A kernel-based learning approach to ad hoc sensor network localization

XuanLong Nguyen, Michael I. Jordan and Bruno Sinopoli
Electrical Engineering and Computer Science
University of California, Berkeley
Jan, 2004

Sensor network localization

- Given a set of sensors $X_1, \ldots, X_m$, let x_i denote the position in R^2 of sensor X_i.
- Suppose that the locations of the first n sensors are known, i.e., $X_1 = x_1, \ldots, X_n = x_n$, where $n \ll m$.
- We want to recover the positions of $X_{n+1}, \ldots, X_m$.
- We want to do this solely on the basis of the receive/transmit signals $l(x_i, x_j)$ between pairs of sensors.

Existing approaches

- Most existing approaches consist of two steps:
  - Ranging: Estimate Euclidean distances from the signal matrix.
  - Localization: Estimate unknown sensor location by least square methods, or triangulation.
- Problems:
  - Ranging step often fails, due to inaccurate models unable to capture multipath effects, ambient noise interference, etc.

Departure from previous approaches

- A sensor network as a *distributed pattern recognition device*
  - Rather than transforming sensor readings into Euclidean, world-centric coordinates, work directly with the non-Euclidean coordinate system given by the physical sensor readings.
- The readings can be viewed as “kernel functions,” inducing a function space for prediction of various extrinsic quantities of interest.
- Kernel-based classification and regression algorithms from statistical learning theory can be used to form such predictors.

Classification problem and kernel methods

- Given n training data $(x_i, y_i)_{i=1}^n$ in $\mathcal{X} \times \{\pm 1\}$, where $\mathcal{X}$ denotes the space of sensor positions.
- Classification problem: Find a discriminant function $y = \mathrm{sign}(f(x))$ that minimizes $P(Y \neq \mathrm{sign}(f(X)))$.

Kernel-based classification algorithm (SVM)

- Kernel function $K(x, x')$ as a similarity measure for x and x'.
- K is a symmetric positive semidefinite kernel function.
  - If $K(x, x') = h(x - x')$ for some h, then the Fourier transform of h is non-negative.
- There exists a feature space $\{\Phi(x) | x \in \mathcal{X}\}$ in which K acts as an inner product, i.e., $K(x, x') = \langle \Phi(x), \Phi(x') \rangle$
- SVM algorithm searches for a hyperplane in the feature space, i.e. $f(x) = \langle \mathrm{w}, \Phi(x) \rangle$ that best separates the training data set $(x_i, y_i)_{i=1}^n$.

Kernel-based representation of $y = f(x)$

- It turns out that $f(x)$ has the following form:

$$f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x).$$

- If we know the kernel function K, we can compute the quantity $f(x)$ of interest.
- Examples of K:
  - Gaussian kernel: $K(x, x') = \exp -||x - x'||^2/\sigma$.
  - Polynomial kernel: $K(x, x') = (\gamma + ||x - x'||)^{-\sigma}$.
- Can we define kernel functions based on the sensor signal matrix?

Kernels based on sensor signal matrix

- 1. $K(x_i, x_j) = l(x_i, x_j)^\sigma$, for some σ.
  - This requires l to be a reasonable approximation to a positive semidefinite kernel function.
- 2. $K = L^T L$, where $L = (l(x_i, x_j))_{1 \le i,j \le m}$.
  - K is formed as inner products of the feature vectors $\{\Phi(x)\}$, defined as

$$\Phi(x_i) = (l(x_i, x_1), l(x_i, x_2), \ldots, l(x_i, x_m)).$$

- 3. There are a variety of positive semidefinite kernels defined on top of the feature space $\{\Phi(x)\}$.

Coarse localization as kernel-based classification

- Fix an arbitrarily constructed region $C \subseteq R^2$.
- Given base sensors $X_i = x_i$, construct n training data samples $(x_i, y_i = \text{sign}(x_i \in C))$.
- Invoke the SVM algorithm using the sensor-signal-based kernel matrix.
- For other sensors $j = n+1, \ldots, m$, we can predict whether $X_j \in C$ or not based on sign$(f(x_j))$ of the form:

$$f(x_j) = \sum_{i=1}^{n} \alpha_i K(x_i, x_j).$$

- Note that $f(x_j)$ can be computed *without knowledge of* x_j.

Fine-grained localization

- To locate the position x_j of X_j, fix a number of overlapping regions $C_\beta, for(\beta = 1, \ldots, U)$.
- For each β, formulate a classification problem and predict whether or not $X_j \in C_\beta$.
- Assign x_j to be a center of the intersection of all C_β predicted to contain X_j.
- Exploiting characteristics of our problem:
  - By picking appropriate regions C_β that are easy to classify using signal-based kernels, we can achieve good coarse localization.
  - Good coarse localization is turned into good fine-grained localization.
  - This intuition can be quantified by statistical learning theory.

Localization error analysis

- Network size $L \times L$, with n base sensors covered by discs of radius R.

Theorem 1. *Letting ϵ be the worst training error of classifying base sensors in these discs, then for any new sensor location x, our algorithm outputs a location estimate $\hat{x}$ that satisfies:*

$$\mathrm{E}||x - \hat{x}|| \leq O(L^{1/3} R^{2/3} (\epsilon + O(1/\sqrt{n}))^{1/3}).$$

Corollary 2. *Assume that the base sensor locations are linearly separable in the feature space $\{\Phi(x)\}$, i.e., $\epsilon = 0$, then the localization error is $O(L^{1/3} R^{2/3} n^{-1/6})$.*

- Usually, if R decreases, ϵ increases.

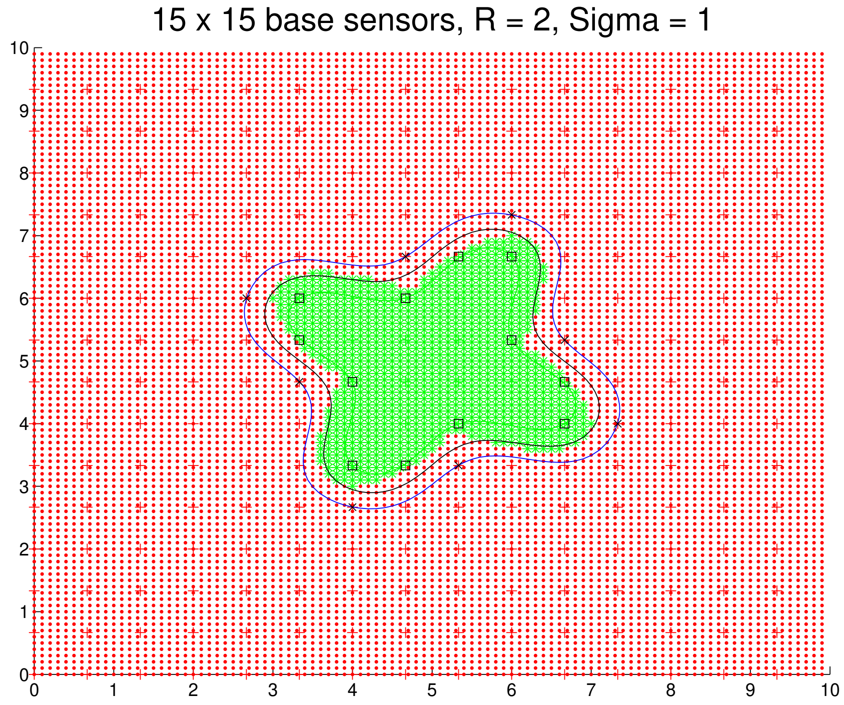
Computational cost

- Training phase is done at base stations and takes $O(n^3k^2)$ time, where k^2 is the number of covering discs.
  - In practice, the computation cost is dependent on the number of support vectors, which tends to be small. Simulation experience suggests a cost of only $O(R^2nk^2)$.
- Fine-grained localization is done at each individual sensor in linear time $O(nk^2)$.
- Each sensor talks to only the n base stations to locate its position.

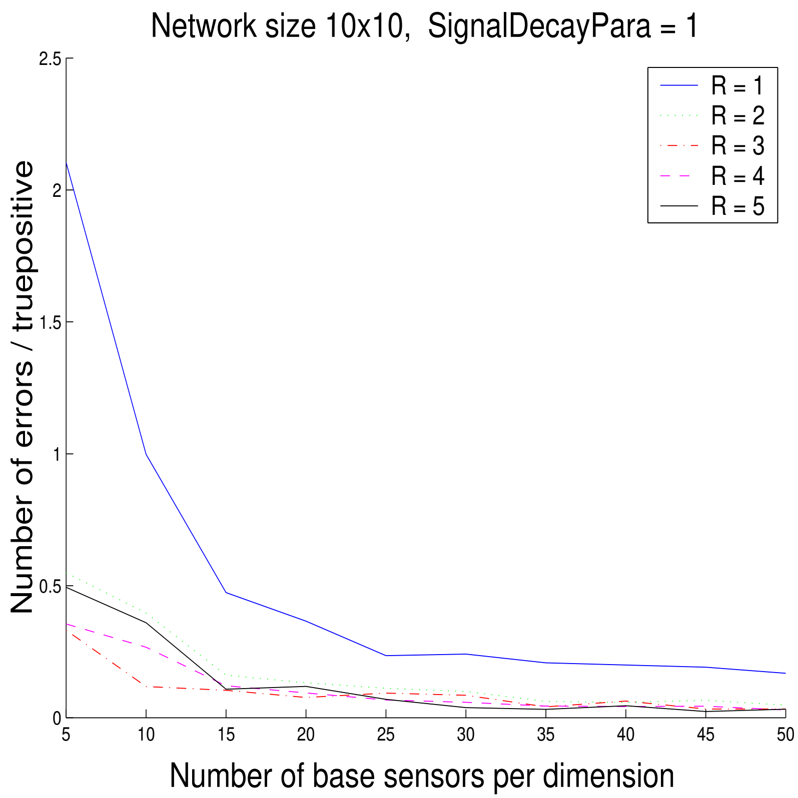
Simulation Results: Set-up

- Consider network of size 10×10 square units.
- n base sensors distributed uniformly in a grid-like structure.
- Take as input of our algorithm the signal matrix $l(x_i, x_j)$ being either a Gaussian or polynomial kernel.
- Consider kernel $k(x_i, x_j) = l(x_i, x_j)^{\sigma_0}$, with varying values of σ_0. This reduces to considering different parameters σ of either Gaussian or polynomial kernel.

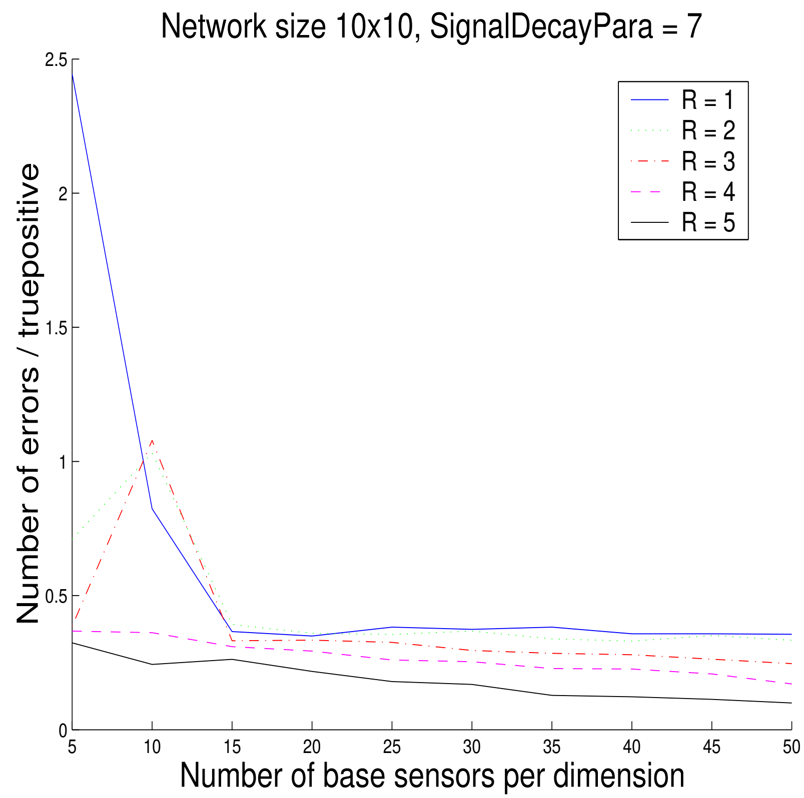
Coarse localization (cont)



Coarse localization (cont)

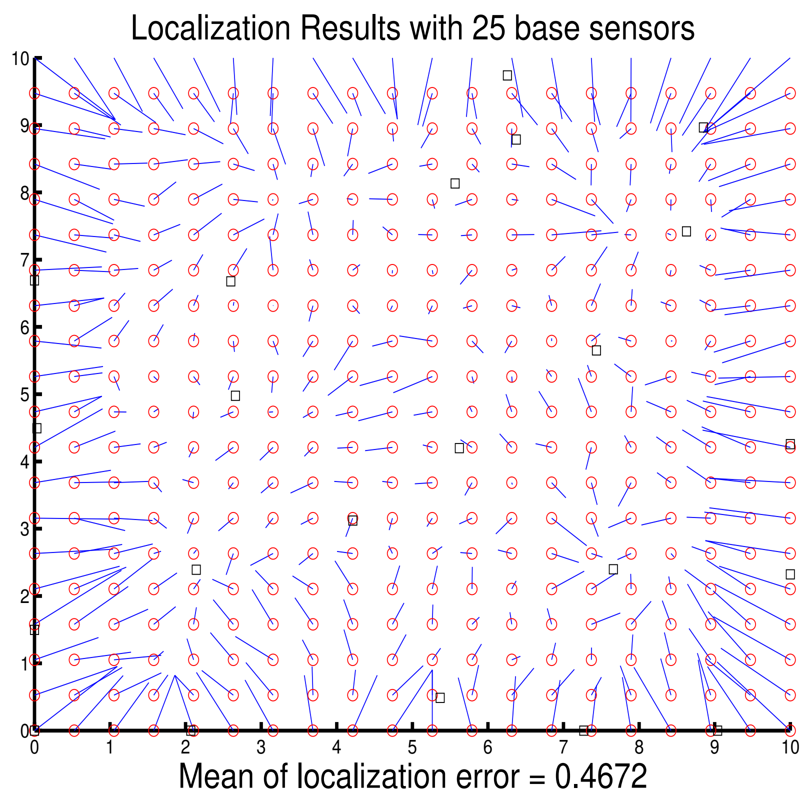


(a)

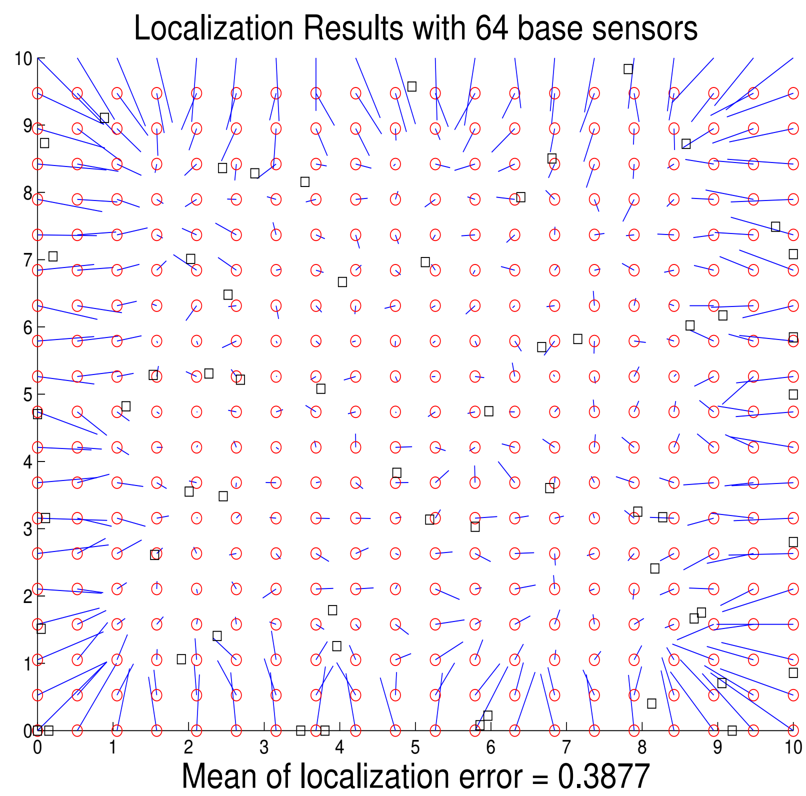


(b)

Fine-grained localization



(c)



(d)

Fine-grained localization (cont)

n	$R = 2.5, \sigma = 5, k = 15^2$	$R = 5, \sigma = 1, k = 10^2$	$R = 5, \sigma = 5, k = 10^2$
3	3.8483 ± 0.0187	3.8418 ± 0.0368	3.5474 ± 0.0954
4	3.5787 ± 0.0599	2.9052 ± 0.3528	2.2925 ± 0.3060
5	3.0292 ± 0.1311	0.9003 ± 0.2876	0.6813 ± 0.2612
6	2.0536 ± 0.3541	0.4789 ± 0.0165	0.4765 ± 0.0767
7	0.9112 ± 0.3018	0.4380 ± 0.0162	0.4309 ± 0.0179
8	0.4228 ± 0.1642	0.4153 ± 0.0127	0.4045 ± 0.0163
9	0.3107 ± 0.0635	0.4057 ± 0.0107	0.3868 ± 0.0106
10	0.2826 ± 0.0065	0.3992 ± 0.0093	0.3748 ± 0.0083

Table 1: Decrease of localization error (mean and standard deviation) as number of base sensors $N = n^2$ increases.

Conclusions

- Sensor network as a distributed pattern recognition device.
- Use of kernel-based learning algorithms in localization.
- Ongoing experiments with Berkeley motes.