ACTIVE DECOMPOSITION AND SENSING IN NETWORKS OF DISTRIBUTED CHEMICAL SENSORS

Rakesh Gosangi and Ricardo Gutierrez-Osuna
Department of Computer Science and Engineering, Texas A&M University
{rakesh,rgutier}@cse.tamu.edu

ABSTRACT

Active sensing enables a sensor to optimize its tunings on-the-fly based on information obtained from previous measurements. When applied to networks of distributed sensors, however, active sensing becomes computationally impractical due to the combinatorial number of sensing configurations. To address this problem, we present an active decomposition and sensing (ADS) method that combines the advantages of classifier decomposition with those of active sensing. Namely, we use class posteriors to decompose the problem across the sensors in the network. Each sensor then applies active sensing to select the next tuning to solve its specific subproblem. As a result, the method scales linearly (rather than combinatorially) with the number of sensors. We validate ADS on a database of infrared absorption spectroscopy containing 50 chemicals. Our results show that active decomposition improves classification performance and reduces sensing costs when compared to using active sensing only at the node level.

Index terms– Active sensing, chemical sensing, sensor networks.

1. INTRODUCTION

The response of chemical sensors is generally measured as a change in a particular physico-chemical property: absorption at a specific wavelength, or conductivity at a particular operating temperature. In many cases, however, additional information can be extracted by modulating some internal property of the sensor. As an example, measuring the conductivity of a metal-oxide (MOX) sensor over a range of temperatures provides a wealth of information. This additional information comes at a cost (e.g., sensing times, power consumption), so feature-subset selection (FSS) techniques are commonly used to identify a subset of the most informative sensor tunings.

Over the past decade, a number of investigators have explored active sensing techniques as an alternative approach to sensor optimization. In contrast with FSS, where the sensor tunings are hardcoded, active sensing can adapt the sensor tunings in real-time based on information obtained from previous measurements. Previous work has shown that active sensing can achieve higher classification performance than FSS while using a fraction of the sensing configurations and lower power consumption, and also provides a trade-off between sensing costs and classification performance [1, 2]. Unfortunately, these active sensing techniques do not scale up to networks of distributed sensors because of the combinatorial explosion in the number of sensing configurations.

In this article, we present an active decomposition and sensing (ADS) algorithm suitable for networks of tunable sensors. ADS borrows concepts from classifier decomposition [3] and our prior work on active sensing with single tunable sensors. In a first step, ADS decomposes the classification problem into independent subproblems using a seeding strategy based on the posterior probability of each class, and then distributes the subproblems among the sensors in the network. In a second step, ADS applies active sensing to select the operating configuration for each sensor to solve its assigned subproblem. In a final step, ADS combines information from each sensor to update the posterior of each class, and the cycle repeats until convergence. In this way, the computational complexity of ADS scales linearly with the number of sensors in the network (rather than combinatorially). We tested the method on a database of infrared spectra with varying network sizes and number of sensing steps.

The rest of the paper is organized as follows. Section 2 provides a short note on related work. In section 3, we describe our proposed methods. In section 4, we present the experimental work and results. We conclude the paper with a discussion and directions for future work in section 5.

2. RELATED WORK

Active sensing has had a long history in robotics [4, 5] and computer vision [6, 7], but its application in chemosensing has been rather limited. In one of the earliest studies, Nakamoto et al. [8] developed a method for active odor blending, where the goal was to reproduce an odor blend by creating a mixture from its individual components. The authors developed a control algorithm that adjusted the mixture ratio so that the response of a gas sensor array to the mixture matched the response to the odor blend. Our initial work on active sensing [1] focused on classification of volatile chemicals using a temperature-modulated metal oxide sensor. In this work, we modeled the sensor dynamics with an input-output hidden Markov model, and used a partially observable Markov decision process (POMDP) to select sensor tunings in real-time. In later work, we extended this approach to concentration-independent discrimination of chemical samples with tunable IR interferometers [9]. Also recently, Dinakarababu et al. [10] developed an adaptive spectrometer for rapid classification of chemicals. Unlike a traditional spectrometer, this device contains a tunable spectral filter (a digital micro-mirror device) that can select a given set of certain spectral bands and direct them onto the detector. Based on previous measurements, the bands are chosen to focus on spectral portions that are most useful for classification.

Our work is also related to previous research on sensor scheduling in large sensor networks. Tharmarasa et al [11] developed an algorithm to select subsets of sensors from a large wireless networks to track multiple targets. The appproach was based on convex optimization with sequential quadratic programming. Additional work includes load allocation algorithms to schedule sensors for signal measurement and reporting tasks [12], and node scheduling algorithms for maximizing sensor

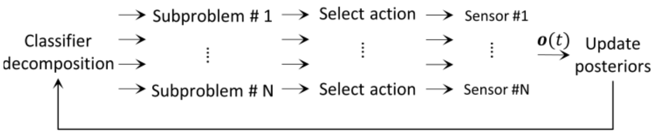

Figure 1 – An overview of the proposed method.

coverage [13]. Due to the distributed nature of sensor networks, the focus of these papers is to minimize data transmission and communication costs.

Contribution and relation to prior work: This paper presents an active sensing approach for classification with a network of distributed chemical sensors. The approach builds on our previous work on active sensing for classification with individual sensors [1, 9], but extends it to scenarios where measurements can be acquired *simultaneously* by multiple tunable sensors. This requires a reformulation of the problem to avoid combinatorial explosion. Our work also relates to previous research on sensor selection and scheduling in wireless networks [11, 12]. In those papers, however, the sensors are assumed to be passive (i.e., each sensor works at a fixed configuration and provides a single measurement), whereas our approach considers the case when sensors are tunable.

3. METHODS

Consider a network of N tunable chemical sensors $\boldsymbol{S} = \langle s_1, s_2, \dots, s_N \rangle$, where each sensor s_i can be operated at d distinct configurations $\mathbf{P} = \langle \rho_1, \rho_2, \dots, \rho_d \rangle$ and all sensors in the network are nominally identical. The sensor network is exposed to a volatile chemical, and the goal is to determine the identity of the chemical from a list of C possible targets $\boldsymbol{\Omega} = \langle \omega_1, \omega_2, \dots, \omega_C \rangle$. To solve this problem, the sensors are operated by a centralized controller, which can adjust the configurations of each sensor and collect their responses. Namely, the goal of the controller is to find an optimal sequence of T action vectors $\{\boldsymbol{a}(1), \boldsymbol{a}(2), \dots, \boldsymbol{a}(T)\}$, where each action vector $\boldsymbol{a}(t)$ corresponds to simultaneously operating N sensors (or a subset of them), each sensor at a particular configuration, and measuring their responses. A naïve active sensing approach would search through all possible sensing configurations (d^N) at each time step, evaluate their expected information content, and select the best configuration. Clearly, this approach will only work for small networks with few sensor tunings.

To avoid this combinatorial explosion, our approach consists of dividing the problem and distributing it among the N sensors. Our algorithm works in three steps, as illustrated in Figure 1. In a first step, we decompose the classification problem into N mutually exclusive subproblems. In a second step, we apply active sensing on each sensor to select its next tuning. In a final step, the controller combines the response from all sensors, and updates the posteriors associated with each class. A detailed description of these three steps follows. The overall algorithm is summarized in Table I.

3.1. Classifier decomposition

Given the distribution $\boldsymbol{p}(t) = \{p_1(t), p_2(t), \dots, p_C(t)\}$, where $p_i(t)$ is the posterior probability of class ω_i at time t given all previous measurements from all sensors $\boldsymbol{O} = \{\boldsymbol{o}(1), \boldsymbol{o}(2), \dots \boldsymbol{o}(t)\}$, we use a seeding strategy to divide the classification problem into N subproblems, one subproblem per sensor. Namely, we rank the classes according to their posteriors and then group them into N subproblems, where the j^{th} subproblem consists of discriminating among the j^{th}, $(N+j)^{th}$, $(2N+j)^{th}$ most likely classes. In this fashion, each subproblem has a maximum of $\lceil C/N \rceil$ classes. This decomposition method has two advantages. First, because the subproblems have similar levels of complexity, the classification problem is distributed relatively uniformly across the network. Second, it reduces the complexity of each subproblem because it combines classes that are significantly different from each other given the current observations[1].

In our experience, it normally takes a few time steps for the posterior to be dominated by a handful of classes. In these cases, it is more efficient to focus the search on the most promising classes. For this reason, at each iteration we prune out those classes whose posteriors fall below a predefined threshold ϵ. The resulting class space becomes $\boldsymbol{\Omega}(\mathbf{t}) = \{\omega_k \mid p_k(t) \geq \epsilon\}$, with cardinality $|\boldsymbol{\Omega}(t)|$. To ensure each subproblem has a minimum of two classes, once $|\boldsymbol{\Omega}(t)| < 2N$, we create only $\lfloor |\boldsymbol{\Omega}(t)|/N \rfloor$ subproblems and distribute them to a subset of the N sensors; this allows us to reduce sensing costs further.

3.2. Active sensing

Once the subproblems are distributed across the network, we apply active sensing at each sensor to select their next configurations. Assuming sensor s_i has been assigned a subproblem consisting of classes $\omega_1, \omega_2, \dots, \omega_r$ with posteriors $p^{(s_i}(t) = \{p_1(t), p_2(t), \dots, p_r(t)\}$, we select the next configuration for s_i such that uncertainty in $p^{(s_i}(t)$ is minimized. To do this, we first estimate the expected reduction in entropy in $p^{(s_i}(t)$ if sensor s_i was operated at configuration ρ_j:

$$J(s_i, \rho_j) = H\big(p^{(s_i}(t)\big) - H\big(p^{(s_i}(t+1) \big| s_i, \rho_j\big) \qquad (1)$$

Then, we select the configuration with the maximum expected reduction $\boldsymbol{a}^{(s_i}(t) = \mathrm{argmax}_{\rho_j} J(s_i, \rho_j)$. We repeat this process for each sensor to create the action vector $\boldsymbol{a}(t)$. While doing so, we track the configurations that were previously used to avoid repetition.

The term $H(p^{(s_i}(t))$ in eq. (1) is the entropy of $p^{(s_i}(t)$, and $H(p^{(s_i}(t+1)|s_i, \rho_j)$ is the expected entropy when s_i is operated at ρ_j. We compute expected entropy as the average over all possible observations, weighted by the probability of each observation[2]:

$$H(p^{(s_i}(t+1)|s_i, \rho_j) = \sum_{\forall o} p\big(o \big| s_i, \rho_j\big) H\big(p^{(s_i}(t+1)|o, s_i, \rho_j\big) \qquad (2)$$

where the term $p(o|s_i, \rho_j)$ is the probability of obtaining response o from s_i at ρ_j, irrespective of the class label.

3.3. Sensor fusion

Once the sensors have been driven with action vector $\boldsymbol{a}(t)$, the controller combines the responses obtained from all the sensors

---

[1] As an example, given a problem with 20 chemicals and 5 sensors, the first sensor is assigned a problem that contains the most likely class, the 6th most likely class, the 11th most likely class, and so on.

[2] Eq. (2) is only applicable for discrete observational spaces. For continuous spaces, it becomes an integration problem. Therefore, during the training stage, we uniformly discretize the sensors' observational spaces.

Table I – Pseudo-code of the ADS algorithm

```
Step 1: Initialization
    - Initialize posteriors: $p_k(0) = \frac{1}{C}$
    - Initialize the list of available configurations: $F(t) = \mathbf{P}$
Step 2: Classifier decomposition
    - Class pruning $\boldsymbol{\Omega}(t) = \{\omega_k \mid p_k(t) \geq \epsilon\}$
        o if $|\boldsymbol{\Omega}(t)| > 1$, distribute $\boldsymbol{\Omega}(t)$ among the sensors
        o else, declare class label $\omega_{out}$
Step 3: Active sensing
    - For each sensor $s_i$
        o Select next action to minimize entropy: $\boldsymbol{a}^{(s_i}(t) = \text{argmax}_{\rho_j \in F(t)} J(s_i, \rho_j)$
        o Remove action from the list: $F = F - \boldsymbol{a}^{(s_i}(t)$
        o Apply action vector $\boldsymbol{a}_t$ to obtain observation vector $\boldsymbol{o}_t$
Step 4: Update class posteriors
    - For each classs $\omega_k$
        o $p_k(t+1) \propto p_k(t) p(\boldsymbol{o}(t) | \omega_k, \boldsymbol{a}(t))$
Step 5: Stopping criterion
    - if $t < T$
        o $t = t + 1$
        o Go to step 2
    - else declare class label $\omega_{out}$
```

$\boldsymbol{o}(t)$ and updates the posteriors associated with each class using the sequential Bayesian update equation:

$$p_k(t+1) = \frac{p_k(t) p(\boldsymbol{o}(t) | \omega_k, \boldsymbol{a}(t))}{p(\boldsymbol{o}(t))} \tag{3}$$

where $p(\boldsymbol{o}(t) | \omega_k, \boldsymbol{a}(t))$ is the probability of obtaining observation vector $\boldsymbol{o}(t)$ when the network is exposed to chemical ω_k and driven with action vector $\boldsymbol{a}(t)$. This value is obtained by combining probabilities of responses obtained at each sensor[3]. The denominator in eq. (3) is a normalization constant that ensures the posteriors sum to one.

At $t = 0$, the controller initializes the posteriors uniformly $p_k(0) = 1/C$. The controller brings the sensing process to a halt in two cases: (1) when T sensing steps are completed, or (2) $|\boldsymbol{\Omega}(t)| = 1$. In either case, the final class label is declared based on maximum a posteriori (MAP) criterion as:

$$\omega_{out} = \underset{1 \leq k \leq C}{\text{argmax}}\, p_k(t) \tag{4}$$

4. EXPERIMENTAL RESULTS

We validated the ADS algorithm on a simulated network of tunable Fabry-Perot interferometers (FPI) [15, 16]. An FPI consists of two partially reflective parallel mirrors forming an optical resonance cavity. An FPI demonstrates constructive interference[4] when the distance between the mirrors is an integer multiple of half the wavelength of the incident waves. Thus, by changing the distance between the mirrors, an FPI can be tuned to different wavelengths.

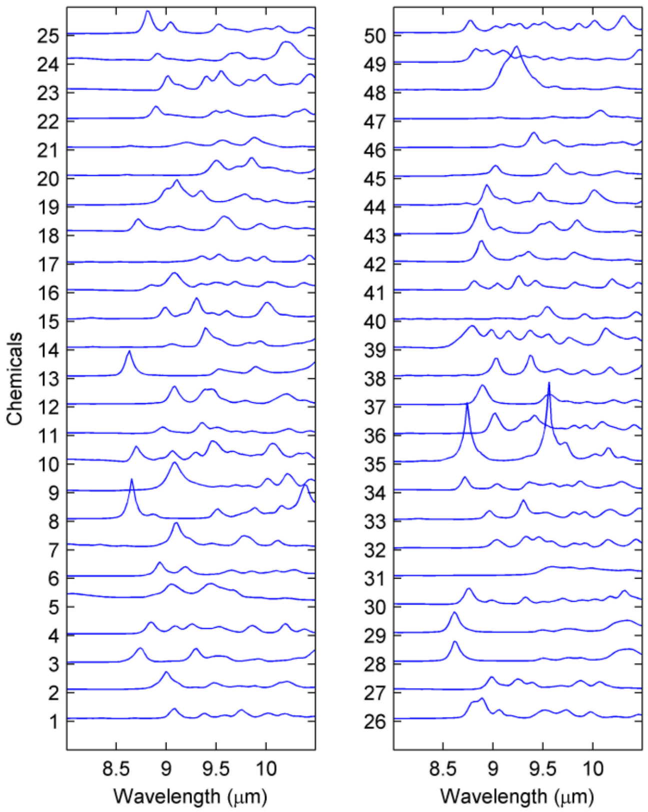

Figure 2 – Absorption spectra all the chemicals as a function of wavelength. For visualization purposes, the spectra are plotted with an offset along the y-axis.

The spectral range and tuning efficiency (resolution) of FPIs is limited by the mirror's reflection coefficients, spectral range of the infrared source, and physical limitations in micro-actuation. FPI is a low-cost portable alternative to Fourier Transform Infrared Spectroscopy (FTIR)[5], though with much lower spectral resolution and range.

4.1. Data

We simulated the response of an FPI to different chemicals using data from NIST chemistry WebBook [17]. This database provides high resolution (250 points per μm) FTIR spectra in the wavelength range of 3-21μm for over 16,000 chemicals. We chose 50 chemicals from this database that have strong absorption peaks in the range 8-10.5μm, and down-sampled the spectra to 116 spectral lines. This is similar to operating range and spectral resolution of commercially available FPI sensors (LFP-80105-337; Infratec, Inc). Resulting spectra are shown in Figure 2. Using these spectra, we generated 10 samples per chemical by adding Gaussian noise ($\sigma = 0.05$) at each wavelength, and used the resulting dataset (500 spectra) as a training set. We tested ADS with varying number of sensors and sensing steps.

We compared ADS against two baseline methods. The first baseline method (AS) can be thought of as a naïve active sensing approach. AS operates in a similar manner as ADS, but it does not decompose the classification problem. Instead, at each time step,

---

[3] During the training stage, we use Gaussian mixture models (GMMs) [14] to learn the probabilistic distribution of the sensor responses to different chemicals at different operating configurations.

[4] Constructive inference is an optical phenomenon where two waves superimpose resulting in a wave of higher amplitude.

[5] FTIR extensively used for chemical analysis, but is limited to laboratory settings due to its cost, size, and power requirements.

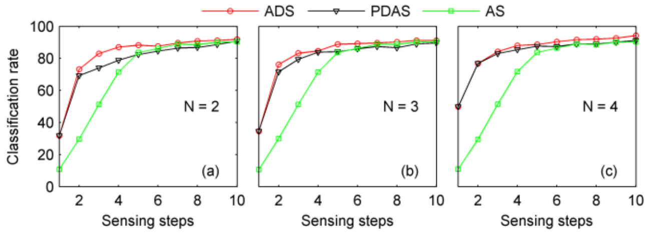

Figure 3 – Classification performance of ADS, PDAS, and AS, as a function of the number of sensing steps for (a) $N = 2$, (b) $N = 3$, and (c) $N = 4$.

all sensors in the network are assigned the same classification problem containing $|\mathbf{\Omega}(t)|$ classes. The second baseline method (PDAS; passive decomposition-active sensing) also operates in a similar fashion as ADS, except that it does not consider class posteriors while decomposing the classification problem. Instead, PDAS decomposes the problem according to the class labels (e.g., sensor s_1 is assigned classes $\omega_1, \omega_2, \ldots, \omega_{[C/N]}$, sensor s_2 is assigned classes $\omega_{[C/N]+1}, \omega_{[C/N]+2}, \ldots, \omega_{[2C/N]}$ etc.) Note that the three methods (ADS, AS, and PDAS) are equivalent for $N = 1$.

4.2. Performance vs. sensing steps

We tested the performance of ADS, AS, and PDAS with varying number of sensing steps $T = 1 - 10$ on three network sizes ($N = 2,3,$ and 4 sensors). We tested the three methods on 500 test samples (10 per chemical), where the samples contained an additive Gaussian noise ($\sigma = 0.1$). The classification performance for the three methods is summarized in Figure 3.

For $N = 2$ sensors, classification performance for all methods increases from 31.8% (ADS and PDAS), and 10.9% (AS) after one sensing step to 91.8% (ADS), 90.8% (PDAS), and 90.4% (AS) after 10 steps. The proposed algorithm (ADS) consistently outperforms both baseline methods (AS, PDAS) for all sensing steps. Differences between ADS and PDAS are more evident in the initial time steps; as additional sensing steps are taken, the cumulative feature sets selected by the two methods converges to the full set, thus reducing their differences. This is also the case for AS, which is surpassed by ADS and PDAS during the initial steps but later converges to the same performance level.

We observed similar behavior with $N = 3$ and $N = 4$ sensors, though the difference between ADS and PDAS becomes smaller. Classification performance for both methods during the first time step also increases (34.6% for $N = 3$; 49.8% for $N = 4$) when compared to $N = 2$ sensors (31.8%). The main reason for this improvement is that both methods acquire more features per time step because of the increase in the number of sensors. As expected, the performance of AS does not change significantly with the number of sensors in the network.

We also analyzed the average number of sensors used by each method as a function of the number of sensing steps. The results are summarized in Figure 4. For $N = 2$, all methods use the available two sensors at the first sensing step. However, with increasing number of sensing steps, the average number of sensors used reduces to 0.97 (ADS), 1.08 (PDAS), and 1.7 (AS) at $T = 10$. Among the three methods, ADS consistently uses the fewest number of sensors at all sensing steps, thus minimizing the sensing costs. We observed similar trends with $N = 3$ and $N = 4$ sensors. However, with increasing number of sensors, the difference between ADS/PDAS and AS becomes more prominent.

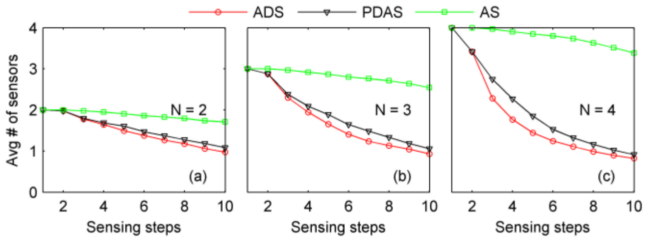

Figure 4 - The average number of sensors used by ADS, PDAS, and AS as a function of the number of sensing steps for (a) $N = 2$, (b) $N = 3$, and (c) $N = 4$.

5. DISCUSSION AND FUTURE WORK

This paper has presented an active decomposition and sensing (ADS) method for classification with a distributed network of tunable chemical sensors. Our results show that ADS obtains better classification performance while using fewer sensors than two baseline techniques: an active sensing approach without decomposition (AS), and an active sensing approach with passive decomposition (PDAS).

In the AS baseline method, all sensors in the network use the same configuration at each sensing step because they are assigned the same problem. This allows AS to improve signal-to-noise ratio by averaging out uncorrelated noise across the N sensors. Though ADS and PDAS do not have this advantage, the decomposition step allows them to acquire complementary information from different sensors, and therefore obtain better classification rates. The benefits of performing classifier decomposition diminish as additional sensing steps are taken, to where AS performs comparably to ADS and PDAS at the latter stages of sensing (Figure 3). This is to be expected, since the three methods must ultimately converge to the same feature set. Though Figure 4 shows the number of sensors used at each time, it also reflects the number of classes in contention at each sensing step ($|\mathbf{\Omega}(t)|$). With this in mind, the results in Figure 4 show that ADS can prune classes at a faster rate than PDAS. This can be attributed to the seeding strategy used by ADS, which combines the more likely classes with the less likely classes. As a result, less likely classes are eliminated at earlier sensing steps.

Several aspects of the model need improvement and thus warrant future research. First, ADS is myopic in nature, i.e. it looks only one step ahead while estimating the expected reduction in entropy. Non-myopic extensions would optimize over multiple sensing steps, though at a computational cost. Second, our method uses a seeding strategy to generate (C/N)-ary subproblems. Other decomposition schemes such as one-versus-all, pair-wise coupling [18], error correcting output codes (ECOC) [3] could also be used. Finally, our study assumed that the concentration of the target chemical remain constant across sensors. Additional work is needed to address situations where chemical concentrations change both with time and sensor location, as is likely to be the case in actual deployments of sensor networks.

Our study demonstrated the effectiveness of ADS on an off-line dataset of infrared absorption spectroscopy. Work in our group is underway to validate the method on experimental sensor data.

6. ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation under Award #1002028.

7. REFERENCES

[1] R. Gosangi, and R. Gutierrez-Osuna, “Active Temperature Programming for Metal-Oxide Chemoresistors,” *Sensors Journal, IEEE,* vol. 10, no. 6, pp. 1075-1082, 2010.

[2] R. Gosangi, and R. Gutierrez-Osuna, “Energy-aware active chemical sensing,” in Sensors, 2010 IEEE, 2010, pp. 1094-1099.

[3] T. G. Dietterich, and G. Bakiri, “Solving multiclass learning problems via error-correcting output codes,” *Journal of Artificial Intelligence Research,* vol. 2, no. 1, pp. 263-286, 1995.

[4] L. Mihaylova, T. Lefebvre, H. Bruyninckx *et al.*, “Active sensing for robotics-a survey,” in Fifth International Conference On Numerical Methods and Applications, 2002, pp. 316–324.

[5] A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien, “Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation,” in Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems' 96, IROS 96, , 1996, pp. 963-972.

[6] D. H. Ballard, “Animate vision,” *Artificial Intelligence,* vol. 48, no. 1, pp. 57-86, 1991.

[7] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, “Active vision,” *International Journal of Computer Vision,* vol. 1, no. 4, pp. 333-356, 1988.

[8] T. Nakamoto, S. Ustumi, N. Yamashita *et al.*, “Active gas/odor sensing system using automatically controlled gas blender and numerical optimization technique,” *Sensors and Actuators B: Chemical,* vol. 20, no. 2–3, pp. 131-137, 1994.

[9] J. Huang, R. Gosangi, and R. Gutierrez-Osuna, “Active Concentration-Independent Chemical Identification With a Tunable Infrared Sensor,” *Sensors Journal, IEEE,* vol. 12, no. 11, pp. 3135-3142, 2012.

[10] D. V. Dinakarababu, D. R. Golish, and M. E. Gehm, “Adaptive feature specific spectroscopy for rapid chemical identification,” *Optics Express,* vol. 19, no. 5, pp. 4595-4610, 2011.

[11] R. Tharmarasa, T. Kirubarajan, and M. L. Hernandez, “Large-Scale Optimal Sensor Array Management for Multitarget Tracking,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol. 37, no. 5, pp. 803-814, 2007.

[12] M. Moges, and T. G. Robertazzi, “Wireless sensor networks: scheduling for measurement and data reporting,” *IEEE Transactions on Aerospace and Electronic Systems,* vol. 42, no. 1, pp. 327-340, 2006.

[13] D. Tian, and N. D. Georganas, “A coverage-preserving node scheduling scheme for large wireless sensor networks,” in Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, Atlanta, Georgia, USA, 2002, pp. 32-41.

[14] D. A. Reynolds, and R. C. Rose, “Robust text-independent speaker identification using Gaussian mixture speaker models,” *IEEE Transactions on Speech and Audio Processing,* vol. 3, no. 1, pp. 72-83, 1995.

[15] O. Setyawati, M. Engenhorst, M. Bartels *et al.*, “Characterization of dry etching of TiO2/SiO2 distributed Bragg reflectors for tunable optical sensor arrays,” *Journal of Micro/Nanolithography, MEMS, and MOEMS,* vol. 9, no. 4, pp. 041110-041110, 2010.

[16] H. Mai, A. Albrecht, C. Woidt *et al.*, “3D nanoimprinted Fabry–Pérot filter arrays and methodologies for optical characterization,” *Applied Physics B: Lasers and Optics,* pp. 1-10, 2012.

[17] S. E. Stein, "Infrared Spectra," *NIST Chemistry WebBook, NIST Standard Reference Database Number 69*: National Institute of Standards and Technology, 2012.

[18] M. Moreira, and E. Mayoraz, “Improved Pairwise Coupling Classification with Correcting Classifiers,” in Proceedings of the 10th European Conference on Machine Learning, 1998, pp. 160-171.