On-line kernel learning for active sensor networks

Stefan Jorgensen*

I. Introduction

In this algorithmic project we consider how to leverage increasingly capable mobile sensor networks to simultaneously exploit a changing environment and maintain low uncertainty about the previously explored regions. This problem, known as 'coverage-estimation' or 'active sensing' in the literature, has applications in many fields: tracking interesting oceanographic or atmospheric data using a robotic network, finding and clearing environmental hazards such as pollutants [1] or radioactive waste [2], searching for humans in wreckage [3], and even game theoretical formulations such as the multi-armed bandit problem [4]. Much of the literature focuses on either the estimation problem of exploring an unknown environment or the coverage problem of exploiting a known environment.

A. Contribution

The main contribution of this project is to apply on-line adaptive kernel estimation techniques to the coverage estimation problem in an environment which reacts to observation. To the author's knowledge, this approach has not been tried in the published literature.

B. Related work

In [5] a kernel based Gaussian Regression approach to coverage estimation is proposed by Carron et al., where a centralized base station computes trajectories for each agent based on the measurement history up to the current time. The data dimension grows linearly with time i and hence the complexity grows as $O(i^3)$ due to a matrix inversion of the problem data. Nodes are directed toward areas with higher variance in order to improve the estimate. Once the maximum estimate variance is below a predefined threshold, the algorithm switches to a partitioning algorithm, which assumes the environment is static and moves agents to a centroidal Voronoi partition (a well-known optimal coverage configuration for distance-based problems, [6]). The algorithm in [5] is not well suited for on-line learning tasks since it does not provide the network with the ability to switch back from this exploitation state to the former exploration state, and the complexity of the exploration algorithm grows as $O(i^3)$.

A distributed, parametric and non-parametric approach to the same problem is proposed in [7]. The statistical model used is more general than [5] because they consider distributions which are not Gaussian Random fields. Starting from an optimal centralized policy, the authors develop a distributed approximation by assuming that the agent locations are all independently identically distributed and using a consensus filter to explicitly evaluate the kernel based feature vector for neighboring measurements (note that this precludes an implicit infinite dimensional feature vector). Only the one-shot estimation case is considered, so the active sensing question of how to move sensors is not addressed.

A method for choosing the most significant features is proposed, and it is claimed that this computation can be run online. However, given the weak assumptions made on the ad-hoc network, it is non-trivial to guarantee that at any given time, the agents have agreed on the basis functions (and ordering) to use in the algorithm. Without this guarantee, the inner-product evaluations that the algorithm relies on are not meaningful. If the 'important' features are chosen ahead of time, the kernel method reduces to a parametric method similar to those described in [8].

C. Motivation of proposed work

The approach to active sensing of a scalar field taken in this project is inspired by the non-parametric methods used in [5]. Parametric methods suffer from the same fundamental problem of linear estimators: due to the finite number of basis functions used to parametrize the space, parametric methods require a large number of functions to estimate general functions well (and there will always be a pathological case which is poorly estimated). In problems where prior knowledge is not available or where the statistics may change significantly, nonparametric kernel based methods provide an adaptive, general learning framework. They still require knowledge of the expected 'smoothness' via the kernel bandwidth.

D. Statement of work

This work examines the coverage estimation problem with kernel adaptive filters, which kernelize familiar linear estimators. In particular, kernel least-mean squares (KLMS) and kernel recursive least squares (KRLS) are used. Derivations and variants of the KLMS and KRLS algorithm are summarized in [9]. The goal of this project is to provide an on-line, kernel based approach to the problem of maximizing the minimum value of time-varying, reactive fields.

*Department of Electrical Engineering, Stanford University. Project advised by Dr. Marco Pavone. This work was funded in part by NSF grant DGE-114747. stefantj@stanford.edu

II. Problem formulation

A. Model

Assume we have N mobile sensors, and a central base station. We denote measurement times by the index i. Agent n has position x_n^i in a polygon $\mathcal{X}$, and takes measurements over a footprint $\mathcal{B}(x_n^i) \subset \mathcal{X}$. The measurements follow the model

$$y_{i,n}(x) = f(x, T(x,i), i) + \nu_{i,n}$$

for $x \in \mathcal{B}(x_n^i)$, where i is the time of measurement, $\nu_{i,n}$ is sensor noise and $T(x,i)$ is given by

$$T(x,i) = \frac{1}{i} \sum_{n=1}^{N} \sum_{j=0}^{i} \mathbf{1}\{x \in \mathcal{B}(x_n^j)\}$$

which is the fraction of total time that the point x has been observed up to time i. In this context, $\mathbf{1}(\cdot)$ is the indicator function. If multiple agents observe the same location, this fraction can be greater than 1 (though not greater than N). Agents move according to noise-less integrator dynamics,

$$x_n^i = x_n^{i-1} + u_n^i$$

B. System requirements

We have the following communication and computation requirements:

(C1) agent n can identify itself to the base station and can send information to the base station;
(C2) agent n can measure: $y(x) = f(x, T(x,i), i) + v_i$, over a region $\mathcal{B}(x_n^i)$.
(C3) agent n can measure its position x_n^i

The base station must be able to:

(C4) store measurements taken at time a given time step by all agents, in addition to the current estimates of f
(C5) compute partitions on $\mathcal{X}$
(C6) send information to each robot

The goal of the network is to develop a observation policy $T(x,i)$ to maximize the minimum value of $f(x, T(x,i), i)$.

C. Example problems

1) Scalar field estimation: This applies immediately, where we take f to be the time-varying error in measurement.

2) Sweeping and Persistent robotic tasks: This problem is proposed in [10], where they describe the field by the differential equation

$$\dot{f}(x) = \begin{cases} p(x) & \text{if } x \notin \cup_{n=1}^{N} \mathcal{B}(x_n) \\ p(x) - c(x) & \text{if } x \in \cup_{n=1}^{N} \mathcal{B}(x_n) \text{ and } f(x) > 0 \\ 0 & \text{if } x \in \cup_{n=1}^{N} \mathcal{B}(x_n) \text{ and } f(x) = 0 \end{cases}$$

Integrating, we find that (assuming that $p(x)$ and $c(x)$ are independent of time)

$$f(x,i) = ip(x) + f(x,0) - T(x,t)c(x) = f(x, T(x,i), i).$$

Their policy seeks to minimize the maximum of the field values, which fits out model by a simple change of variables.

3) Border patrol and intruder detection: Assume that there are intruders trying to penetrate a region $\mathcal{X}$ undetected. Their risk of failure is dictated by environmental factors and being caught by a patrol. Intruders have access to the patrol frequency $T(x,i)$ along the region, but not to real time position information of patrols. At every time step i, the intruder examines the environment and frequency of patrols, and compares it to a risk threshold τ_i. We also assume that intruders are impatient, so their risk threshold increases over time: $\tau_i > \tau_{i-1}$.

Patrolling agents move to a location and wait until they see an intruder or grow impatient, then move on in order to maximize the minimum intrusion time over $\mathcal{X}$. In a game theoretic context maximizing the minimum is the Nash Equilibrium, since to choose any other policy will reduce the patrolling agents value function (by definition), and the intruders cannot unilaterally increase their value function (to get in quickly).

We can formulate this problem as maximizing the minimum of a field representing the risk of failure (for intruders) over the patrol frequencies $T(x,i)$. Explicitly constructing this problem analytically typically requires strong statistical assumptions (such as are used in multi-armed bandit problems), but even without those we can say that this is a problem of maximizing the minimum of some function of space, time and average coverage time. Thus, this fits the model from Section II-A.

III. Proposed Solution

Our proposed solution is outlined in algorithm 1, which is very similar to [5], with two major differences: an adaptive kernel estimator is used in place of Gaussian Regression, and the field is assumed to respond to observation. Coverage is achieved by running a range-limited version of Lloyd's clustering algorithm [11]: each agent finds the centroid of a reward function based on f over all points nearest to itself and travels there. In a static environment, this algorithm will eventually converge to the centroidal Voronoi partition, which is the solution to many locational optimization problems. In our case, the field is time varying, so the locations of agents will not converge. Instead, we look for convergence of the smallest field values.

The methods $estimator.update(u, d)$ and $d = estimator.predict\_d(u)$, and $r = estimator.predict\_r(u)$ are described in [9] for KLMS and KRLS, but are not repeated here for brevity. The argument u is the measurement vector, d is the scalar output of the model (or observation) and r is the variance of the estimate d.

The method $moveReward(map, x_j^i)$ only considers the points which are closest to agent j (e.g., in its Voronoi partition). The method $reward(d, r)$ is described next.

Algorithm 1: Coverage Estimation

```
for i > 1 do
    for x ∈ 𝒳 do
        d = estimator.predict_d([x,T(x,i),i])
        r = estimator.predict_r([x,T(x,i),i])
        map(x) = reward(d, r)
    end for
    p = partition(map)
    c = centroids(p)
    for j ← 1, N do
        if (event or timeout) then
            estimator.update([x_j^i, T(x_j^i, i), i], y(x_j^i))
            x_j^{i+1} ← moveReward(map, x_j^i)
        else
            x_j^{i+1} ← x_j^i
        end if
    end for
end for
```

A. Reward Function Selection

The behavior of the algorithm will depend greatly on the choice of reward function $reward(d, r)$, which drives agent motion. Three functions were considered in this project:

1) $reward(d, r) = exp(-d)$*:* This method directly rewards traveling to areas with short intrusion times (low risk). The big downside is that there is no benefit given to exploration, which results in poor performance of regions which contain many local minima.

2) $reward(d, r) = (b-d)^2/(r^2+(b-d)^2)$*:* This function comes from the Cantelli probability inequality which states that for a positive, real random variable x with mean μ and variance σ^2, $\Pr(x - \mu \geq a) \leq \sigma^2/(\sigma^2 + a^2)$. Switching then probability and substituting $b = a + \mu$ yields:

$$\Pr(x \leq b) \geq \frac{(b-\mu)^2}{\sigma^2 + (b-\mu)^2}.$$

This choice of reward then corresponds to a conservative policy which minimizes risk. This runs into a similar problem as with the direct method, because when the variance r is large, the lower bound tends to 0. This is addressed in the next function.

3) $reward(d, r) = -d + \sqrt{2r\log(i)/s} + 3r\log(i)/s$*:* This is based on a function developed in [12] based on a multi-armed bandit formulation. The multi-armed bandit problem is the idealized coverage-estimation problem: everything is independent, identically distributed, Markov, and time-invariant. A regret bound is also given in [12]. Here $s = iT(x, i)$ is the number of times x has been sampled and b is the support of the reward function (in this formulation, the impatience of the patrolling agents). This approach tries to balance exploration and exploitation in a meaningful way.

The first two methods do not explore well. Unless otherwise specified, $reward(d, r)$ is taken as the regret-minimization function from the multi-armed bandit problem.

IV. Simulations

Algorithm 1 was implemented in C in order to allow reasonable simulation time for modest size problems. Numerous problem settings were explored, with various environments and reward functions. In the following, only the most recent results are presented as they most clearly express the properties of our algorithm.

A. Problem set-up

The equation used by intruders to assess difficulty at point x is

$$D(x) = 10(prior(x) + T(x, i)).$$

The prior function used for this problem is of the form $\sin(x)\sin(y)$, and is shown in Fig. 1. Note the severe non-convexity of the prior, with two distinct local minima regions. The intruder impatience starts at 0.1 and grows by 1 with each time step, and an event occurs when the impatience at x exceeds $D(x)$ (impatience resets on event). A Gaussian kernel was chosen with bandwidth parameter 10. Agents have impatience of 10 (if nothing is seen for 10 iterations, they move on).

Simulations are run over a 50x50 grid with 3 agents that can see manhattan distance 3. This means that for every time step, 75 of 2500 grid points are observed (3% of the region).

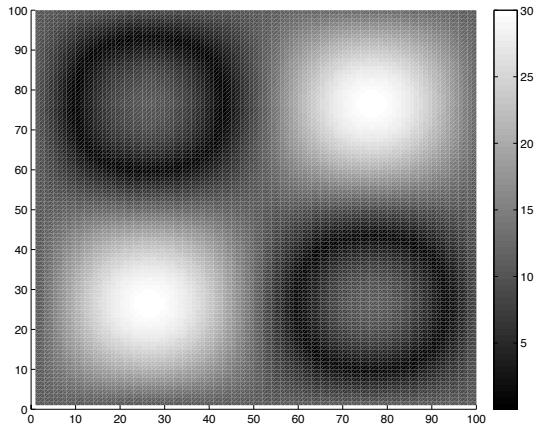

Fig. 1. Prior field difficulty

B. Desired results

Intuitively, we should see agents focusing their efforts in the upper-left and lower-right corners, to compensate for the relatively easy environment there. Fig. 2 shows the difference between the initial and final difficulty function for the KRLS implementation. Qualitatively, the patrols are taking the right action and we quantify how good their responses were below.

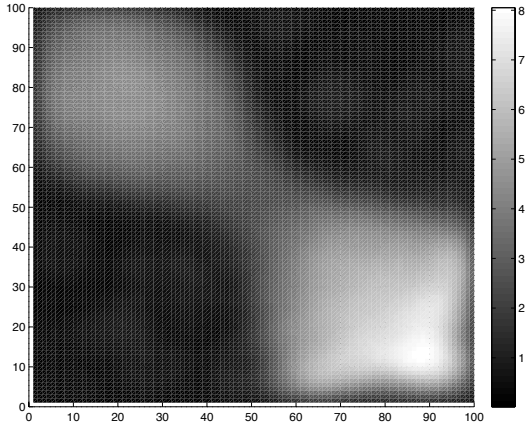

Fig. 2. Change in perceived difficulty over 1230 iterations, single KRLS trial

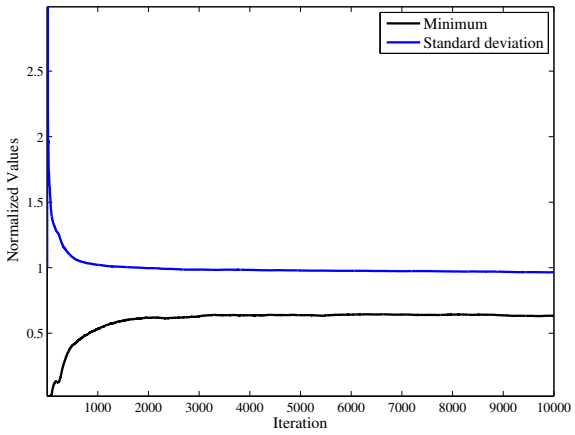

Fig. 3. KLMS results, averaged over 180 trials

An upper bound for the maximum minimum field value was computed using a centralized, omniscient controller with no constraints on how to apply patrol effort. In particular, frequency allocation at a given time step can be scattered and is not restricted to points within a given manhattan distance of the patrolling agents.

Ideally, the minimum field value will approach this upper bound as time goes to infinity. From the problem structure, we also expect the variance of the field to decrease over time. If there are sufficient agents, then the difficulty field could theoretically be flattened. In our case, we only consider a sparse patrol scheme so we simply look for decreasing variance over time. We expect a transient at the beginning while agents are exploring the region.

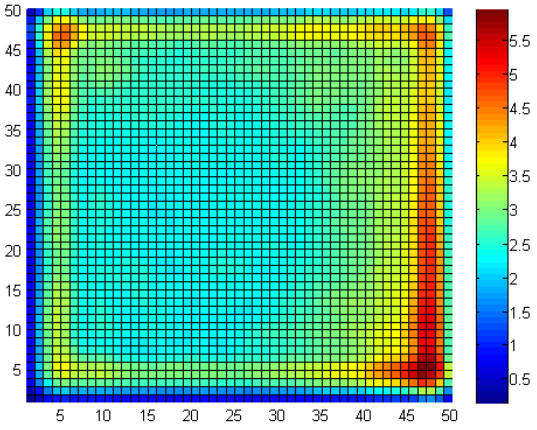

Fig. 4. Change in perceived difficulty over 10,000 iterations, single KLMS trial

C. KLMS results

The minimum and variance of the field for the KLMS algorithm are shown in Fig. 3. The curves were generated by averaging the results of 180 runs of 10,000 iterations. Clearly, the algorithm is not able to generate a coherent policy, since the field variance settles near its initial value. This implies that the algorithm is simply generating random walk trajectories, which is verified by looking at the change in perceived difficulty for a single trial (shown in Fig. 4). From this we can conclude that the random walk trajectory is able to accomplish approximately 60% of the optimal maximum minimum field value.

Our best explanation for the poor performance of KLMS is that the algorithm is not able to learn 'quickly enough' for the given problem dynamics, since in some more pedantic environments KLMS performed more reasonably. This lack of generalizability is a serious problem, and even though KLMS has linear time complexity (versus $O(i^2)$ for KRLS), it is not suitable for our application.

D. KRLS results

The minimum and variance of the field for the KRLS algorithm are shown in Fig. 5. The curves were generated by averaging the results of 30 runs of 3,000 iterations. The reduced number of iterations is because of the significant increase in complexity for KRLS. After approximately 250 iterations, the agents (implicitly) decide on the locations of interest and begin to focus their efforts there. After 3000 iterations, the variance reduces by 36% from its initial value, and the maximum minimum achieves 83.5% of the upper bound. Given the very minimal modeling and tuning of this problem this is impressive.

Sparsification techniques such as the ones outlined in [9] were implemented, but turned out to be very sensitive. A 10% difference in threshold value switched behavior from overly aggressive (never accepting a new point) to useless (accepting all points). Due to this sensitivity, it is not practical to use for a generic problem scenario.

Table 1 summarizes the simulation results.

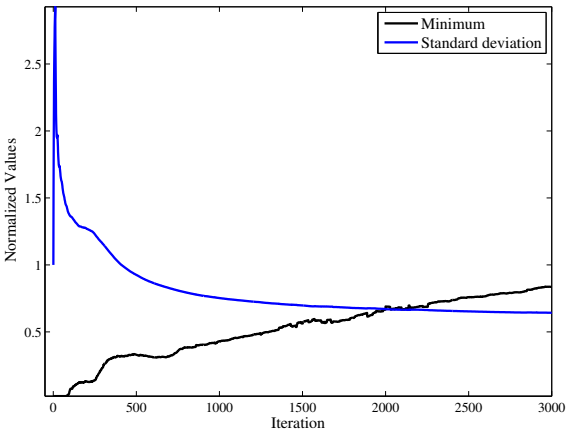

Fig. 5. KRLS results, averaged over 30 trials

Metric (normalized)	KLMS	KRLS	Ideal
Minimum	0.6331	0.8355	1.0
Variance	0.9651	0.6423	0.0

TABLE I
SUMMARY OF RESULTS

V. Conclusions

This project applied on-line adaptive kernel estimation techniques to the coverage estimation problem in an environment which reacts to observation. This approach has not been tried in the literature, and for KRLS there is potential for success. The KLMS approach suffered from sensitivity to problem dynamics, the specifics of which should be investigated in future work. The KRLS approach was able to get within 83.5% of an upper bound of the solution, which is good considering that the bound is not particularly tight.

The results are particularly impressive given then non-convexity of the problem and existence of local minima. The three agents learn to divide their time among the two local minima, crossing a less desirable region occasionally to balance effort. The main drawback from using KRLS is its need for many samples to learn, and quadratic complexity in the number of samples used. Sparsification techniques were tried, but proved too difficult to tune for general applications.

Future work should focus on developing theoretical guarantees for applying this algorithm (or a variant) to the interactive coverage-estimation problem. Properties such as convergence time (as related to environment dynamics), expected regret, minimum number of agents to achieve a given maximum minimum field value, and probabilistic service guarantees are derived for similar problems and would provide much insight to this problem. There are quite a few applications in the literature and industry which fit the model that was formulated in Section II-A, and there is the potential for combining the results from several fields if a theoretical backing can be developed here.

References

[1] J. Oyekan, Huosheng Hu, and Dongbing Gu. A novel bio-inspired distributed coverage controller for pollution monitoring. In *Mechatronics and Automation (ICMA), 2011 International Conference on*, pages 1651–1656, Aug 2011.

[2] R. Andres Cortez, Herbert G. Tanner, and Ron Lumia. Distributed robotic radiation mapping. In *Tracts in Advanced Robotics*, pages 147–156. Springer, 2009.

[3] Vijay Kumar, D. Rus, and Sanjiv Singh. Robot and sensor networks for first responders. *Pervasive Computing, IEEE*, 3(4):24–33, Oct 2004.

[4] Jerome Le Ny, Munther Dahleh, and Eric Feron. Multi-uav dynamic routing with partial observations using restless bandit allocation indices. In *American Control Conference, 2008*, pages 4220–4225. IEEE, 2008.

[5] Andrea Carron, Marco Todescato, Ruggero Carli, Luca Schenato, and Gianluigi Pillonetto. Multi-agents adaptive estimation and coverage control using gaussian regression. *CoRR*, abs/1407.5807, 2014.

[6] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: applications and algorithms. *SIAM review*, 41(4):637–676, 1999.

[7] Damiano Varagnolo, Gianluigi Pillonetto, and Luca Schenato. Distributed parametric and nonparametric regression with on-line performance bounds computation. *Automatica*, 48(10):2468–2481, 2012.

[8] Mac Schwager, Daniela Rus, and Jean-Jacques Slotine. Decentralized, adaptive coverage control for networked robots. *The International Journal of Robotics Research*, 28(3):357–375, 2009.

[9] Weifeng Liu, Jose C Principe, and Simon Haykin. *Kernel Adaptive Filtering: A Comprehensive Introduction*, volume 57. John Wiley & Sons, 2011.

[10] Stephen L Smith, Mac Schwager, and Daniela Rus. Persistent robotic tasks: Monitoring and sweeping in changing environments. *Robotics, IEEE Transactions on*, 28(2):410–426, 2012.

[11] Francesco Bullo, Jorge Cortés, and Sonia Martinez. *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009.

[12] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.