regularization by:

$$\|f\|^2_{\mathcal{M}} = \mathbf{f}_D^T \mathbf{L} \mathbf{f}_D = \frac{1}{2} \sum_{i,j=1}^{n_D} K_{ij} \left(f(\mathbf{h}_i) - f(\mathbf{h}_j) \right)^2$$

$\mathbf{f}_D^T = [f_1, f_2, \ldots, f_{n_D}]$ comprising labelled and unlabelled training data



Optimization and Manifold Regularization

The optimization problem can be recast as:

$$f^* = \underset{f \in \mathcal{H}_k}{\operatorname{argmin}} \frac{1}{n_L} \sum_{i=1}^{n_L} (\bar{p}_i - f(\mathbf{h}_i))^2 + \gamma_k \|f\|^2_{\mathcal{H}_k} + \gamma_M \mathbf{f}_D^T \mathbf{L} \mathbf{f}_D$$

Optimization and Manifold Regularization

The optimization problem can be recast as:

$$f^* = \underset{f \in \mathcal{H}_k}{\operatorname{argmin}} \frac{1}{n_L} \sum_{i=1}^{n_L} (\bar{p}_i - f(\mathbf{h}_i))^2 + \gamma_k \|f\|^2_{\mathcal{H}_k} + \gamma_M \mathbf{f}_D^T \mathbf{L} \mathbf{f}_D$$

The Representer theorem: [Schölkopf et al., 2001]

The minimizer over $\mathcal{H}_k$ of the regularized optimization is represented by:

$$f^*(\mathbf{h}) = \sum_{i=1}^{n_D} a_i k(\mathbf{h}_i, \mathbf{h})$$

where $k : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ is the reproducing kernel of $\mathcal{H}_k$
with $K_{ij} = k(\mathbf{h}_i, \mathbf{h}_j) = \exp\left\{-\frac{\|\mathbf{h}_i - \mathbf{h}_j\|^2}{\varepsilon}\right\}$

Optimization and Manifold Regularization

The optimization problem can be recast as:

$$f^* = \underset{f \in \mathcal{H}_k}{\operatorname{argmin}} \frac{1}{n_L} \sum_{i=1}^{n_L} (\bar{p}_i - f(\mathbf{h}_i))^2 + \gamma_k \|f\|^2_{\mathcal{H}_k} + \gamma_M \mathbf{f}_D^T \mathbf{L} \mathbf{f}_D$$

The Representer theorem:

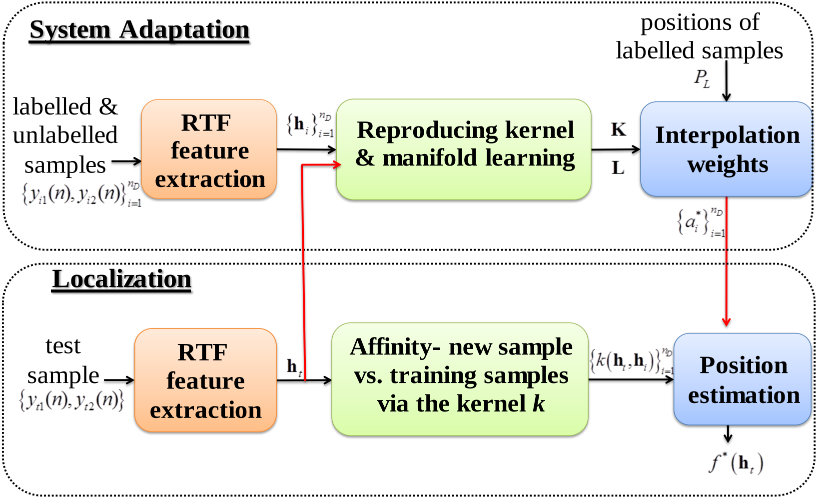
The minimizer over $\mathcal{H}_k$ of the regularized optimization is represented by:

$$f^*(\mathbf{h}) = \sum_{i=1}^{n_D} a_i k(\mathbf{h}_i, \mathbf{h}) \quad \Rightarrow \quad \text{closed-form solution for } \mathbf{a}^*$$

where $k : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}$ is the reproducing kernel of $\mathcal{H}_k$

Manifold Regularization for Localization (MRL)

[Laufer-Goldshtein et al., 2017]



$K_{ij} = k(\mathbf{h}_i, \mathbf{h}_j)$

Simulation Results

Setup:

- **Source positions:** angles between $10^\circ \div 60^\circ$
- **Training:** 6 labelled, 400 unlabelled (SNR=10 dB)

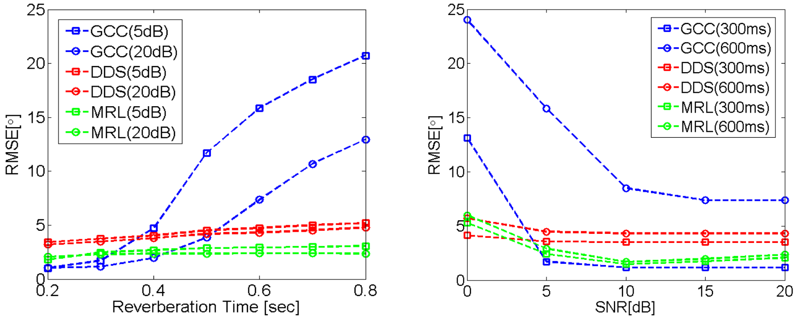


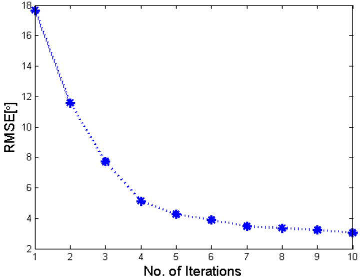
Figure: RMSEs of GCC, DDS and MRL as a function of reverberation time (left), SNR (right)

MRL achieves 2° accuracy in typical noisy and reverberant environments

Simulation Results - MRL

Iterative simulation:

- Source positions: angles between $0° \div 180°$
- Start with 19 labelled samples
- Each iteration add 80 unlabelled samples
- $T_{60} = 500$ ms and SNR=20 dB

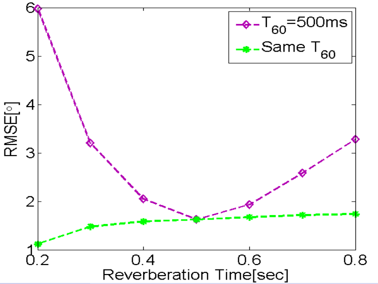


Sensitivity to reverberation level:

- Train with a fixed reverberation time of 500 ms.

→ for small mismatch - small increase in error level

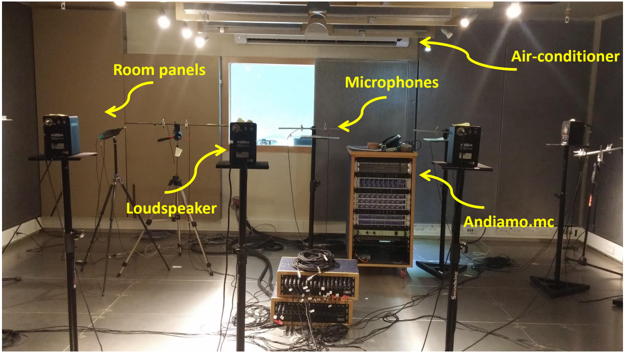
→ for large mismatch - large increase in error level



Recordings setup

Setup:

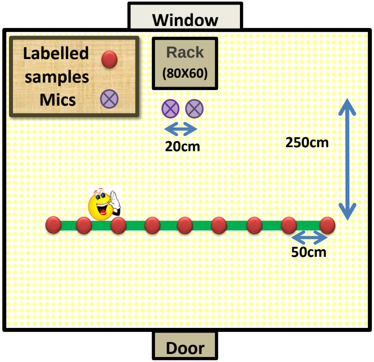
- Real recordings carried out at Bar-Ilan acoustic lab
- A $6 \times 6 \times 2.4$m room controllable reverberation time (set to 620ms)
- Region of interest: a 4m long line at 2.5m distance from the mics



Recordings setup

Setup:

- Real recordings carried out at Bar-Ilan acoustic lab
- A $6 \times 6 \times 2.4$m room controllable reverberation time (set to 620ms)
- Region of interest: a 4m long line at 2.5m distance from the mics



Experimental Results

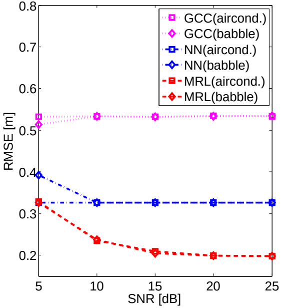
[Laufer-Goldshtein et al., 2016b]

Setup:

- Training: 5 labelled samples (1m resolution), 75 unlabelled samples
- Test: 30 random samples in the defined region
- Two noise types: air-conditioner noise and babble noise

Compare with:

- Nearest-neighbour (NN)
- Generalized cross-correlation (GCC) method [Knapp and Carter, 1976]



Experimental Results

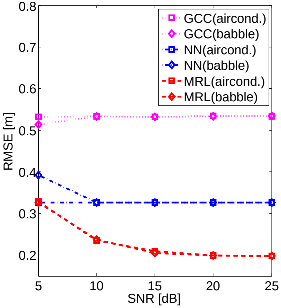
[Laufer-Goldshtein et al., 2016b]

Setup:

- Training: 5 labelled samples (1m resolution), 75 unlabelled samples
- Test: 30 random samples in the defined region
- Two noise types: air-conditioner noise and babble noise

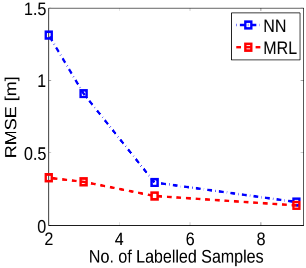
Compare with:

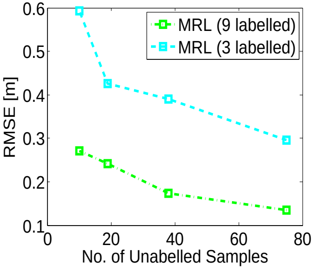
- Nearest-neighbour (NN)
- Generalized cross-correlation (GCC) method [Knapp and Carter, 1976]



The MRL algorithm outperforms the two other methods

Effect of Labelled & Unlabelled Samples





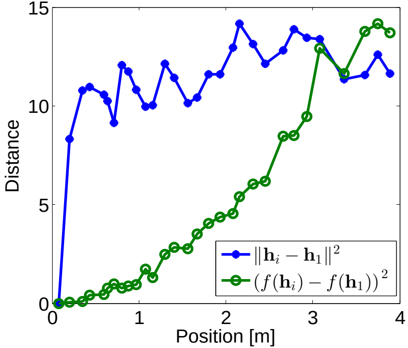
Effect of increasing the amount of labelled/unlabelled samples

→ As the size of the labelled set is reduced - performance gap increases

→ Locate the source even with few labelled samples, using unlabelled information

Why does Nearest-Neighbour Fail?

Compare distances before and after mapping

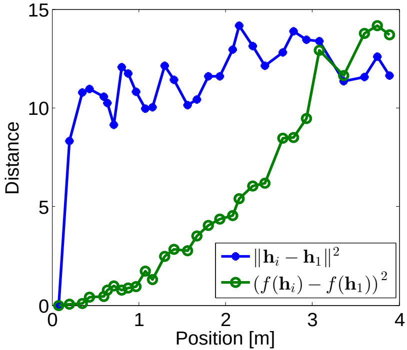


Monotony/Order

- Before mapping - monotonic/ordered only in a limited region
- After mapping - monotonic/ordered for almost the entire range

Why does Nearest-Neighbour Fail?

Compare distances before and after mapping



Monotony/Order

- Before mapping - monotonic/ordered only in a limited region
- After mapping - monotonic/ordered for almost the entire range

We conclude:

$\rightarrow$ RTFs lie on a nonlinear manifold - linear only for small patches

$\rightarrow$ NN ignores the manifold, MRL exploits the manifold structure

Localization on Manifolds

Two Data-Driven Localization Algorithms

Diffusion Distance Search (DDS)

- Diffusion Embedding
- Neighbors Search in the Embedded Space

1. Two stage
2. Better than GCC

Diffusion Maps

Manifold Regularization for Localization (MRL)

- Manifold Regularization
- Regularized Optimization

Optimization in RKHS

1. One stage
2. Best performance

Outline

Manifold-Based Bayesian Inference [Laufer-Goldshtein et al., 2016a]

Estimate the function f which transforms an RTF to position using a Bayesian approach with a data-driven geometric model

Update Belief

$$p(f|P_L, H_L) \propto p(P_L|f, H_L) \cdot p(f|H_L)$$

Posterior | **Likelihood** | **Prior**

Likelihood: Correspondence between the function values and the labels

Prior: A priori belief on the behavior of f - control smoothness

Manifold-Based Bayesian Inference [Laufer-Goldshtein et al., 2016a]

Estimate the function f which transforms an RTF to position using a Bayesian approach with a data-driven geometric model

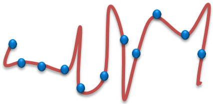
Update Belief

$$p(f|P_L, H_L) \propto p(P_L|f, H_L) \cdot p(f|H_L)$$

Posterior — **Likelihood** — **Prior**

Likelihood: Correspondence between the function values and the labels

Prior: A priori belief on the behavior of f - control smoothness



Manifold-Based Bayesian Inference [Laufer-Goldshtein et al., 2016a]

Estimate the function f which transforms an RTF to position using a Bayesian approach with a data-driven geometric model

Update Belief

$$p(f|P_L, H_L) \propto p(P_L|f, H_L) \cdot p(f|H_L)$$

Posterior **Likelihood** **Prior**

Correspondence between the function values and the labels

A priori belief on the behavior of f - control smoothness

Manifold-Based Bayesian Inference [Laufer-Goldshtein et al., 2016a]

Estimate the function f which transforms an RTF to position using a Bayesian approach with a data-driven geometric model

Update Belief

$$p(f|P_L, H_L) \propto p(P_L|f, H_L) \cdot p(f|H_L)$$

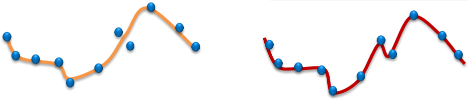
Posterior

Likelihood

Prior

Correspondence between the function values and the labels

A priori belief on the behavior of f - control smoothness



Manifold-Based Bayesian Inference [Laufer-Goldshtein et al., 2016a]

Estimate the function f which transforms an RTF to position using a Bayesian approach with a data-driven geometric model

Update Belief

$$\underbrace{p(f|P_L, H_L, H_U)}_{\text{Posterior}} \propto \underbrace{p(P_L|f, H_L)}_{\text{Likelihood}} \cdot \underbrace{p(f|H_L, H_U)}_{\text{Manifold-Based Prior}}$$

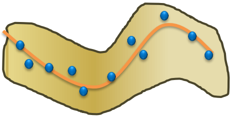
Posterior

Likelihood

Correspondence between the function values and the labels

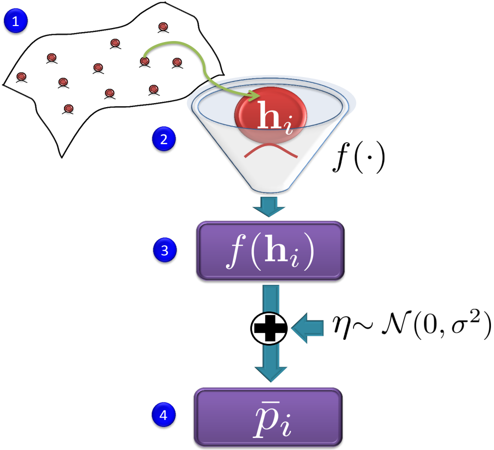
Manifold-Based Prior

a priori belief on the properties of f - smoothness with respect to the manifold



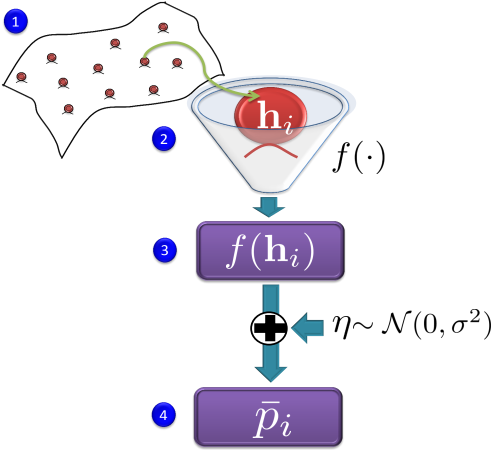
The Likelihood Function

1. An RTF is sampled from the manifold $\mathcal{M}$
2. The function f follows a stochastic process
3. The function receives an RTF sample and returns the position
4. Measure a noisy position due to imperfect calibration



The Likelihood Function

1. An RTF is sampled from the manifold $\mathcal{M}$
2. The function f follows a stochastic process
3. The function receives an RTF sample and returns the position
4. Measure a noisy position due to imperfect calibration



→ Likelihood function: $p(P_L | f, H_L) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n_L}(\bar{p}_i - f(\mathbf{h}_i))^2\right\}$

Standard Prior Probability

Standard Gaussian process [Rasmussen and Williams, 2006]:

- The function f follows a Gaussian process:

$$f(\mathbf{h}) \sim \mathcal{GP}\left(\nu(\mathbf{h}), k(\mathbf{h}, \mathbf{h}_i)\right)$$

- ν is the mean function (choose $\nu \equiv 0$)
- k is the covariance function.
- The r.v. $\mathbf{f}_H = [f(\mathbf{h}_1), \ldots, f(\mathbf{h}_n)]$ has a joint Gaussian distribution:

$$\mathbf{f}_H \sim \mathcal{N}(\mathbf{0}_n, \mathbf{\Sigma}_{HH})$$

where $\mathbf{\Sigma}_{HH}$ is the covariance matrix with elements $k(\mathbf{h}_i, \mathbf{h}_j)$
- Common choice: a Gaussian kernel $k(\mathbf{h}_i, \mathbf{h}_j) = \exp\{-\|\mathbf{h}_i - \mathbf{h}_j\|^2/\varepsilon_k\}$

Standard Prior Probability

Standard Gaussian process [Rasmussen and Williams, 2006]:

- The function f follows a Gaussian process:

$$f(\mathbf{h}) \sim \mathcal{GP}\left(\nu(\mathbf{h}), k(\mathbf{h}, \mathbf{h}_i)\right)$$

- ν is the mean function (choose $\nu \equiv 0$)
- k is the covariance function.
- The r.v. $\mathbf{f}_H = [f(\mathbf{h}_1), \ldots, f(\mathbf{h}_n)]$ has a joint Gaussian distribution:

$$\mathbf{f}_H \sim \mathcal{N}(\mathbf{0}_n, \mathbf{\Sigma}_{HH})$$

where $\mathbf{\Sigma}_{HH}$ is the covariance matrix with elements $k(\mathbf{h}_i, \mathbf{h}_j)$

- Common choice: a Gaussian kernel $k(\mathbf{h}_i, \mathbf{h}_j) = \exp\{-\|\mathbf{h}_i - \mathbf{h}_j\|^2/\varepsilon_k\}$

✗ The correlation for intermediate distances may be incorrectly assessed

✗ Does not exploit the available set of unlabelled data H_U

Manifold-Based Prior Probability [Sindhwani et al., 2007]

Discretization of the manifold

- The manifold is empirically represented by a graph G, with weights:

$$W_{ij} = \begin{cases} \exp\left\{-\frac{\|\mathbf{h}_i - \mathbf{h}_j\|^2}{\varepsilon_w}\right\} & \text{if } \mathbf{h}_j \in \mathcal{N}_i \text{ or } \mathbf{h}_i \in \mathcal{N}_j \\ 0 & \text{otherwise} \end{cases}$$

- The graph Laplacian of G: $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $D_{ii} = \sum_{j=1}^{n} W_{ij}$.

Manifold-Based Prior Probability [Sindhwani et al., 2007]

Discretization of the manifold

- The manifold is empirically represented by a graph G, with weights:

$$W_{ij} = \begin{cases} \exp\left\{-\frac{\|\mathbf{h}_i-\mathbf{h}_j\|^2}{\varepsilon_w}\right\} & \text{if } \mathbf{h}_j \in \mathcal{N}_i \text{ or } \mathbf{h}_i \in \mathcal{N}_j \\ 0 & \text{otherwise} \end{cases}$$

- The graph Laplacian of G: $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $D_{ii} = \sum_{j=1}^{n} W_{ij}$.

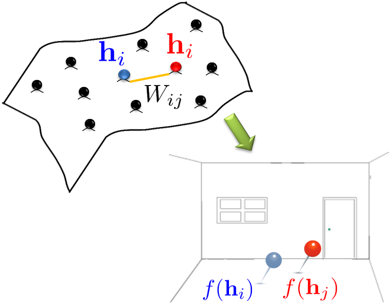
Statistical formulation

- Geometry variables $\mathcal{G}$ – represent the manifold structure
- The likelihood of $\mathcal{G}$:

$$P(\mathcal{G}|\mathbf{f}_D) \propto \exp\left\{-\frac{\gamma_M}{2}\left(\mathbf{f}_D^T\mathbf{L}\mathbf{f}_D\right)\right\}$$

- We showed (based on all n_D training samples):

$$\mathbf{f}_D^T\mathbf{L}\mathbf{f}_D = \frac{1}{2}\sum_{i,j=1}^{n_D} W_{ij}\left(f(\mathbf{h}_i) - f(\mathbf{h}_j)\right)^2$$



Manifold-Based Prior Probability [Sindhwani et al., 2007]

Manifold-Based GP Prior — **Likelihood of Geometry Variables** — **Standard GP Prior**

$$p(\mathbf{f}_H|\mathcal{G}) \propto p(\mathcal{G}|\mathbf{f}_H) \cdot p(\mathbf{f}_H)$$

The covariance is formed by a manifold-based kernel $\tilde{k}$

correspondence between the function values and the manifold structure

The covariance is formed by a standard kernel k

Assume: $p(\mathcal{G}|\mathbf{f}_H) \propto p(\mathcal{G}|\mathbf{f}_D)$

$$p(\mathbf{f}_H|\mathcal{G}) = \mathcal{N}(\mathbf{0}_m, \tilde{\mathbf{\Sigma}}_{HH})$$

$$p(\mathcal{G}|\mathbf{f}_D) \propto \exp\left\{-\frac{\gamma_M}{2}\left(\mathbf{f}_D^T \mathbf{L} \mathbf{f}_D\right)\right\}$$

$$p(\mathbf{f}_H) = \mathcal{N}(\mathbf{0}_m, \mathbf{\Sigma}_{HH})$$

$$\tilde{\Sigma}_{\mathrm{HH}} \Leftrightarrow \tilde{k}(\mathbf{h}_i, \mathbf{h}_j)$$

Manifold-Based Bayesian Inference [Laufer-Goldshtein et al., 2016a]

$$f^* = \operatorname{argmin}_{f \in \mathcal{H}_k} \frac{1}{n_L} \sum_{i=1}^{n_L} (\bar{p}_i - f(\mathbf{h}_i))^2 + \gamma_k \|f\|^2_{\mathcal{H}_k} + \gamma_M \mathbf{f}_D^T \mathbf{M} \mathbf{f}_D$$

Search in RKHS defined by the kernel k

Cost Function

$\mathcal{H}_k$ **norm**

Manifold Regularization

Manifold-Based Bayesian Inference [Laufer-Goldshtein et al., 2016a]

$$f^* = \operatorname{argmin}_{f \in \mathcal{H}_k} \underbrace{\frac{1}{n_L} \sum_{i=1}^{n_L} (\bar{p}_i - f(\mathbf{h}_i))^2}_{\text{Cost Function}} + \underbrace{\gamma_k \|f\|^2_{\mathcal{H}_k}}_{\mathcal{H}_k \text{ norm}} + \underbrace{\gamma_M \mathbf{f}_D^T \mathbf{M} \mathbf{f}_D}_{\text{Manifold Regularization}}$$

Search in RKHS defined by the kernel k

Cost Function

$\mathcal{H}_k$ norm

Manifold Regularization

$$f^* = \operatorname{argmin}_{f \in \mathcal{H}_{\tilde{k}}} \underbrace{\frac{1}{n_L} \sum_{i=1}^{n_L} (\bar{p}_i - f(\mathbf{h}_i))^2}_{\text{Cost Function}} + \underbrace{\gamma_k \|f\|^2_{\mathcal{H}_{\tilde{k}}}}_{\mathcal{H}_{\tilde{k}} \text{ norm}}$$

Search in RKHS defined by the kernel $\tilde{k}$

Cost Function

$\mathcal{H}_{\tilde{k}}$ norm

Manifold-Based Bayesian Inference [Laufer-Goldshtein et al., 2016a]

$$f^* = \text{argmin}_{f\in\mathcal{H}_k} \frac{1}{n_L}\sum_{i=1}^{n_L}(\bar{p}_i - f(\mathbf{h}_i))^2 + \gamma_k\|f\|^2_{\mathcal{H}_k} + \gamma_M \mathbf{f}_D^T\mathbf{M}\mathbf{f}_D$$

- Search in RKHS defined by the kernel k
- Cost Function
- $\mathcal{H}_k$ norm
- Manifold Regularization

$$f^* = \text{argmin}_{f\in\mathcal{H}_{\tilde{k}}} \frac{1}{n_L}\sum_{i=1}^{n_L}(\bar{p}_i - f(\mathbf{h}_i))^2 + \gamma_k\|f\|^2_{\mathcal{H}_{\tilde{k}}}$$

- Search in RKHS defined by the kernel $\tilde{k}$
- Cost Function
- $\mathcal{H}_{\tilde{k}}$ norm

$$p(f|P_L, H_L, H_U) \propto p(P_L|f, H_L) \cdot p(f|H_L, H_U)$$

- Posterior
- Likelihood Function
- Manifold-Based Prior

f is a Gaussian Process with Covariance $\tilde{k}$

Bayesian Localization

Joint probability:

- Goal: estimate the function value at some test sample $\mathbf{h}_t \in \mathcal{M}$
- The training positions $\bar{\mathbf{p}}_L = \mathrm{vec}\{P_L\}$ and $f(\mathbf{h}_t)$ are jointly Gaussian:

$$\begin{bmatrix} \bar{\mathbf{p}}_L \\ f(\mathbf{h}_t) \end{bmatrix} \Big| H_L, H_U \sim \mathcal{N}\left(\mathbf{0}_{n_L+1}, \begin{bmatrix} \tilde{\mathbf{\Sigma}}_{LL} + \sigma^2 \mathbf{I}_{n_L} & \tilde{\mathbf{\Sigma}}_{Lt} \\ \tilde{\mathbf{\Sigma}}_{Lt}^T & \tilde{\mathbf{\Sigma}}_{tt} \end{bmatrix}\right)$$

- The elements of $\tilde{\mathbf{\Sigma}}_{LL}$, $\tilde{\mathbf{\Sigma}}_{Lt}$ and $\tilde{\mathbf{\Sigma}}_{tt}$ are calculated by the manifold-regularized kernel

$$\mathrm{cov}(f(\mathbf{h}_r), f(\mathbf{h}_l)) \equiv \tilde{k}(\mathbf{h}_r, \mathbf{h}_l)$$

- Note that the unlabelled points are implicitly considered in the covariance terms

Bayesian Localization (cont.)

MAP/MMSE estimator:

- The posterior

$$p(f(\mathbf{h}_t)|P_L, H_L, H_U) \sim \mathcal{N}(\hat{f}(\mathbf{h}_t), \mathrm{var}(\hat{f}(\mathbf{h}_t)))$$

is a multivariate Gaussian, where:

  - The MAP/MMSE estimator of $f(\mathbf{h}_t)$ is given by:

$$\hat{f}(\mathbf{h}_t) = \tilde{\mathbf{\Sigma}}_{Lt}^T \left(\tilde{\mathbf{\Sigma}}_{LL} + \sigma^2 \mathbf{I}_{n_L}\right)^{-1} \bar{\mathbf{p}}_L$$

  - The estimation confidence:

$$\mathrm{var}(\hat{f}(\mathbf{h}_t)) = \tilde{\Sigma}_{tt} - \tilde{\mathbf{\Sigma}}_{Lt}^T \left(\tilde{\mathbf{\Sigma}}_{LL} + \sigma^2 \mathbf{I}_{n_L}\right)^{-1} \tilde{\mathbf{\Sigma}}_{Lt}$$

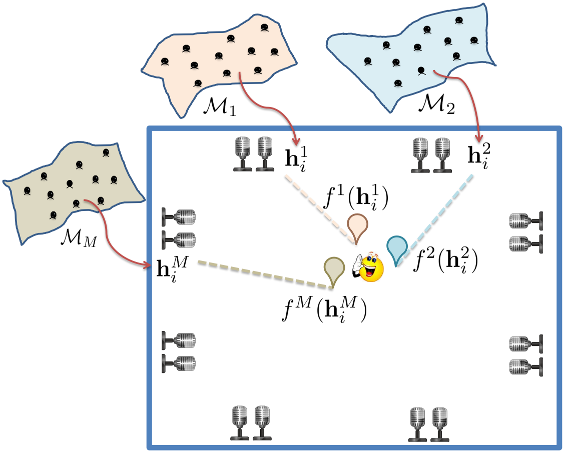
Learning the Hyperparameters: [Laufer-Goldshtein et al., 2017]

- The hyperparameters:
  - Kernel scales ϵ
  - Weights γ (Gaussian process variance)

  can be inferred from the data by optimizing the likelihood function of the labelled samples

Outline

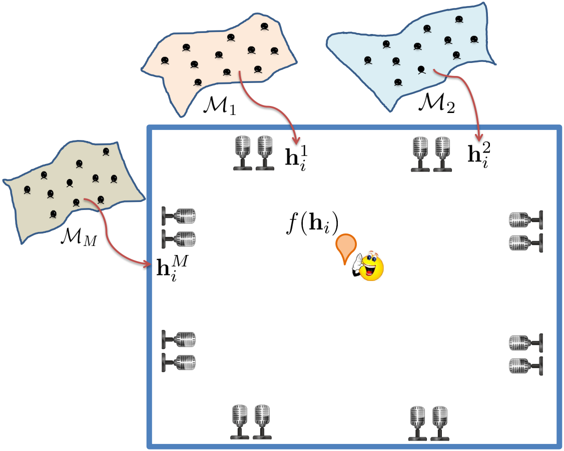
Source Localization with Ad Hoc Array [Laufer-Goldshtein et al., 2017]



Each node

- Represents a different view point on the same acoustic event
- Induces relations between RTFs according to the associated manifold

Source Localization with Ad Hoc Array [Laufer-Goldshtein et al., 2017]

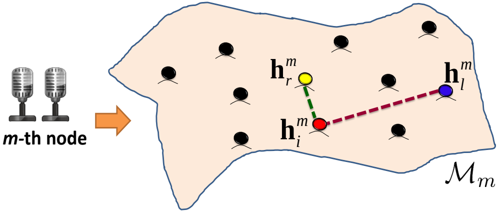


$$\mathbf{h}_i = [(\mathbf{h}_i^1)^T, (\mathbf{h}_i^2)^T, \dots, (\mathbf{h}_i^M)^T]^T \in \cup_{m=1}^M \mathcal{M}_m$$

How to fuse the different views in a unified mapping $f : \cup_{m=1}^M \mathcal{M}_m \mapsto \mathbb{R}$?

Inta-Manifold Relations

The mapping follows a Gaussian process $f^m(\mathbf{h}^m) \sim \mathcal{GP}(0, \tilde{k}_m(\mathbf{h}^m, \mathbf{h}_i^m))$



Covariance function

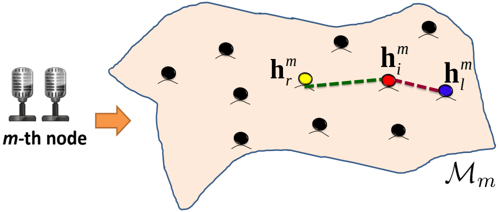
Defined by a new manifold-based covariance function:

$$\mathrm{cov}\left(f^m(\mathbf{h}_r^m), f^m(\mathbf{h}_l^m)\right) \equiv \tilde{k}_m(\mathbf{h}_r^m, \mathbf{h}_l^m) = \sum_{i=1}^{n_D} k_m(\mathbf{h}_r^m, \mathbf{h}_i^m) k_m(\mathbf{h}_l^m, \mathbf{h}_i^m)$$

$$= 2k_m(\mathbf{h}_r^m, \mathbf{h}_l^m) + \sum_{\substack{i=1 \\ i \neq l,r}}^{n_D} k_m(\mathbf{h}_r^m, \mathbf{h}_i^m) k_m(\mathbf{h}_l^m, \mathbf{h}_i^m)$$

Inta-Manifold Relations

The mapping follows a Gaussian process $f^m(\mathbf{h}^m) \sim \mathcal{GP}(0, \tilde{k}_m(\mathbf{h}^m, \mathbf{h}_i^m))$



Covariance function

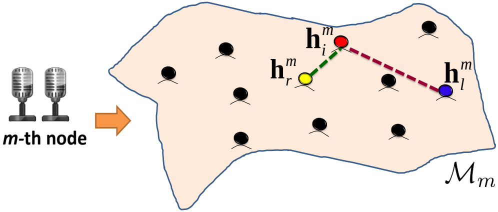
Defined by a new manifold-based covariance function:

$$\mathrm{cov}\left(f^m(\mathbf{h}_r^m), f^m(\mathbf{h}_l^m)\right) \equiv \tilde{k}_m(\mathbf{h}_r^m, \mathbf{h}_l^m) = \sum_{i=1}^{n_D} k_m(\mathbf{h}_r^m, \mathbf{h}_i^m) k_m(\mathbf{h}_l^m, \mathbf{h}_i^m)$$

$$= 2k_m(\mathbf{h}_r^m, \mathbf{h}_l^m) + \sum_{\substack{i=1 \\ i \neq l,r}}^{n_D} k_m(\mathbf{h}_r^m, \mathbf{h}_i^m) k_m(\mathbf{h}_l^m, \mathbf{h}_i^m)$$

Inta-Manifold Relations

The mapping follows a Gaussian process $f^m(\mathbf{h}^m) \sim \mathcal{GP}(0, \tilde{k}_m(\mathbf{h}^m, \mathbf{h}_i^m))$



Covariance function

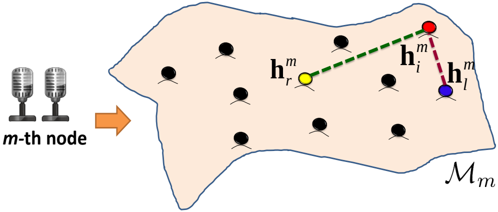
Defined by a new manifold-based covariance function:

$$\mathrm{cov}\left(f^m(\mathbf{h}_r^m), f^m(\mathbf{h}_l^m)\right) \equiv \tilde{k}_m(\mathbf{h}_r^m, \mathbf{h}_l^m) = \sum_{i=1}^{n_D} k_m(\mathbf{h}_r^m, \mathbf{h}_i^m) k_m(\mathbf{h}_l^m, \mathbf{h}_i^m)$$

$$= 2k_m(\mathbf{h}_r^m, \mathbf{h}_l^m) + \sum_{\substack{i=1 \\ i \neq l,r}}^{n_D} k_m(\mathbf{h}_r^m, \mathbf{h}_i^m) k_m(\mathbf{h}_l^m, \mathbf{h}_i^m)$$

Inta-Manifold Relations

The mapping follows a Gaussian process $f^m(\mathbf{h}^m) \sim \mathcal{GP}(0, \tilde{k}_m(\mathbf{h}^m, \mathbf{h}_i^m))$



Covariance function

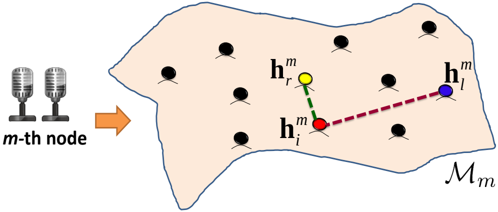
Defined by a new manifold-based covariance function:

$$\mathrm{cov}\left(f^m(\mathbf{h}_r^m), f^m(\mathbf{h}_l^m)\right) \equiv \tilde{k}_m(\mathbf{h}_r^m, \mathbf{h}_l^m) = \sum_{i=1}^{n_D} k_m(\mathbf{h}_r^m, \mathbf{h}_i^m) k_m(\mathbf{h}_l^m, \mathbf{h}_i^m)$$

$$= 2k_m(\mathbf{h}_r^m, \mathbf{h}_l^m) + \sum_{\substack{i=1 \\ i \neq l,r}}^{n_D} k_m(\mathbf{h}_r^m, \mathbf{h}_i^m) k_m(\mathbf{h}_l^m, \mathbf{h}_i^m)$$

Inta-Manifold Relations

The mapping follows a Gaussian process $f^m(\mathbf{h}^m) \sim \mathcal{GP}(0, \tilde{k}_m(\mathbf{h}^m, \mathbf{h}_i^m))$



Covariance function

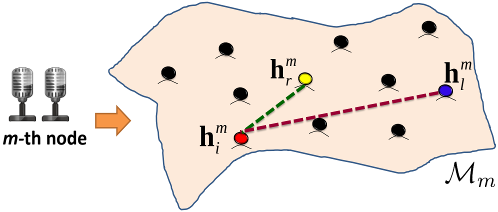
Defined by a new manifold-based covariance function:

$$\mathrm{cov}\left(f^m(\mathbf{h}_r^m), f^m(\mathbf{h}_l^m)\right) \equiv \tilde{k}_m(\mathbf{h}_r^m, \mathbf{h}_l^m) = \sum_{i=1}^{n_D} k_m(\mathbf{h}_r^m, \mathbf{h}_i^m) k_m(\mathbf{h}_l^m, \mathbf{h}_i^m)$$

$$= 2k_m(\mathbf{h}_r^m, \mathbf{h}_l^m) + \sum_{\substack{i=1 \\ i \neq l,r}}^{n_D} k_m(\mathbf{h}_r^m, \mathbf{h}_i^m) k_m(\mathbf{h}_l^m, \mathbf{h}_i^m)$$

Inta-Manifold Relations

The mapping follows a Gaussian process $f^m(\mathbf{h}^m) \sim \mathcal{GP}(0, \tilde{k}_m(\mathbf{h}^m, \mathbf{h}_i^m))$



Covariance function

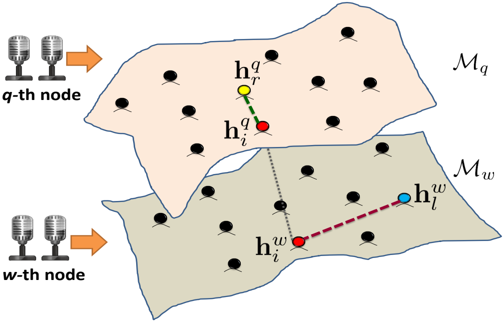
Defined by a new manifold-based covariance function:

$$\operatorname{cov}(f^m(\mathbf{h}_r^m), f^m(\mathbf{h}_l^m)) \equiv \tilde{k}_m(\mathbf{h}_r^m, \mathbf{h}_l^m) = \sum_{i=1}^{n_D} k_m(\mathbf{h}_r^m, \mathbf{h}_i^m) k_m(\mathbf{h}_l^m, \mathbf{h}_i^m)$$

$$= 2k_m(\mathbf{h}_r^m, \mathbf{h}_l^m) + \sum_{\substack{i=1 \\ i \neq l,r}}^{n_D} k_m(\mathbf{h}_r^m, \mathbf{h}_i^m) k_m(\mathbf{h}_l^m, \mathbf{h}_i^m)$$

Inter-Manifold Relations

How to measure relations between RTFs from different nodes?



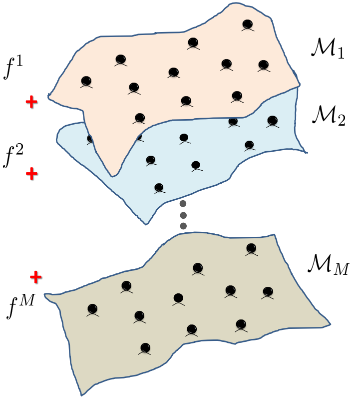
Multi-node covariance

The covariance between $f^q(\mathbf{h}_r^q)$ and $f^w(\mathbf{h}_r^w)$:

$$\mathrm{cov}\left(f^q(\mathbf{h}_r^q), f^w(\mathbf{h}_r^w)\right) = \sum_{i=1}^{n_D} k_q(\mathbf{h}_r^q, \mathbf{h}_i^q) k_w(\mathbf{h}_l^w, \mathbf{h}_i^w)$$

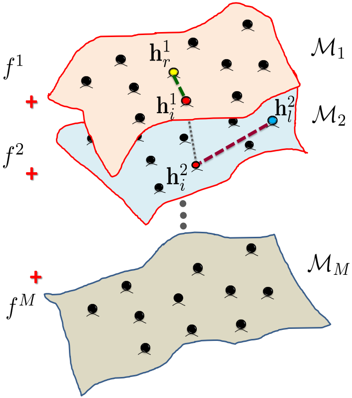
Multiple Manifold Gaussian Process (MMGP)

Define the average process $f = \frac{1}{M}(f^1 + f^2 + \ldots + f^M) \sim \mathcal{GP}(0, \tilde{k})$



Multiple Manifold Gaussian Process (MMGP)

Define the average process $f = \frac{1}{M}(f^1 + f^2 + \ldots + f^M) \sim \mathcal{GP}(0, \tilde{k})$

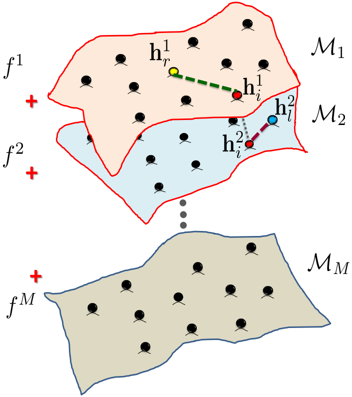


The covariance between $f(\mathbf{h}_r)$ and $f(\mathbf{h}_l)$

$$\mathrm{cov}(f(\mathbf{h}_r), f(\mathbf{h}_l)) \equiv \tilde{k}(\mathbf{h}_r, \mathbf{h}_l) = \frac{1}{M^2} \sum_{q,w=1}^{M} \sum_{i=1}^{n_D} k_q(\mathbf{h}_r^q, \mathbf{h}_i^q) k_w(\mathbf{h}_l^w, \mathbf{h}_i^w)$$

Multiple Manifold Gaussian Process (MMGP)

Define the average process $f = \frac{1}{M}(f^1 + f^2 + \ldots + f^M) \sim \mathcal{GP}(0, \tilde{k})$

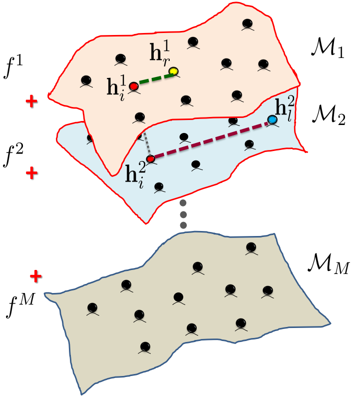


The covariance between $f(\mathbf{h}_r)$ and $f(\mathbf{h}_l)$

$$\mathrm{cov}(f(\mathbf{h}_r), f(\mathbf{h}_l)) \equiv \tilde{k}(\mathbf{h}_r, \mathbf{h}_l) = \frac{1}{M^2} \sum_{q,w=1}^{M} \sum_{i=1}^{n_D} k_q(\mathbf{h}_r^q, \mathbf{h}_i^q) k_w(\mathbf{h}_l^w, \mathbf{h}_i^w)$$

Multiple Manifold Gaussian Process (MMGP)

Define the average process $f = \frac{1}{M}(f^1 + f^2 + \ldots + f^M) \sim \mathcal{GP}(0, \tilde{k})$

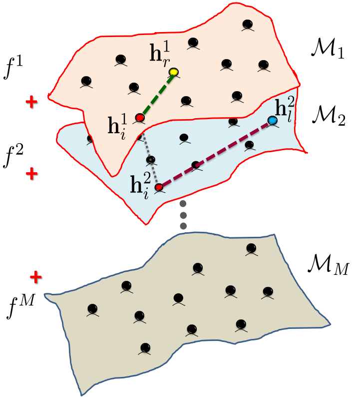


The covariance between $f(\mathbf{h}_r)$ and $f(\mathbf{h}_l)$

$$\mathrm{cov}(f(\mathbf{h}_r), f(\mathbf{h}_l)) \equiv \tilde{k}(\mathbf{h}_r, \mathbf{h}_l) = \frac{1}{M^2} \sum_{q,w=1}^{M} \sum_{i=1}^{n_D} k_q(\mathbf{h}_r^q, \mathbf{h}_i^q) k_w(\mathbf{h}_l^w, \mathbf{h}_i^w)$$

Multiple Manifold Gaussian Process (MMGP)

Define the average process $f = \frac{1}{M}(f^1 + f^2 + \ldots + f^M) \sim \mathcal{GP}(0, \tilde{k})$

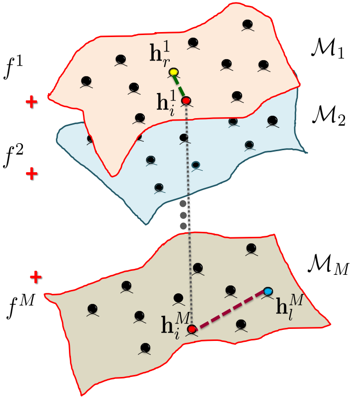


The covariance between $f(\mathbf{h}_r)$ and $f(\mathbf{h}_l)$

$$\mathrm{cov}(f(\mathbf{h}_r), f(\mathbf{h}_l)) \equiv \tilde{k}(\mathbf{h}_r, \mathbf{h}_l) = \frac{1}{M^2} \sum_{q,w=1}^{M} \sum_{i=1}^{n_D} k_q(\mathbf{h}_r^q, \mathbf{h}_i^q) k_w(\mathbf{h}_l^w, \mathbf{h}_i^w)$$

Multiple Manifold Gaussian Process (MMGP)

Define the average process $f = \frac{1}{M}(f^1 + f^2 + \ldots + f^M) \sim \mathcal{GP}(0, \tilde{k})$

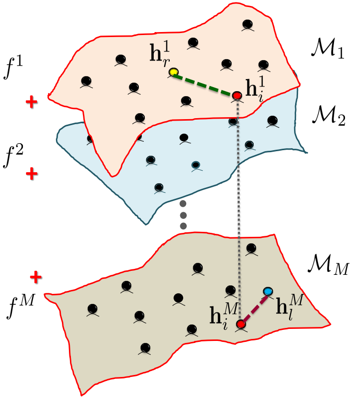


The covariance between $f(\mathbf{h}_r)$ and $f(\mathbf{h}_l)$

$$\mathrm{cov}(f(\mathbf{h}_r), f(\mathbf{h}_l)) \equiv \tilde{k}(\mathbf{h}_r, \mathbf{h}_l) = \frac{1}{M^2} \sum_{q,w=1}^{M} \sum_{i=1}^{n_D} k_q(\mathbf{h}_r^q, \mathbf{h}_i^q) k_w(\mathbf{h}_l^w, \mathbf{h}_i^w)$$

Multiple Manifold Gaussian Process (MMGP)

Define the average process $f = \frac{1}{M}(f^1 + f^2 + \ldots + f^M) \sim \mathcal{GP}(0, \tilde{k})$

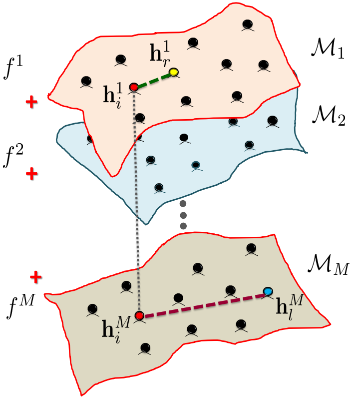


The covariance between $f(\mathbf{h}_r)$ and $f(\mathbf{h}_l)$

$$\mathrm{cov}(f(\mathbf{h}_r), f(\mathbf{h}_l)) \equiv \tilde{k}(\mathbf{h}_r, \mathbf{h}_l) = \frac{1}{M^2} \sum_{q,w=1}^{M} \sum_{i=1}^{n_D} k_q(\mathbf{h}_r^q, \mathbf{h}_i^q) k_w(\mathbf{h}_l^w, \mathbf{h}_i^w)$$

Multiple Manifold Gaussian Process (MMGP)

Define the average process $f = \frac{1}{M}(f^1 + f^2 + \ldots + f^M) \sim \mathcal{GP}(0, \tilde{k})$

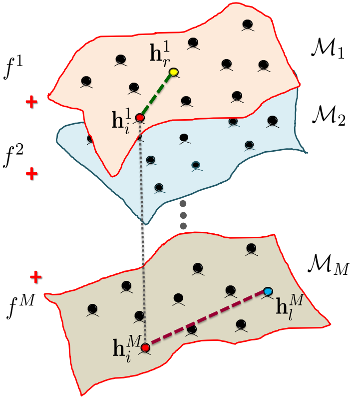


The covariance between $f(\mathbf{h}_r)$ and $f(\mathbf{h}_l)$

$$\mathrm{cov}(f(\mathbf{h}_r), f(\mathbf{h}_l)) \equiv \tilde{k}(\mathbf{h}_r, \mathbf{h}_l) = \frac{1}{M^2} \sum_{q,w=1}^{M} \sum_{i=1}^{n_D} k_q(\mathbf{h}_r^q, \mathbf{h}_i^q) k_w(\mathbf{h}_l^w, \mathbf{h}_i^w)$$

Multiple Manifold Gaussian Process (MMGP)

Define the average process $f = \frac{1}{M}(f^1 + f^2 + \ldots + f^M) \sim \mathcal{GP}(0, \tilde{k})$



The covariance between $f(\mathbf{h}_r)$ and $f(\mathbf{h}_l)$

$$\mathrm{cov}(f(\mathbf{h}_r), f(\mathbf{h}_l)) \equiv \tilde{k}(\mathbf{h}_r, \mathbf{h}_l) = \frac{1}{M^2} \sum_{q,w=1}^{M} \sum_{i=1}^{n_D} k_q(\mathbf{h}_r^q, \mathbf{h}_i^q) k_w(\mathbf{h}_l^w, \mathbf{h}_i^w)$$

Bayesian Multi-View Localization

Joint probability

- Goal: estimate the function value at some test sample $\mathbf{h}_t$
- The training positions $\bar{\mathbf{p}}_L = \mathrm{vec}\{P_L\}$ and $f(\mathbf{h}_t)$ are jointly Gaussian:

$$\begin{bmatrix} \bar{\mathbf{p}}_L \\ f(\mathbf{h}_t) \end{bmatrix} \Bigg| H_L, H_U \sim \mathcal{N}\left(\mathbf{0}_{n_L+1}, \begin{bmatrix} \tilde{\mathbf{\Sigma}}_{LL} + \sigma^2 \mathbf{I}_{n_L} & \tilde{\mathbf{\Sigma}}_{Lt} \\ \tilde{\mathbf{\Sigma}}_{Lt}^T & \tilde{\mathbf{\Sigma}}_{tt} \end{bmatrix}\right)$$

- The elements of $\tilde{\mathbf{\Sigma}}_{LL}$, $\tilde{\mathbf{\Sigma}}_{Lt}$ and $\tilde{\mathbf{\Sigma}}_{tt}$ are calculated by the multiple manifold kernel

$$\mathrm{cov}(f(\mathbf{h}_r), f(\mathbf{h}_l)) \equiv \tilde{k}(\mathbf{h}_r, \mathbf{h}_l)$$

- Note that the unlabelled points are implicitly considered in the covariance terms

Bayesian Multi-View Localization (cont.)

MAP/MMSE estimator:

- The posterior

$$p(f(\mathbf{h}_t)|P_L, H_L, H_U) \sim \mathcal{N}(\hat{f}(\mathbf{h}_t), \mathrm{var}(\hat{f}(\mathbf{h}_t)))$$

is a multivariate Gaussian, where:

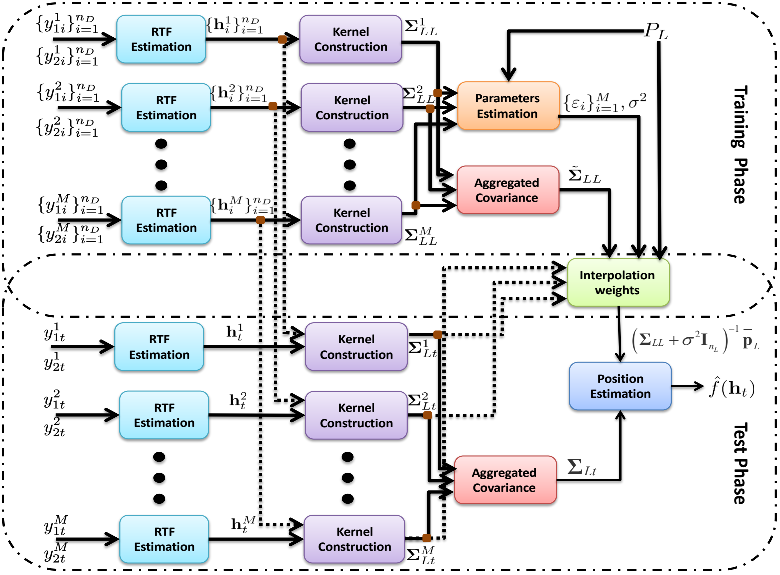
- The MAP/MMSE estimator of $f(\mathbf{h}_t)$ is given by:

$$\hat{f}(\mathbf{h}_t) = \tilde{\mathbf{\Sigma}}_{Lt}^T \left(\tilde{\mathbf{\Sigma}}_{LL} + \sigma^2 \mathbf{I}_{n_L}\right)^{-1} \bar{\mathbf{p}}_L$$

- The estimation confidence

$$\mathrm{var}(\hat{f}(\mathbf{h}_t)) = \tilde{\Sigma}_{tt} - \tilde{\mathbf{\Sigma}}_{Lt}^T \left(\tilde{\mathbf{\Sigma}}_{LL} + \sigma^2 \mathbf{I}_{n_L}\right)^{-1} \tilde{\mathbf{\Sigma}}_{Lt}$$

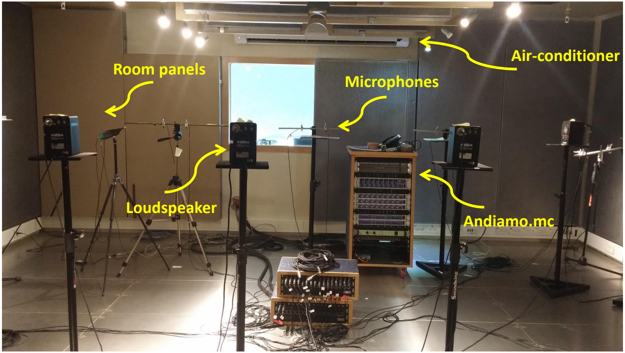
Multiple-Manifold Gaussian Process (MMGP)



Recordings Setup

Setup:

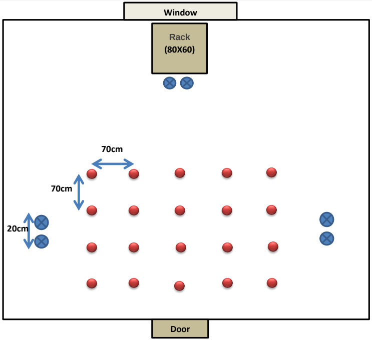
- Real recordings carried out at Bar-Ilan acoustic lab
- A $6 \times 6 \times 2.4$m room controllable reverberation time (set to 620ms)
- Region of interest: Source position is confined to a 2.8×2.1m area
- 3 microphone pairs with inter-distance of 0.2m



Recordings Setup

Setup:

- Real recordings carried out at Bar-Ilan acoustic lab
- A $6 \times 6 \times 2.4$m room controllable reverberation time (set to 620ms)
- Region of interest: Source position is confined to a 2.8×2.1m area
- 3 microphone pairs with inter-distance of 0.2m



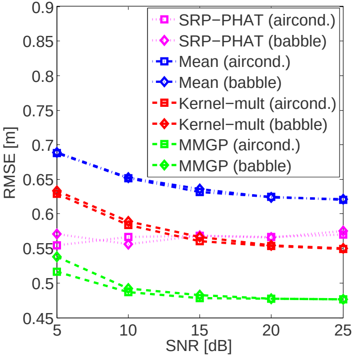
Experimental Results [Laufer-Goldshtein et al., 2017]

Setup:

- Training: 20 labelled samples (0.7m resolution), 50 unlabelled samples
- Test: 25 random samples in the defined region
- Two noise types: air-conditioner noise and babble noise

Compare with:

- Concatenated independent measurements (Kernel-mult)
- Average of single-node estimates (Mean)
- Beamformer scanning (SRP-PHAT [DiBiase et al., 2001])

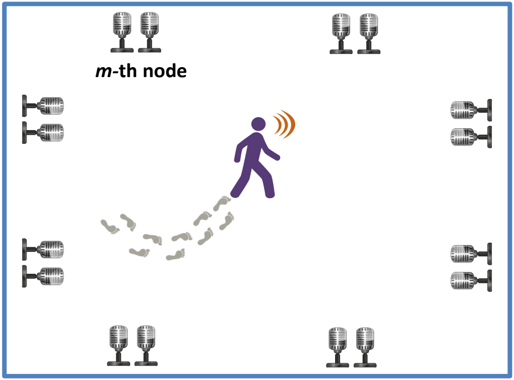


Outline

Speaker Tracking

Scenario:

- A source is moving in a reverberant enclosure
- Measured by an ad-hoc network with distributed microphones
- Microphones are arranged in M pairs - "nodes"



Bayesian Inference for Source Tracking

Standard state-space model

$$\mathbf{p}(t) = b(\mathbf{p}(t-1)) + \boldsymbol{\xi}(t)$$
$$\mathbf{q}(t) = c(\mathbf{p}(t)) + \boldsymbol{\zeta}(t)$$

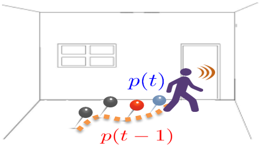
Bayesian Inference for Source Tracking

Standard state-space model

$$\mathbf{p}(t) = b(\mathbf{p}(t-1)) + \boldsymbol{\xi}(t)$$
$$\mathbf{q}(t) = c(\mathbf{p}(t)) + \zeta(t)$$

Propagation model

- Relates current and previous positions using random walk model or Langevin model
- Independent of measurements
- Noise statistic is unknown



Bayesian Inference for Source Tracking

Standard state-space model

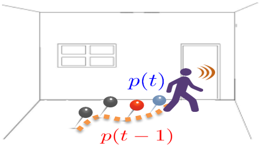
$$\mathbf{p}(t) = b(\mathbf{p}(t-1)) + \xi(t)$$
$$\mathbf{q}(t) = c(\mathbf{p}(t)) + \zeta(t)$$

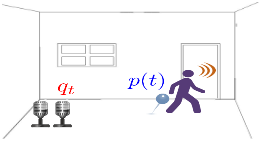
Propagation model

- Relates current and previous positions using random walk model or Langevin model
- Independent of measurements
- Noise statistic is unknown

Observation model

- Relates current position to measurements
- Examples: TDOA readings or SRP output
- Noise statistic is unknown





Data Model

Microphone signals:

The signal measured by the jth microphone in the mth node:

$$y^{mj}(t) = \sum_{\tau} a_t^{mj}(\tau) s(t-\tau) + u^{mj}(t), \quad 1 \leq m \leq M, \; j = 1, 2$$

- t - time index
- $s(t)$ - source signal
- a_t^{mj} - time-varying acoustic impulse response (AIR)
- $u^{mj}(t)$ - noise signal

Feature extraction:

- Use the RTF:

$$H^m(t,f) = \frac{A^{m2}(t,f)}{A^{m1}(t,f)}$$

- Represents the acoustic paths and is independent of the source signal

Time-Varying Relative Transfer Function (RTF)

- Instantaneous RTFs are estimated using the PSD and cross-PSD of the microphone signals at node m (low-noise):

$$\hat{H}_0^m(t,f) \simeq \frac{\hat{\Phi}_{21}^m(t,f)}{\hat{\Phi}_{11}^m(t,f)} = \frac{\sum_{n=t-L/2}^{t+L/2} Y^{m2}(n,f)Y^{m1*}(n,f)}{\sum_{n=t-L/2}^{t+L/2} Y^{m1}(n,f)Y^{m1*}(n,f)}$$

- Time-varying RTFs are estimated by recursive smoothing:

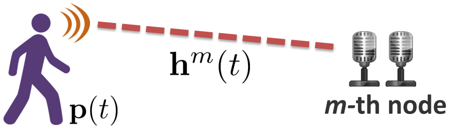
$$\hat{H}^m(t,f) = \gamma\hat{H}_0^m(t,f) + (1-\gamma)\hat{H}^m(t-1,f)$$

- Feature vectors are obtained by concatenating all relevant frequencies and all nodes:

$$\mathbf{h}^m(t) = \left[\hat{H}^m(t,f_1),\ldots,\hat{H}^m(t,f_F)\right]$$

$$\mathbf{h}(t) = \left[\mathbf{h}^{1T}(t),\ldots,\mathbf{h}^{MT}(t)\right]^T$$

Time-Varying Relative Transfer Function (RTF) (cont.)



- We assume the availability of n_L labelled RTFs with known positions:

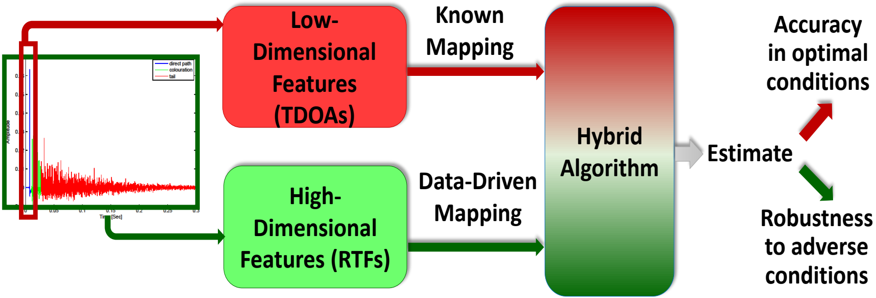
$$\{\mathbf{h}\}_{i=1}^{n_L} \Leftrightarrow \{\mathbf{p}\}_{i=1}^{n_L}$$

- These training RTFs can be estimated with static sources, hence a long observation interval L can be used and the recursive smoothing is not required

Hybrid Tracking [Laufer-Goldshtein et al., 2018b]

Combine TDOA-based approach with manifold-based approach:

- Manifold-based propagation model (non-arbitrary)
- TDOA-based observation model
- Combines Classical TDOA-based localization with the entire acoustic fingerprint



Derivation of the Manifold-Based Propagation Model

- Let $\mathbf{h}(t)$ be a test sample with unknown position $\mathbf{p}(t)$
- Define a subset of $N \leq n_L$ neighboring training samples $\{\mathbf{h}_{t_i}\}_{i=1}^{N}$:

$$\{\mathbf{h}_{t_i} | \|\mathbf{h}(t) - \mathbf{h}_{t_i}\| < \eta(N),\ i = 1, \dots, N,\ t_i \in \{1, \dots, n_L\}\}$$

  with $\eta(N)$ the neighborhood radius
- Let $\mathbf{f}_{t,c} = [f_c(\mathbf{h}(t)), f_c(\mathbf{h}_{t_1}), \dots, f_c(\mathbf{h}_{t_N})]^T$ denote their positions, with $c \in \{x, y, z\}$
- Joint normal distribution for $\mathbf{f}_{t,c}$ and $\mathbf{f}_{t-1,c}$:

$$\begin{bmatrix} \mathbf{f}_{t,c} \\ \mathbf{f}_{t-1,c} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}_{2(N+1)}, \begin{bmatrix} \boldsymbol{\Sigma}_{t,t} & \boldsymbol{\Sigma}_{t,t-1} \\ \boldsymbol{\Sigma}_{t,t-1}^T & \boldsymbol{\Sigma}_{t-1,t-1} \end{bmatrix}\right)$$

Derivation of the Manifold-Based Propagation Model (cont.)

- The elements of $\boldsymbol{\Sigma}_{t,\tau}$ are given by the multiple manifold kernel:

$$\mathrm{cov}(f(\mathbf{h}_r), f(\mathbf{h}_l)) \equiv \tilde{k}(\mathbf{h}_r, \mathbf{h}_l) = \frac{1}{M^2} \sum_{q,w=1}^{M} \sum_{i=1}^{n_L} k_q(\mathbf{h}_r^q, \mathbf{h}_i^q) k_w(\mathbf{h}_l^w, \mathbf{h}_i^w)$$

- The conditional probability is then given by:

$$\mathrm{Pr}\Big(\mathbf{f}_{t,c} | \mathbf{f}_{t-1,c}\Big) = \mathcal{N}\Big(\mathbf{A}_t \mathbf{f}_{t-1,c}, \mathbf{Q}_t\Big)$$

where

$$\mathbf{A}_t = \boldsymbol{\Sigma}_{t,t-1} \boldsymbol{\Sigma}_{t-1,t-1}^{-1}$$
$$\mathbf{Q}_t = \boldsymbol{\Sigma}_{t,t} - \boldsymbol{\Sigma}_{t,t-1} \boldsymbol{\Sigma}_{t-1,t-1}^{-1} \boldsymbol{\Sigma}_{t,t-1}^{T}$$

Derivation of the Manifold-Based Propagation Model (cont.)

- The conditional probability induces a linear propagation equation:

$$\mathbf{f}_{t,c} = \mathbf{A}_t \mathbf{f}_{t-1,c} + \boldsymbol{\xi}_t$$

  where $\boldsymbol{\xi}_t \sim \mathcal{N}\left(\mathbf{0}_{N+1}, \mathbf{Q}_t\right)$

- The propagation matrix $\mathbf{A}_t$ and the covariance of the innovation noise $\mathbf{Q}_t$ are time-varying and inferred from the manifold based on the previous and current RTFs and their associated neighbors:

$$\mathbf{h}(t-1), \{\mathbf{h}_{(t-1)_i}\}_{i=1}^{N}, \mathbf{h}(t), \{\mathbf{h}_{t_i}\}_{i=1}^{N}$$

- The position estimate of the test sample $f_c(\mathbf{h}(t))$ is propagated from the previous position estimate, as well as the set of previous neighborhood of the training samples, using the matrices $\mathbf{A}_t$ and $\mathbf{Q}_t$

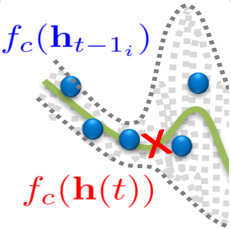
Derivation of the Manifold-Based Propagation Model (cont.)

The full propagation model for the 3-D position

Let $\mathbf{f}_t = \left[\mathbf{f}_{t,x}^T, \mathbf{f}_{t,y}^T, \mathbf{f}_{t,z}^T\right]^T$:

$$\mathbf{f}_t = \mathbf{A}_{3t}\mathbf{f}_{t-1} + \xi_{3t}$$

where $\mathbf{A}_{3t} = \mathbf{A}_t \otimes \mathbf{I}_3$ and $\xi_{3t} \sim \mathcal{N}\left(\mathbf{0}_{3(N+1)}, \mathbf{Q}_{3t}\right)$ with $\mathbf{Q}_{3t} = \mathbf{Q}_t \otimes \mathbf{I}_3$



TDOA-based observation Model

TDOA-based observations:

- Define observations as range differences:

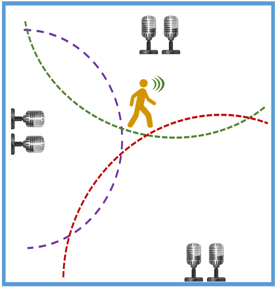
$$\boldsymbol{r} = \left[r^1, \ldots, r^M\right]^T$$

- Known nonlinear relation to the source position (requires microphones' positions):

$$r^m = g\left(\mathbf{p}\right) = \left\|\mathbf{p} - \mathbf{q}^{m2}\right\|_2 - \left\|\mathbf{p} - \mathbf{q}^{m1}\right\|_2$$

- The range differences can be extracted from the estimated RTFs [Dvorkind and Gannot, 2005]:

$$\hat{r}^m(t) = \frac{1}{c} \operatorname*{argmax}_{\tau} \hat{h}^m(t,\tau) \equiv \mathrm{IDFT}\left\{\hat{H}^m(t,k)\right\}$$



TDOA-Based Observation Model

- A nonlinear observation model is formed by:

$$\hat{\mathbf{r}}_t = \mathbf{g}(\mathbf{f}_t) + \zeta_t$$

where $\mathbf{g}(\mathbf{f}_t) = [\mathbf{g}^T(\mathbf{p}(t)), \mathbf{g}^T(\mathbf{p}_{t_1}), \ldots, \mathbf{g}^T(\mathbf{p}_{t_N})]^T$ and

$$\mathbf{g}(\mathbf{p}) = \begin{bmatrix} \left\|\mathbf{p} - \mathbf{q}^{12}\right\|_2 - \left\|\mathbf{p} - \mathbf{q}^{11}\right\|_2 \\ \vdots \\ \left\|\mathbf{p} - \mathbf{q}^{M2}\right\|_2 - \left\|\mathbf{p} - \mathbf{q}^{M1}\right\|_2 \end{bmatrix}$$

and $\zeta_t \sim \mathcal{N}\left(\mathbf{0}_{M(N+1)}, \mathbf{R}_t\right)$ is the observation error

- Linearization of the observation model (Extended Kalman filter - EKF [Smith et al., 1962]):

$$\nabla_{\mathbf{f}}\mathbf{g}(\mathbf{f}_t) = \mathrm{blkdiag}\left\{\nabla_{\mathbf{p}}\mathbf{g}(\mathbf{p}(t)), \nabla_{\mathbf{p}}\mathbf{g}(\mathbf{p}_{t_1}), \ldots, \nabla_{\mathbf{p}}\mathbf{g}(\mathbf{p}_{t_N})\right\}$$

Tracking Algorithm

Space-state representation:

$$\mathbf{f}_t = \mathbf{A}_{3t}\mathbf{f}_{t-1} + \boldsymbol{\xi}_{3t}$$
$$\hat{\mathbf{r}}_t = \mathbf{g}(\mathbf{f}_t) + \zeta_t$$

EKF: Additional notations

- $\hat{\mathbf{f}}(t|t)$ - The estimate of $\mathbf{f}_t$ based on measurements up to time t
- $\mathbf{\Pi}(t|t)$ - The associated error covariance matrix
- $\mathbf{G}_t = \nabla_{\mathbf{f}}\mathbf{g}(\hat{\mathbf{f}}(t|t-1))$ - linearized measurement matrix
- $\mathbf{R}_t$ - Measurement noise (diagonal) covariance matrix, which is significantly lower for the training samples, since their position is known
- $\mathbf{\Gamma}(t)$ - Kalman gain

Tracking Algorithm (cont.)

Extended Kalman Filter

Time Update

- Predicted Position:

$$\hat{\mathbf{f}}(t|t-1) = \mathbf{A}_{3t}\hat{\mathbf{f}}(t-1|t-1)$$

- Predicted Covariance:

$$\mathbf{\Pi}(t|t-1) = \mathbf{A}_{3t}\mathbf{\Pi}(t-1|t-1)\mathbf{A}_{3t}^T + \mathbf{Q}_{3t}$$

Measurement Update

- Kalman Gain:

$$\mathbf{\Gamma}(t) = \mathbf{\Pi}(t|t-1)\mathbf{G}_t^T\left(\mathbf{G}_t\mathbf{\Pi}(t|t-1)\mathbf{G}_t^T + \mathbf{R}_t\right)^{-1}$$

- Updated position estimate:

$$\hat{\mathbf{f}}(t|t) = \hat{\mathbf{f}}(t|t-1) + \mathbf{\Gamma}(t)\left(\hat{\mathbf{r}}_t - \mathbf{g}\left(\hat{\mathbf{f}}(t|t-1)\right)\right)$$

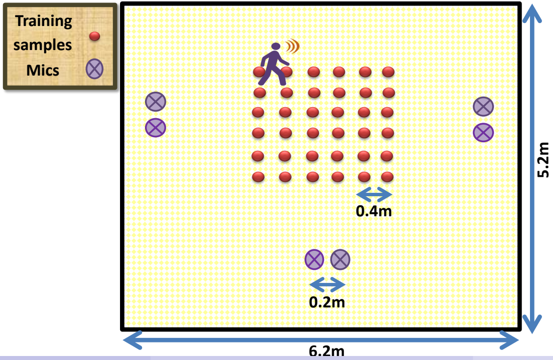
- Updated Covariance:

$$\mathbf{\Pi}(t|t) = \left(\mathbf{I}_{3(N+1)} - \mathbf{\Gamma}(t)\mathbf{G}_t\right)\mathbf{\Pi}(t|t-1)$$

Experimental Results

Setup:

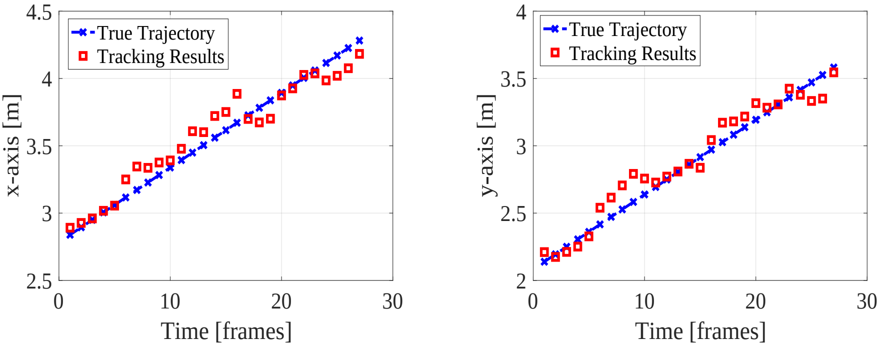
- A $5.2 \times 6.2 \times 3$m room with $T_{60} = 300$ms
- $M = 4$ nodes with 0.2m distance between microphones
- Region of interest: a 2×2m square region
- Training: 36 samples (0.4m resolution)



Results

Test I:

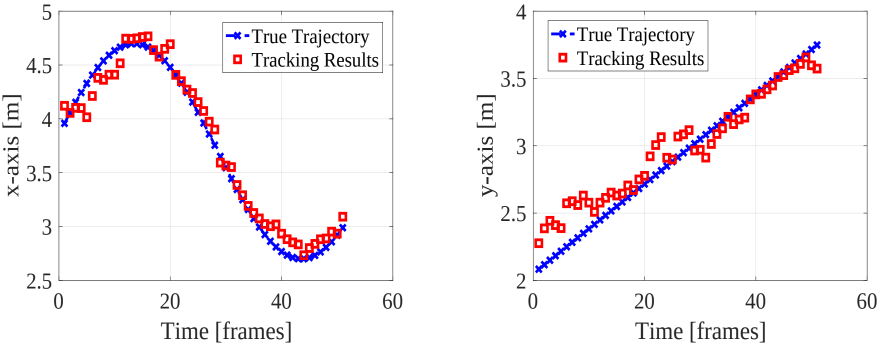
- Trajectory: straight line (for 3s)
- Velocity: approximately 1m/s



Results

Test II:

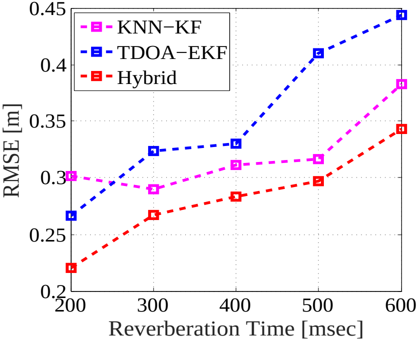
- Trajectory: sinusoid (for 5s)
- Velocity: approximately 1m/s

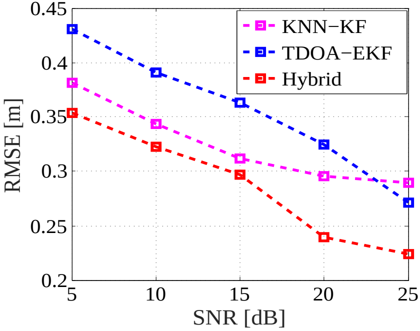


Results

Compare with:

- TDOA-based tracker ('TDOA-EKF') [Gannot and Dvorkind, 2006]: random walk propagation model
- Learning-based approach ('KNN-KF') [Wang and Chaib-Draa, 2013]: linear observation model of labelled positions

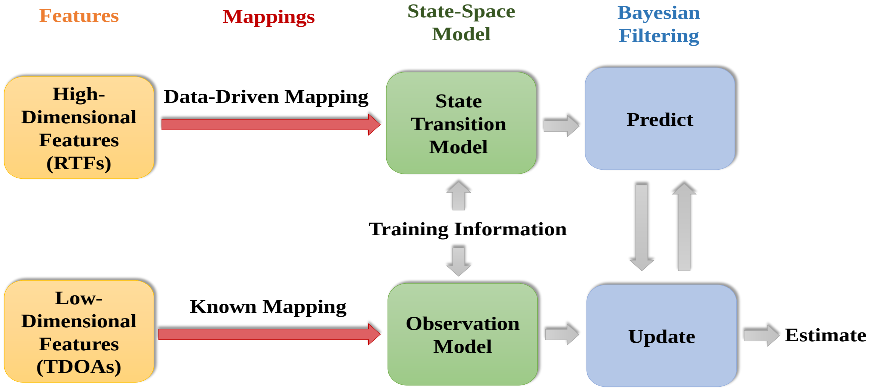


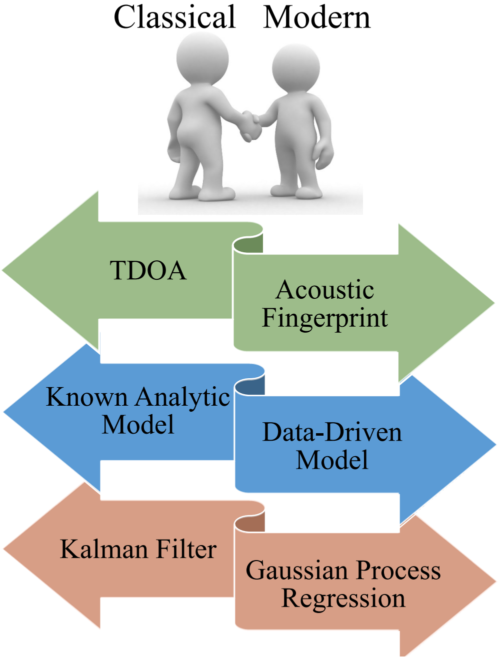


Combining Data Modalities

Combine two data modalities of different types:

- High-dimensional features - data-driven model with acoustic fingerprints
- Low dimensional features - known physical model (TDOA-based)




Classical
Modern
TDOA
Acoustic Fingerprint
Known Analytic Model
Data-Driven Model
Kalman Filter
Gaussian Process Regression

Conclusions

Summary

- Manifold learning approach for source localization
- Data-driven manifold inference
- Location is the controlling variable of the RTF manifold
- Devise algorithms for source localization and tracking using either regularized optimization or Bayesian inference
  - Presents data fusion of several manifolds
  - Dynamics of the source are learned from the variations of the corresponding RTFs on the manifold
- Data-driven, training-based approach, was successfully applied to real-life recordings
- The dynamics on the manifold can be transformed to linear propagation for the source moving in tracking scenarios

Challenges and Perspectives

Challenges

- Robustness to environmental changes:
  - Mismatch between train and test
  - Movements
- Can we apply the approach to multiple concurrent speakers?
- Beamforming is more complicated as it targets enhanced speech rather than its location. Can we extend the approach?
  - A first attempt using projections to the inferred manifold [Talmon and Gannot, 2013]

References

[Adavanne et al., 2018] Adavanne, S., Politis, A., and Virtanen, T. (2018).
Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network.
In *Proc. of European Signal Processing Conference (EUSIPCO)*, pages 1462–1466.

[Affes and Grenier, 1997] Affes, S. and Grenier, Y. (1997).
A signal subspace tracking algorithm for microphone array processing of speech.
IEEE Transactions on Speech and Audio Processing, 5(5):425–437.

[Allen and Berkley, 1979] Allen, J. B. and Berkley, D. A. (1979).
Image method for efficiently simulating small-room acoustics.
The Journal of the Acoustical Society of America, 65(4):943–950.

[Aronszajn, 1950] Aronszajn, N. (1950).
Theory of reproducing kernels.
Transactions of the American mathematical society, 68(3):337–404.

[Ban et al., 2019] Ban, Y., Alameda-Pineda, X., Evers, C., and Horaud, R. (2019).
Tracking multiple audio sources with the von Mises distribution and variational EM.
IEEE Signal Processing Letters, 26(6):798–802.

[Belkin and Niyogi, 2003] Belkin, M. and Niyogi, P. (2003).
Laplacian eigenmaps for dimensionality reduction and data representation.
Neural computation, 15(6):1373–1396.

[Belkin et al., 2006] Belkin, M., Niyogi, P., and Sindhwani, V. (2006).
Manifold regularization: A geometric framework for learning from labeled and unlabeled examples.
Journal of machine learning research, 7(Nov):2399–2434.

[Benesty, 2000] Benesty, J. (2000).
Adaptive eigenvalue decomposition algorithm for passive acoustic source localization.
The Journal of the Acoustical Society of America, 107(1):384–391.

References (cont.)

[Bérard et al., 1994] Bérard, P., Besson, G., and Gallot, S. (1994).
Embedding riemannian manifolds by their heat kernel.
Geometric & Functional Analysis GAFA, 4(4):373–398.

[Berlinet and Thomas-Agnan, 2011] Berlinet, A. and Thomas-Agnan, C. (2011).
Reproducing kernel Hilbert spaces in probability and statistics.
Springer Science & Business Media.

[Brandstein et al., 1997] Brandstein, M. S., Adcock, J. E., and Silverman, H. F. (1997).
A closed-form location estimator for use with room environment microphone arrays.
IEEE transactions on Speech and Audio Processing, 5(1):45–50.

[Brendel et al., 2018] Brendel, A., Gannot, S., and Kellermann, W. (2018).
Localization of multiple simultaneously active speakers in an acoustic sensor network.
In *IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Sheffield, United Kingdom (Great Britain).

[Brendel and Kellermann, 2019] Brendel, A. and Kellermann, W. (2019).
Distributed source localization in acoustic sensor networks using the coherent-to-diffuse power ratio.
IEEE Journal of Selected Topics in Signal Processing, 13(1):61–75.

[Chakrabarty and Habets, 2019] Chakrabarty, S. and Habets, E. A. (2019).
Multi-speaker doa estimation using deep convolutional networks trained with noise signals.
IEEE Journal of Selected Topics in Signal Processing.

[Coifman and Lafon, 2006] Coifman, R. R. and Lafon, S. (2006).
Diffusion maps.
Applied and Computational Harmonic Analysis, 21(1):5–30.

[Dal Degan and Prati, 1988] Dal Degan, N. and Prati, C. (1988).
Acoustic noise analysis and speech enhancement techniques for mobile radio applications.
Signal Processing, 15(1):43–56.

References (cont.)

[Deleforge et al., 2013] Deleforge, A., Forbes, F., and Horaud, R. (2013).
Variational EM for binaural sound-source separation and localization.
In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 76–80.

[Deleforge et al., 2015] Deleforge, A., Forbes, F., and Horaud, R. (2015).
Acoustic space learning for sound-source separation and localization on binaural manifolds.
International journal of neural systems, 25(1).

[Deleforge and Horaud, 2012] Deleforge, A. and Horaud, R. (2012).
2D sound-source localization on the binaural manifold.
In *Proc. of IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Santander, Spain.

[DiBiase et al., 2001] DiBiase, J. H., Silverman, H. F., and Brandstein, M. S. (2001).
Robust localization in reverberant rooms.
In *Microphone Arrays*, pages 157–180. Springer.

[Do et al., 2007] Do, H., Silverman, H. F., and Yu, Y. (2007).
A real-time SRP-PHAT source location implementation using stochastic region contraction (SRC) on a large-aperture microphone array.
In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 121–124.

[Doclo and Moonen, 2003] Doclo, S. and Moonen, M. (2003).
Robust adaptive time delay estimation for speaker localization in noisy and reverberant acoustic environments.
EURASIP Journal on Applied Signal Processing, 2003:1110–1124.

[Dorfan and Gannot, 2015] Dorfan, Y. and Gannot, S. (2015).
Tree-based recursive expectation-maximization algorithm for localization of acoustic sources.
IEEE/ACM Transactions on Audio, Speech and Language Processing, 23(10):1692–1703.

References (cont.)

[Dorfan et al., 2018] Dorfan, Y., Plinge, A., Hazan, G., and Gannot, S. (2018).
Distributed expectation-maximization algorithm for speaker localization in reverberant environments.
IEEE/ACM Transactions on Audio, Speech and Language Processing, 26(3):682–695.

[Dorfan et al., 2016] Dorfan, Y., Schwartz, O., Schwartz, B., Habets, E. A., and Gannot, S. (2016).
Multiple DOA estimation and blind source separation using expectation-maximization algorithm.
In *International conference on the science of electrical engineering (ICSEE)*, Eilat, Israel.

[Dvorkind and Gannot, 2005] Dvorkind, T. and Gannot, S. (2005).
Time difference of arrival estimation of speech source in a noisy and reverberant environment.
Signal Processing, 85(1):177–204.

[Evers and Naylor, 2017] Evers, C. and Naylor, P. A. (2017).
Optimized self-localization for slam in dynamic scenes using probability hypothesis density filters.
IEEE Transactions on Signal Processing, 66(4):863–878.

[Faubel et al., 2009] Faubel, F., McDonough, J., and Klakow, D. (2009).
The split and merge unscented gaussian mixture filter.
IEEE Signal Processing Letters, 16(9):786–789.

[Fourier, 1822] Fourier, B. J. B. J. (1822).
Théorie analytique de la chaleur.
F. Didot.

[Gannot et al., 2001] Gannot, S., Burshtein, D., and Weinstein, E. (2001).
Signal enhancement using beamforming and nonstationarity with applications to speech.
IEEE Transactions on Signal Processing, 49(8):1614–1626.

[Gannot and Dvorkind, 2006] Gannot, S. and Dvorkind, T. G. (2006).
Microphone array speaker localizers using spatial-temporal information.
EURASIP Journal on Advances in Signal Processing, 2006(1):1–17.

References (cont.)

[Habets and Gannot, 2007] Habets, E. and Gannot, S. (2007).
Generating sensor signals in isotropic noise fields.
The Journal of the Acoustical Society of America, 122:3464–3470.

[Hadad and Gannot, 2018] Hadad, E. and Gannot, S. (2018).
Multi-speaker direction of arrival estimation using SRP-PHAT algorithm with a weighted histogram.
In *IEEE International Conference on the Science of Electrical Engineering in Israel (ICSEE)*.

[Hu et al., 2019] Hu, Y., Samarasinghe, P. N., and Abhayapala, T. D. (2019).
Sound source localization using relative harmonic coefficients in modal domain.
In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA.

[Huang et al., 2000] Huang, Y., Benesty, J., and Elko, G. W. (2000).
Passive acoustic source localization for video camera steering.
In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 909–912.

[Huang et al., 2001] Huang, Y., Benesty, J., Elko, G. W., and Mersereati, R. M. (2001).
Real-time passive source localization: A practical linear-correction least-squares approach.
IEEE transactions on Speech and Audio Processing, 9(8):943–956.

[Jot et al., 1997] Jot, J.-M., Cerveau, L., and Warusfel, O. (1997).
Analysis and synthesis of room reverberation based on a statistical time-frequency model.
In *Audio Engineering Society Convention 103*. Audio Engineering Society.

[Klee et al., 2006] Klee, U., Gehrig, T., and McDonough, J. (2006).
Kalman filters for time delay of arrival-based source localization.
EURASIP Journal on Applied Signal Processing, 2006:167–167.

[Knapp and Carter, 1976] Knapp, C. and Carter, G. (1976).
The generalized correlation method for estimation of time delay.
IEEE Transactions on Acoustics, Speech, and Signal Processing, 24(4):320–327.

References (cont.)

[Koldovký et al., 2015] Koldovký, Z., Malek, J., and Gannot, S. (2015).
Spatial source subtraction based on incomplete measurements of relative transfer function.
IEEE/ACM Transactions on Audio, Speech and Language Processing, pages 1335–1347.

[Kruskal, 1964] Kruskal, J. B. (1964).
Nonmetric multidimensional scaling: a numerical method.
Psychometrika, 29(2):115–129.

[Laufer-Goldshtein et al., 2018a] Laufer-Goldshtein, B., Talmon, R., Cohen, I., and Gannot, S. (2018a).
Multi-view source localization based on power ratios.
In *IEEE International Conference on Audio and Acoustic Signal Processing (ICASSP)*, Calgary, Alberta, Canada.

[Laufer-Goldshtein et al., 2013] Laufer-Goldshtein, B., Talmon, R., and Gannot, S. (2013).
Relative transfer function modeling for supervised source localization.
In *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 1–4.

[Laufer-Goldshtein et al., 2015] Laufer-Goldshtein, B., Talmon, R., and Gannot, S. (2015).
Study on manifolds of acoustic responses.
In *Proc. of Interntional Conference on Latent Variable Analysis and Signal Seperation (LVA/ICA)*, pages 203–210.

[Laufer-Goldshtein et al., 2016a] Laufer-Goldshtein, B., Talmon, R., and Gannot, S. (2016a).
Manifold-based Bayesian inference for semi-supervised source localization.
In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6335–6339.

[Laufer-Goldshtein et al., 2016b] Laufer-Goldshtein, B., Talmon, R., and Gannot, S. (2016b).
Semi-supervised sound source localization based on manifold regularization.
IEEE Transactions on Audio, Speech, and Language Processing, 24(8):1393–1407.

[Laufer-Goldshtein et al., 2017] Laufer-Goldshtein, B., Talmon, R., and Gannot, S. (2017).
Semi-supervised source localization on multiple-manifolds with distributed microphones.
IEEE/ACM Transactions on Audio, Speech, and Language Processing, 25(7):1477–1491.

References (cont.)

[Laufer-Goldshtein et al., 2018b] Laufer-Goldshtein, B., Talmon, R., and Gannot, S. (2018b).
A hybrid approach for speaker tracking based on tdoa and data-driven models.
IEEE/ACM Transactions on Audio, Speech and Language Processing, 26(4):725–735.

[Laufer-Goldshtein et al., 2018c] Laufer-Goldshtein, B., Talmon, R., and Gannot, S. (2018c).
Source counting and separation based on simplex analysis.
IEEE Transactions on Signal Processing, 66(24):6458–6473.

[Lederman and Talmon, 2018] Lederman, R. R. and Talmon, R. (2018).
Learning the geometry of common latent variables using alternating-diffusion.
Applied and Computational Harmonic Analysis, 44(3):509–536.

[Lehmann and Williamson, 2006] Lehmann, E. A. and Williamson, R. C. (2006).
Particle filter design using importance sampling for acoustic source localisation and tracking in reverberant environments.
EURASIP Journal on Applied Signal Processing, 2006:168–168.

[Levy et al., 2011] Levy, A., Gannot, S., and Habets, E. A. (2011).
Multiple-hypothesis extended particle filter for acoustic source localization in reverberant environments.
IEEE Transactions on Audio, Speech, and Language Processing, 19(6):1540–1555.

[Löllmann et al., 2018] Löllmann, H. W., Evers, C., Schmidt, A., Mellmann, H., Barfuss, H., Naylor, P. A., and Kellermann, W. (2018).
The locata challenge data corpus for acoustic source localization and tracking.
In *IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 410–414.

[Madhu and Martin, 2018] Madhu, N. and Martin, R. (2018).
Source number estimation for multi-speaker localisation and tracking.
In *WSPD2018, the Workshop on speech processing for voice, speech and hearing disorders*, pages 1–5.

References (cont.)

[Madhu et al., 2008] Madhu, N., Martin, R., Heute, U., and Antweiler, C. (2008).
Acoustic source localization with microphone arrays.
Advances in Digital Speech Transmission, pages 135–170.

[Mandel et al., 2007] Mandel, M. I., Ellis, D. P., and Jebara, T. (2007).
An EM algorithm for localizing multiple sound sources in reverberant environments.
In *Proc. of Advances in neural information processing systems*, pages 953–960.

[Mandel et al., 2010] Mandel, M. I., Weiss, R. J., and Ellis, D. P. W. (2010).
Model-based expectation-maximization source separation and localization.
IEEE Transactions on Audio, Speech, and Language Processing, 18(2):382–394.

[Markovich et al., 2009] Markovich, S., Gannot, S., and Cohen, I. (2009).
Multichannel eigenspace beamforming in a reverberant noisy environment with multiple interfering speech signals.
IEEE Transactions on Audio, Speech, and Language Processing, 17(6):1071–1086.

[Markovich-Golan et al., 2010] Markovich-Golan, S., Gannot, S., and Cohen, I. (2010).
Subspace tracking of multiple sources and its application to speakers extraction.
In *The IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 201–204, Dallas, Texas, USA.

[Markovich-Golan et al., 2018] Markovich-Golan, S., Gannot, S., and Kellermann, W. (2018).
Performance analysis of the Covariance-Whitening and the Covariance-Subtraction methods for estimating the relative transfer function.
In *The 26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy.

[May et al., 2011] May, T., van de Par, S., and Kohlrausch, A. (2011).
A probabilistic model for robust localization based on a binaural auditory front-end.
IEEE Transactions on Audio, Speech, and Language Processing, 19(1):1–13.

References (cont.)

[Opochinsky et al., 2019] Opochinsky, R., Laufer, B., Gannot, S., and Chechik, G. (2019).
Deep Ranking-Based sound source localization.
In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA.

[Pearson, 1901] Pearson, K. (1901).
Principal components analysis.
The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 6(2):559.

[Perotin et al., 2019] Perotin, L., Serizel, R., Vincent, E., and Guerin, A. (2019).
Crnn-based multiple doa estimation using acoustic intensity features for ambisonics recordings.
IEEE Journal of Selected Topics in Signal Processing.

[Peterson, 1986] Peterson, P. M. (1986).
Simulating the response of multiple microphones to a single acoustic source in a reverberant room.
The Journal of the Acoustical Society of America, 80(5):1527–1529.

[Polack, 1993] Polack, J.-D. (1993).
Playing billiards in the concert hall: The mathematical foundations of geometrical room acoustics.
Applied Acoustics, 38(2):235–244.

[Press et al., 2007] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007).
Numerical recipes 3rd edition: The art of scientific computing.
Cambridge university press.

[Rasmussen and Williams, 2006] Rasmussen, C. E. and Williams, C. K. I. (2006).
Gaussian processes for machine learning.
MIT Press.

[Reindl et al., 2013] Reindl, K., Markovich-Golan, S., Barfuss, H., Gannot, S., and Kellermann, W. (2013).
Geometrically constrained TRINICON-based relative transfer function estimation in underdetermined scenarios.
In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA.

References (cont.)

[Reuven et al., 2008] Reuven, G., Gannot, S., and Cohen, I. (2008).
Dual-source transfer-function generalized sidelobe canceller.
IEEE transactions on audio, speech, and language processing, 16(4):711–727.

[Roman et al., 2003] Roman, N., Wang, D., and Brown, G. J. (2003).
Speech segregation based on sound localization.
The Journal of the Acoustical Society of America, 114(4):2236–2252.

[Roweis and Saul, 2000] Roweis, S. T. and Saul, L. K. (2000).
Nonlinear dimensionality reduction by locally linear embedding.
science, 290(5500):2323–2326.

[Roy and Kailath, 1989] Roy, R. and Kailath, T. (1989).
ESPRIT-estimation of signal parameters via rotational invariance techniques.
IEEE Transactions on Acoustics, Speech and Signal Processing, 37(7):984–995.

[Schau and Robinson, 1987] Schau, H. and Robinson, A. (1987).
Passive source localization employing intersecting spherical surfaces from time-of-arrival differences.
IEEE Transactions on Acoustics, Speech, and Signal Processing, 35(8):1223–1225.

[Schmidt, 1986] Schmidt, R. O. (1986).
Multiple emitter location and signal parameter estimation.
IEEE Transactions on Antennas and Propagation, 34(3):276–280.

[Schölkopf et al., 2001] Schölkopf, B., Herbrich, R., and Smola, A. J. (2001).
A generalized representer theorem.
In *Proc. of The 14th Annual Conference on Computational learning theory (COLT)*, pages 416–426. Springer.

[Schroeder, 1996] Schroeder, M. R. (1996).
The "schroeder frequency" revisited.
The Journal of the Acoustical Society of America, 99(5):3240–3241.

References (cont.)

[Schwartz et al., 2017] Schwartz, O., Dorfan, Y., Taseska, M., Habets, E. A., and Gannot, S. (2017).
DOA estimation in noisy environment with unknown noise power using the EM algorithm.
In *The 5th Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*, San-Francisco, CA, USA.

[Schwartz and Gannot, 2013] Schwartz, O. and Gannot, S. (2013).
Speaker tracking using recursive EM algorithms.
IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22(2):392–402.

[Shalvi and Weinstein, 1996] Shalvi, O. and Weinstein, E. (1996).
System identification using nonstationary signals.
IEEE transactions on signal processing, 44(8):2055–2063.

[Sindhwani et al., 2007] Sindhwani, V., Chu, W., and Keerthi, S. S. (2007).
Semi-supervised Gaussian process classifiers.
In *Proc. of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1059–1064.

[Smith et al., 1962] Smith, G. L., Schmidt, S. F., and McGee, L. A. (1962).
Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle.

[Smith and Abel, 1987] Smith, J. and Abel, J. (1987).
Closed-form least-squares source location estimation from range-difference measurements.
IEEE Transactions on Acoustics, Speech, and Signal Processing, 35(12):1661–1669.

[Soussana and Gannot, 2019] Soussana, Y. and Gannot, S. (2019).
Variational inference for DOA estimation in reverberant conditions.
In *27th European Signal Processing Conference (EUSIPCO)*, A Coruña, Spain.

[Takeda and Komatani, 2016] Takeda, R. and Komatani, K. (2016).
Sound source localization based on deep neural networks with directional activate function exploiting phase information.
In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 405–409.

References (cont.)

[Talmon and Gannot, 2013] Talmon, R. and Gannot, S. (2013).
Relative transfer function identification on manifolds for supervised gsc beamformers.
In *Proc. of 21st European Signal Processing Conference (EUSIPCO)*, pages 1–5.

[Talmon et al., 2011] Talmon, R., Kushnir, D., Coifman, R. R., Cohen, I., and Gannot, S. (2011).
Parametrization of linear systems using diffusion kernels.
IEEE Transactions on Signal Processing, 60(3):1159–1173.

[Tenenbaum et al., 2000] Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000).
A global geometric framework for nonlinear dimensionality reduction.
science, 290(5500):2319–2323.

[Teutsch and Kellermann, 2005] Teutsch, H. and Kellermann, W. (2005).
EB-ESPRIT: 2D localization of multiple wideband acoustic sources using eigen-beams.
In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages 89–92.

[Thiergart et al., 2014] Thiergart, O., Taseska, M., and Habets, E. A. P. (2014).
An informed parametric spatial filter based on instantaneous direction-of-arrival estimates.
IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22(12):2182–2196.

[Traa and Smaragdis, 2014] Traa, J. and Smaragdis, P. (2014).
Multichannel source separation and tracking with ransac and directional statistics.
IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22(12):2233–2243.

[Wang and Chaib-Draa, 2013] Wang, Y. and Chaib-Draa, B. (2013).
A KNN based Kalman filter Gaussian process regression.
In *Proc. of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1771–1777.

[Ward et al., 2003] Ward, D. B., Lehmann, E. A., and Williamson, R. C. (2003).
Particle filtering algorithms for tracking an acoustic source in a reverberant environment.
IEEE Transactions on speech and audio processing, 11(6):826–836.

References (cont.)

[Weisberg et al., 2019] Weisberg, K., Gannot, S., and Schwartz, O. (2019).
An online multiple-speaker DOA tracking using the Cappé-Moulines recursive expectation-maximization algorithm.
In *IEEE International Conference on Audio and Acoustic Signal Processing (ICASSP)*, pages 656–660.

[Xiao et al., 2015] Xiao, X., Zhao, S., Zhong, X., Jones, D. L., Chng, E. S., and Li, H. (2015).
A learning-based approach to direction of arrival estimation in noisy and reverberant environments.
In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 76–80.

[Yao et al., 2002] Yao, K., Chen, J. C., and Hudson, R. E. (2002).
Maximum-likelihood acoustic source localization: experimental results.
In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages 2949–2952.

[Yilmaz and Rickard, 2004] Yilmaz, O. and Rickard, S. (2004).
Blind separation of speech mixtures via time-frequency masking.
IEEE Transactions on Signal Processing, 52(7):1830–1847.

[Zhong and Hopgood, 2008] Zhong, X. and Hopgood, J. R. (2008).
Nonconcurrent multiple speakers tracking based on extended Kalman particle filter.
In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 293–296.