Kernel Characterization

A function \( K: X \times X \Rightarrow \mathbb{R} \)
Which is either continuous or has a countable domain, can be decomposed \( K(x,z) = \langle \phi(x), \phi(y) \rangle \)
into a feature map \( \phi \) into a Hilbert space \( F \) applied to both its arguments followed by the evaluation of the inner product in \( F \) if and only if it satisfies the finitely positive semi-definite property. (Shawe-Taylor, Cristianini 2004)
Computing Kernels

Input space

Feature space

Direct

Kernels

$K(x, y) = \langle \phi(x), \phi(y) \rangle$
Constructing Hilbert spaces from Kernels

- Construct a linear vector space with an inner product from kernels. Let \( f(x) \), \( g(x) \) be defined by
  \[
  f(x) = \sum_i \alpha(i)K(x(i),x) \quad \text{and} \quad g(x) = \sum_j \beta(j)K(z(j),x)
  \]

Define inner product by

\[
<f,g> = \sum_{i,j} \alpha(i)\beta(j) K(x(i),z(j))
\]

Can easily show that this is a valid inner product.

- Completeness and separability

- Reproducing property: (RKHS)
  \[
  <f,K(x,\cdot)> = \sum_i \alpha(i)K(x(i),x) = f(x)
  \]
Example 2: Sensor Network Localization

- **Applications**
  - Indoor environmental control
  - Security system
  - Context-aware computing
  - Environmental monitoring
  - Wild animal tracking

- **Location information needed to correctly interpret sensed data**
Categorization and Solution Methods

- Range-free methods
  - No ranging devices
  - Other information like topology
- Range-based methods
  - Ranging devices necessary
  - Small and cheap preferred but noisy

Statistical methods to improve range-based methods

- Multilateration
- Maximum likelihood estimation
- EM algorithm
- Bayesian networks
- Kernel methods
Sensor Network

known location
unknown location
Problem statement

- Given: A sensor network of size $n$
  - First $m$ base sensors have known locations
    $$X_1 = x_1, X_2 = x_2, \ldots X_m = x_m$$
  - Each sensor is able to receive and transmit signal strength to all other sensors $s(x_i, x_j), 1 \leq i, j \leq n$
- Find the locations of the remaining $(n-m)$ sensors
  $$X_{m+1} = ?, X_{m+2} = ?, \ldots X_n = ?$$
Kernel methods

Nguyen et al. proposed a kernel method (SVM)
  - Exploited the redundancy of raw data
  - Robust
  - Reasonable computational cost

Our methods
  - Retain advantages of Nguyen’s method
  - Reduced computational cost
  - On-line implementation
  - Suitable for large dense sensor networks
Classification using Kernel Methods

- Use kernel LS subspace algorithm to train on subset of known sensors. Classification algorithm determines if sensor is in $C$ or not. Can refine subspace choice by picking known sensors that are in vicinity of boundary of $C$.

- Test data is represented by unknown sensors. From decision region determine if unknown sensor is in region $C$ or not.

- Repeat training on all other regions and test unknown sensors.
Fine Localization Algorithm

Assume

* A set of regions $C$
* Each region $C$ has center $c(C)$
* Every location is covered by a subset of $C$
* Sensor $i$ is covered by a subset of regions $C_i$

We estimate the location of sensor $i$ as

$$x_i = \frac{\sum_{C \in C_i} c(C)}{|C_i|}$$
General Setup

* Sensor network deployed in a $10 \times 10$ region
* Signal Model

\[ s(x, x') = \exp \left\{ -\frac{\|x - x'\|^2}{\Sigma} + N(0, \tau) \right\} \]

* Gaussian Kernel

\[ K_{ij} = \exp \left\{ -\frac{\sum_{t=1}^{m} [s(x_i, x_t) - s(x_j, x_t)]^2}{2\sigma^2} \right\} \]
Coarse Localization Setup

* Region

\[ C = \{ x | (x - \nu)' H_1 (x - \nu) < R \text{ or } (x - \nu)' H_2 (x - \nu) < R \} \]

where

\[ H_1 = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}, \quad H_2 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \text{ and } \nu = \begin{bmatrix} 5 \\ 5 \end{bmatrix} \]

* \( m \times m \) base sensors on the lattice spanning the area
* \( 100 \times 100 \) uniformly distributed test locations
* \( \tau = 0 \)
* \( l_s = \sqrt{m} + 10 \)
Fine Localization Setup

* Perturb base sensor location by $N(0,10/(2\sqrt{m}))$
* $k^2$ disks with its center on a $k \times k$ lattice spanning the area
* 400 test locations
* $\tau = 0.2$
* Mean and variance obtained by running 20 simulations
Fine Localization Performance

K=15 and mS = m
Fine Localization Performance (cont.)

$\Sigma=2$, $m=100$, and $m_s=30$
Other Sensor localization research

- Analytically calculate average squared error of location versus L, k, R, m.
- Develop more computationally efficient algorithm to only choose base stations close to desired region C.
- Use regression methods to estimate location (complex valued network to give locations)
- Test algorithms on real ad hoc sensor networks.
- Assume unknown sensors are mobile and use on-line recursive LS subspace methods to track moving sensors
- Use kernel algorithms to process other environmental information from sensor networks