ACTIVE CONTROL STRATEGIES FOR CHEMICAL SENSORS AND SENSOR ARRAYS

A Dissertation

by

RAKESH GOSANGI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Ricardo Gutierrez-Osuna
Committee Members,	Yoonsuck Choe
	Radu Stoleru
	Byung-Jun Yoon
Head of Department,	Duncan M. Walker

August 2013

Major Subject: Computer Engineering

ABSTRACT

Chemical sensors are generally used as one-dimensional devices, where one measures the sensor's response at a fixed setting, e.g., infrared absorption at a specific wavelength, or conductivity of a solid-state sensor at a specific operating temperature. In many cases, additional information can be extracted by modulating some internal property (e.g., temperature, voltage) of the sensor. However, this additional information comes at a cost (e.g., sensing times, power consumption), so offline optimization techniques (such as feature-subset selection) are commonly used to identify a subset of the most informative sensor tunings. An alternative to offline techniques is active sensing, where the sensor tunings are adapted in real-time based on the information obtained from previous measurements. Prior work in domains such as vision, robotics, and target tracking has shown that active sensing can schedule agile sensors to manage their sensing resources more efficiently than passive sensing, and also balance between sensing costs and performance. Inspired from the history of active sensing, in this dissertation, we developed active sensing algorithms that address three different computational problems in chemical sensing.

First, we consider the problem of classification with a single tunable chemical sensor. We formulate the classification problem as a partially observable Markov decision process, and solve it with a myopic algorithm. At each step, the algorithm estimates the utility of each sensing configuration as the difference between expected reduction in Bayesian risk and sensing cost, and selects the configuration with maximum utility. We evaluated this approach on simulated Fabry-Perot interferometers (FPI), and experimentally validated on metal-oxide (MOX) sensors. Our results show that the active sensing method obtains better classification performance than

passive sensing methods, and also is more robust to additive Gaussian noise in sensor measurements.

Second, we consider the problem of estimating concentrations of the constituents in a gas mixture using a tunable sensor. We formulate this multicomponent-analysis problem as that of probabilistic state estimation, where each state represents a different concentration profile. We maintain a belief distribution that assigns a probability to each profile, and update the distribution by incorporating the latest sensor measurements. To select the sensor's next operating configuration, we use a myopic algorithm that chooses the operating configuration expected to best reduce the uncertainty in the future belief distribution. We validated this approach on both simulated and real MOX sensors. The results again demonstrate improved estimation performance and robustness to noise.

Lastly, we present an algorithm that extends active sensing to sensor arrays. This algorithm borrows concepts from feature subset selection to enable an array of tunable sensors operate collaboratively for the classification of gas samples. The algorithm constructs an optimized action vector at each sensing step, which contains separate operating configurations for each sensor in the array. When dealing with sensor arrays, one needs to account for the correlation among sensors. To this end, we developed two objective functions: weighted Fisher scores, and dynamic mutual information, which can quantify the discriminatory information and redundancy of a given action vector with respect to the measurements already acquired. Once again, we validated the approach on simulated FPI arrays and experimentally tested it on an array of MOX sensors. The results show improved classification performance and robustness to additive noise.

ACKNOWLEDGEMENTS

I thank my advisor Dr. Ricardo Gutierrez-Osuna for his advice, support, and encouragement during my graduate studies. His guidance not only shaped this dissertation but also drove me to be a better researcher and a writer. I am grateful for his enthusiasm and commitment.

I also thank my committee members Dr. Yoonsuck Choe, Dr. Radu Stoleru, and Dr. Byung-Jun Yoon for their interest in my research and also for their suggestions and feedback.

I am always grateful for the unconditional love and support from my mom, dad, and sister.

Lastly, I thank my lab mates Daniel, Jong, Joseph, Jin, Sandesh, Folami, and Avinash for being great peers and making my grad school experience fun.

TABLE OF CONTENTS

Page

Page

Page

LIST OF FIGURES

Page

Page

Page

LIST OF TABLES

1. INTRODUCTION

Chemical sensors are generally used as one-dimensional devices, where one measures a particular physico-chemical property at a fixed setting: for example, the conductivity of a metal-oxide chemical sensor, or the absorption of an optical sensor, with this property preferably measured when the sensor is at a steady-state. This mode of operation, though straightforward, will generate only one measurement (or one feature) per sensor [1]. This measurement can be used to estimate concentration of the sample, if provided with a mapping between the sensor responses and concentrations of a target chemical. However, a single measurement cannot determine if the sensor responded due to the presence of the target, an interferent, or a combination of the two. Therefore, for realistic applications, the measurements from chemical sensors should be able to address complex problems such as distinguishing between different chemicals, or quantifying the concentrations of chemical mixtures, or detecting the presence of interferents. This is only possible if the sensors can provide multiple measurements thus constructing a high-dimensional feature space. In this context, a feature space corresponds to an abstract *n*-dimensional space in which a given chemical sample can be represented as a point; each dimension corresponds to a feature obtained from the chemical sensors.

Researchers have been exploring different methods to acquire multiple features from chemical sensors and thus effectively expand the feature space. One such method is to measure the physico-chemical property while modulating some internal or external configuration of the sensor. For example, measuring the conductivity of a metal-oxide (MOX) sensor at different temperatures, or measuring the absorption of an optical sensor at different incident wavelengths. Sensors that allow for one or more of their configurations to be modulated are called tunable sensors. In tunable chemical sensors, the tuning changes the reaction/interaction conditions

between the sensors and the samples, and thus provide with useful analytical information. An alternative to tunable sensors, for the purposes of obtaining obtain multiple features, is sensor arrays. In sensor arrays, a group of chemical sensors are deployed together on a single platform such that they are all exposed to the sample simultaneously, and the responses collected from all sensors construct the desired larger feature space.

With tunable sensors and sensors arrays, there is always a cost (such as power consumption, or time) associated with acquiring the features, and also the acquired features may be correlated with each other. To address these issues of sensing costs and feature correlation, researchers have been using optimization routines (e.g. feature subset selection) that identify a subset of the most informative sensors and sensor tunings. Feature selection can speed up the sensing process and also improve the performance of the sensors since irrelevant information is ignored. However, one of the drawbacks of feature selection is that it is passive; the optimal subset of tunings is hardcoded and the sensors driven accordingly, irrespective of what they are being exposed to. An alternative approach is active sensing, where the sensor tunings are selected in real-time based on the information obtained from previous measurements. Prior work in active sensing has shown that it can enable agile sensors to efficiently manage their sensing budgets and time resources for various computational tasks such as classification, detection, estimation, sampling, and tracking [2]. The last couple of decades have seen active sensing used in various domains such as computer vision, robotics, and surveillance. However, there has hardly been any work done in adding autonomy to chemical sensors.

The objective of this dissertation is to bridge the gap between active sensing and chemical sensors. To pursue this objective, we developed active chemical sensing algorithms that address the following three computational problems: (1) classification of chemicals with a single tunable sensor, (2) multicomponent analysis with a single tunable sensor, and (3)

classification of chemicals with an array of tunable sensors. Our hypothesis is that on-line active sensing strategies can improve the selectivity and performance of chemical sensors when compared to using the traditional non-adaptive sensing techniques.

1.1 Contributions of the dissertation

The main contributions of this dissertation can be summarized as follows:

- We developed active sensing algorithms that allow a tunable chemical sensor to optimize its operating configurations in real-time for classification and multicomponent analysis problems.
- We developed an active sensing algorithm for an array of tunable chemical sensors. This method allows for the sensors to coordinate their sensing actions to minimize redundancies in the acquired measurements, and collaboratively classify a given sample.
- We developed two objective filters that can evaluate the 'usefulness' of a given array configuration towards classifying the sample under investigation. These filters are a critical element in extending active sensing to multiple sensors.

1.2 Organization of the document

The rest of this dissertation is organized as follows. Chapter 2 provides an overview of the structure, working principles, and applications of chemical sensors, with a focus on MOX sensors and Farby-Perot Interferometers (FPIs). Chapter 3 reviews active sensing, and surveys its applications in computer vision, robotics, and target tracking. It then describes four mathematical frameworks commonly used to formulate active sensing problems. Chapter 4 presents an active sensing method for chemical classification problem, and experimentally validates it with varying sensing budgets and sensor noise levels. Similarly, chapter 5 presents an active sensing method for multicomponent analysis and the corresponding experimental work. Chapter 6 presents an

active sensing method for classification with sensor arrays. This chapter also describes two objective functions that evaluate the expected contributions of a given array configuration. Finally, chapter 7 presents the conclusions of this dissertation, and provides directions for future work.

2. CHEMICAL SENSORS

This chapter first provides a broad overview on the structure, working principles, and applications of chemical sensors. It then discusses tunable sensors and sensor arrays: two of the most commonly used approaches to expand chemical sensors' feature spaces. Finally, it describes the operating principles of MOX chemical sensors and FPIs, the two tunable sensor technologies used in this dissertation to evaluate the proposed active sensing methods.

2.1 Principles of chemical sensing

Chemical sensors are defined as electronic devices that convert physico-chemical properties of a gas sample into analytically useful electrical signals. A chemical sensor typically consists of two main components: (1) a chemically active material, and (2) a transducer. When the sensor is exposed to a chemical vapor, the active material adsorbs the vapor molecules and exhibits a change in one of its physical properties such as conductivity, capacitance, temperature, resonant frequency, or color. The transducer then converts these changes into electrical signals, which are processed to interpret the identity or concentration of the sample.

There are different types of chemically-active materials used in the development of chemosensors such as semiconducting metal-oxides [3], conducting organic polymers (polyethers, polyfluorocarbons, etc.) [4], molecular crystals (phthalocyanines, porphyrins, etc.) [5], and supramolecular crystals (zeolites, cyclophanes, cyclopeptides, etc.) [6]. These sensing materials are often combined with the appropriate catalysts or dopants to improve their sensitivity towards the target gases. In recent years, the use of bioligands (such as antibodies, enzymes, nucleic acids, cellular structures, aptamers, and oligonucleotides) as sensing materials has been gaining a lot of interest because they have the ability to strongly bind with proteins, hormones, vitamins, sugars, and other biomarkers. Detecting of such analytes is important in

clinical diagnostics, research in life sciences, and pharmaceuticals. These bioligand-based sensors are often called 'biosensors' to be able to distinguish them from the conventional chemical sensors.

As with materials, there are various types of transduction principles used in chemical sensors such as resistance, impedance, current, capacitance, work function, gravimetry, absorption, and temperature. The choice of the transduction principle is usually coupled with the choice of the sensing material. Table 1 presents a list (not exhaustive) of different types of chemical sensors, their sensing materials, the corresponding transduction principles, and some of the advantages and disadvantages.

Table 1 Different types of chemical sensors, materials, transduction principles. Abbreviations: MOX – Metal-Oxide chemical sensors, CP – Conducting Polymer sensors, QCM – Quartz Crystal Microbalance sensors, OP – Optical sensors, EC – Electrochemical sensors, MOSFET – Metal-Oxide Semiconductor Field Effect Transistors, and SAW – Surface Acoustic Wave sensors.

Sensors	**Materials**	**Principle**	**Advantages**	**Disadvantages**
MOX	SiO_2, TiO_2, ZnO_2	Conductivity	Inexpensive, micro-fabricated	High operating temperatures, sensitive to humidity, drift
CP	Polypyrrole, polyaniline	Conductivity	Inexpensive, operate at room temperature	Sensitive to humidity, long term drift
QCM	Quartz crystals	Gravimetry	Low power, small size, rugged	Low selectivity, sensitive to temperature changes and flow rates
OP	Optical fibers	Fluorescence	Selectivity, immune to electrical noise	Expensive, sensitive to ambient light
EC	Pt electrodes, Au catalyst	Amperometry	Low cost, rugged	Size, short shelf life
MOSFET	SiO_2, TiO_2, ZnO_2,	Capacitance	Inexpensive, micro-fabricated	High operating temperatures, limited selectivity
SAW	ZnO, lithium niobate, polymers	Piezoelectricity	High sensitivity, response time, reproducibility	Sensitive to temperature changes, low selectivity

There is a second group of chemical sensors (called spectroscopic chemical sensors), whose operation is not based on chemical reactions between samples and sensing materials; instead they measure the interactions of the samples with electromagnetic radiation. In spectroscopic chemical sensors, the sensing apparatus usually consists of three elements: an emitter (a source that emits electromagnetic radiation of a known spectral range), a gas chamber (where the sample interacts with the radiation), and a detector (that measures the emitted radiation after interaction with the sample). The differences in the amplitudes of the emitted radiation and the detected radiation can measure various physical properties of the sample such as infrared absorption [7], or nuclear magnetic resonance (NMR) [8]. These properties can be used to determine the identity and composition of the sample. For a more thorough review on different types of chemical sensors and their working principles, please refer to Janata [9].

The desired properties of chemical sensors include highly selectivity towards the intended target gases, low sensitivity towards interferents, long-term stability (they should not drift and be rugged), small size (making them portable), low manufacturing costs, and high reproducibility (which allows for large-scale manufacturing and deployment). Unfortunately, a chemical sensor exhibiting all these desired properties is an unrealizable ideal.

Of the various properties mentioned above, sensitivity and selectivity are the most important aspects of a chemical sensor because they determine the usefulness of a sensor for any given application. These properties are determined by the type of reactions that take place between the target analytes and the sensing materials. Ideally, the reactions are fast, which ensures high sensitivity and low response times, and reversible, which ensures that the reaction rates would follow changes in the concentration of the samples and therefore be able to quantify them. Also, the reversible nature ensures the sensors are reusable. However, high sensitivity and selectivity on one end, and reversibility on the other end, impose contradictory constraints on the

choice of sensing material [10]. Strong sensitivity and selectivity are associated with strong chemical reactions between the sensing materials and target analytes, but reversibility is associated with weak chemical reactions. The strength of a chemical reaction is usually measured in terms of the change of free energy. The selectivity of a chemical sensor is also affected by the choice of the transduction principle.

2.1.1 Applications

The most important applications of chemical sensors could be broken down into three categories [11]: (1) environmental monitoring, (2) medical diagnostics, and (3) food and beverage. The following paragraphs will provide a short overview on these three applications along with relevant examples.

Environmental monitoring encompasses a broad range of activities that involves tracking concentrations of various chemicals emitted into the environment. These chemicals are generated either through man-made or natural processes, and are of public concern or scientific interest. Of the three applications mentioned above, environmental monitoring is generally the most challenging because in this application, the sensors have to detect and quantify target analytes in dynamic (due to changing temperatures and humidity) and sometimes turbulent (due to wind patterns) backgrounds. Despite this challenge, research and commercial interests in environmental monitoring haven't decreased in the last couple of decades, especially due to the increased awareness on the effects of pollution on health and environmental quality [12]. A few examples of environmental monitoring applications include: monitoring air quality for particulates (e.g. pollen [13]), pollutants (e.g. Nitrogen dioxide [14]), and greenhouse gases (e.g. methane [15]). Apart from air quality monitoring, chemosensors are also used in aquatic environments for measuring salinity levels of oceans [16], nutrient contents [17], dissolved gases

in aquatic eco systems [18], etc. Other examples include monitoring of soil quality [19], wildlife habitats [20], and volcanic activity [21].

Many diseases and intoxications are often accompanied with characteristic changes in properties of bodily fluids (such as breathe, sweat, blood, tears, etc.) and their recognition can provide important diagnostic clues [22]. Due to the diverse nature of the target analytes, developing a universal chemical sensing system for medical diagnostics is impractical. Therefore, the sensing systems have to be adapted to particular needs or specific diseases. Though gas chromatography is widely accepted as a standard tool for analysis amongst chemists, it hasn't been credited as a diagnostic tool because of its cost, complexity of usage (requires trained professionals), and long measurement times. In diagnostics, the use of biosensors is more prevalent than the conventional chemical sensors because biosensors have ability to bind with analytes of interest, which include proteins, hormones, vitamins, sugars, etc. A few examples of diagnostic applications include detection of cancer markers [23], allergy markers [24], cardiac muscle injuries [25], and diabetes management [26]. In the recent years, there is also a growing interest the use of wearable chemical sensors [12], and biochips [27]. Such portable chemosensors are critical in driving the idea of point-of-care-diagnostics.

The third main application of chemical sensors is in the area of food and beverages, which includes inspection of quality, supervision of manufacturing processes, and investigation of shelf life [11]. Food aromas are usually complex and require sensory analysis by a panel of human experts. This process is costly because it needs trained professions and could also be inconsistent due to variability between the individual human evaluators [28]. Therefore, chemical sensors have become a popular alternative for objective characterization of food flavors [29]. A few examples of applications in food and beverage include: process monitoring, e.g. discrimination between different fermentation stages of wines [30], shelf life investigation, e.g.

discrimination between ripening stages of applies [31], freshness evaluation, e.g. spoilage classification of fish [32], authenticity assessment for wines [33], and other applications such as quality of olive oils [34].

Apart from the three main applications discussed above, chemosensors are also used in other areas: detection of explosive substances such as trinitrotoluene (TNT), dinitrotoluene (DNT), and cyclonite (RDX) [35-37], detection and location of landmines [38], intruder detection in urban areas [39], detecting biological and chemical warfare agents [40, 41], rescue robotics [42], and cosmetics and fragrance [43]. Recent advances in nano-technology and nano-fabrication have led to the development of many light-weight, power-efficient chemical sensors. These sensors can be deployed on mobile platforms like cellular phones, which allows to map airborne toxins or pollutants over large urban areas [44], and effectively crowd source chemical sensing.

2.2 Tunable sensors

As mentioned in chapter 1, the feature space of a chemical sensor can be expanded by measuring one of its physical properties while modulating some internal or external configuration of the sensor. In this section, we present examples of tunable chemical sensors, where such parameter modulation has been achieved. They are divided into two categories based on the type of modulation: internal or external.

2.2.1 Internal parameter modulation

The first example of internal parameter modulation is that of the operating voltage in voltammetric chemical sensors. Voltammetry encompasses a group of electrochemical techniques, where researches study the changes of current in response to changes in applied potential. Voltammetry is one of the oldest chemosensing techniques, and is widely used to

identify or quantify electro-active chemical species in liquid phases: for example dissolved oxygen in blood and tissue [45], hydrogen peroxide in pools [46], or chlorine in drinking water [47]. A voltammetric sensor is composed of an electrochemical cell (where the analyte is placed), a working electrode, a reference electrode, and an auxiliary electrode. A potential is applied between the working and reference electrodes, and the current at the working electrode is measured. This potential could be varied with linear, stepwise, or cyclic waveforms [48, 49]. The changes in the potential cause reduction or oxidation of the analytes, and these reactions are registered as peaks or troughs in the current signal (see Figure 1). Thus, the modulation of the applied potential can be used to acquire multiple features from a given voltammetric sensor.

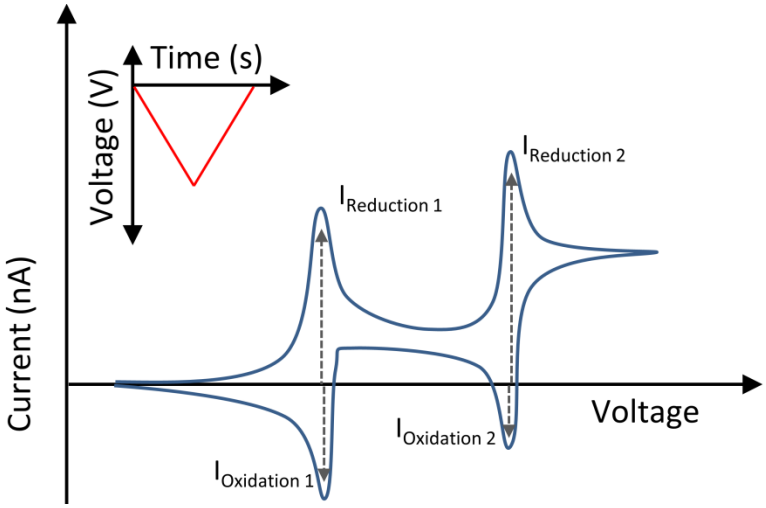

Figure 1 An example of voltammetry, where the potential is varied with a triangular wave resulting in two reversible redox reactions. This figure was adapted from Gutierrez-Osuna et al. [1].

The second example of internal modulation is that of operating wavelengths in Quantum Cascade Laser (QCL) based chemical sensors. QCLs are semiconductor light sources that emit radiation in the mid-to-far infrared range [50]. QCLs are used in spectroscopic applications [51],

where they are incorporated into the so-called external cavity architecture. In this architecture, the QCL device is aligned with a collimating lens, a diffraction grating, and a mirror. When the diffraction grating is rotated, the wavelength can be tuned across the entire bandwidth of the QCL. Therefore, in QCL-based chemical sensors, the angel of the diffraction grating can be modulated to obtain different features.

Two other examples of internal parameter modulation include modulation of temperatures in MOX sensors [52], and modulation of mirror distance in FPI sensors. These two examples are discussed in a greater detail in section 2.4.1 and section 2.5.2 respectively.

2.2.2 External parameter modulation

A chemical sensor's feature space may also be expanded by measuring the physical property while modulating one of its external parameters. The first such example is the modulation of the sample-delivery process, which can be achieved by using actuating valves and dilutors. These valves and dilutors are located upstream from the sensor and can be modulated to either vary the concentration of the analyte delivered to the sensor [53], or vary how long the sensors are exposed to the analytes [54], or switch between the analyte and a reference gas [55]. The transient responses collected form the sensor placed downstream can then be processed to obtain various features.

The second example of external parameter modulation is the temperature of external catalysts, as demonstrated in Schweizer-Berberich et al. [56]. In this paper, the authors used a platinum catalyst located upstream of an amperometric sensor. The catalyst was heated to different temperatures, which caused the incoming analytes to be decomposed (oxidized) at different rates, and the sensor (located downstream) detected the products resulting from the reactions upstream. Thus by varying the catalyst's temperature, the authors were able to expand

the sensor's feature space and eventually use the measurements to characterize freshness of various food products.

2.3 Sensor arrays

An alternative to tunable sensors, for the purposes of extending the feature space and gain additional information in chemosensing, is to use sensor arrays. Early work in this area by Persaud and Dodd [57] sought inspiration from biological olfactory systems, which contain numerous olfactory receptors working simultaneously to identify different odors. Many experiments have shown that the olfactory receptors are not highly selective towards any specific analyte; in fact, each receptor responds to many analytes and many receptors respond to any given analyte [58]. Similarly, in sensor arrays, the constituent sensors are usually cross-sensitive, in that they respond to many chemicals. As a result, the array as-a-whole responds to a wide range of chemicals.

Based on the homogeneity of the sensing materials and transducers, sensor arrays could be classified into four categories. In the first type of arrays, all the sensors are identical and use the same type of transducers. One advantage of such homogenous arrays is that the measurements can be averaged to reduce errors. For example, Wilson et al. [59] have experimentally shown that the variance in the averaged output of a homogenous array of MOX sensors is inversely proportional to the square root of the number of sensors in the array. To acquire multiple features from such arrays, the sensors have to be modulated using any of the techniques described in the previous section. However, if each sensor in the array is modulated independently, a homogenous array can be used to acquire multiple features simultaneously. For example, given an array of identical MOX sensors, each sensor can be operated at a different temperature and thus multiple measurements obtained simultaneously. However, in this mode of

operation, there could be crosstalk between the sensors: the reactions and configurations of one sensor could affect the responses of other sensors.

In the second type of arrays, the sensors are made of the same materials but use different transducers. As an example, Hagleitner et al. [60] developed a single-chip gas detection system which consisted of a polymer layer connected to three different transducers: a micro-machined cantilever, a planar capacitor, and thermoelectric calorimeter. The system was use to classify and quantify various organic compounds and their mixtures [61]. Likewise, Polk et al. [62] developed an array of eight field-effect-transistor-based sensors for simultaneous potentiometric and impedance measurements.

In the third type of arrays, the sensors are made of different materials but use same transduction principles. One such example, is array of MOX sensors developed by Tomchenko et al. [63] for flue gas analysis. The authors fabricated an array consisting of five different MOXs (SnO_2, ZnO, WO_3, CuO, and In_2O_3) deposited onto a common platform and examined the changes in the sensors' conductivity towards different gaseous pollutants. Another example includes colorimetric sensor arrays developed by Suslick et al. [64]. Unlike most of the other examples provided in this section, colorimetric sensor arrays are not tunable. These arrays include various metalloporphyrin dyes dissolved in polymer films and deposited as small spots on teflon sheets. When this array is exposed to a chemical, the dyes react and show a change in their colors. The difference between the original dye colors and the changed colors, measured in an RGB space, provides unique signatures to identify the chemicals.

The fourth type of sensor arrays contains sensors made of different materials and working on different principles, but deployed on a common platform. One such example includes the gas sensor microsystem developed by Li et al. [65], which was comprised of two sets of polymer-based sensors connected to capacitive and gravimetric transducers, and another

set of MOX sensors, all deployed on a single chip. Another example is a multimodal ozone sensing system developed by Sauter et al. [66]. This system consisted of electrochemical sensors, MOX sensors, and shear mode resonator sensors, all mounted on separate modules. The modules were exposed to the gases with a parallel-flow delivery system. This example could be considered as more of a multimodal sensing system rather than a sensor array. For a more thorough review on different type of sensor arrays and their applications, please refer to review article by Albert et al. [67].

2.4 Semiconducting metal-oxide gas sensors

In this dissertation, we tested active sensing algorithms on two types of tunable chemical sensors: MOX gas sensors and FPIs. This section describes the structure and operating principles of MOX sensors. It also describes how temperature modulation can be used to acquire multiple measurements from a given MOX sensor, thus increasing the dimensionality of the sensor's feature space. It concludes with a review of two papers that present offline approaches for optimizing temperature programs applied to MOX sensors. The next section describes the operating principles and optical properties of a Fabry-Perot (FP) filter. It also explains how the distance between the mirrors of a FP filter can be modulated to allow different wavelengths to pass through, which allows for acquiring different measurements and thus effectively expands the sensor's feature space.

MOX gas sensors are chemoresistors; they exhibit a change in resistance when exposed to chemical vapors. The surface of MOX sensor is made of semiconducting compounds such as SnO_2, ZnO_2, TiO_2, or IrO_2; of these compounds, SnO_2 is most widely used in commercial devices. The MOX surface is doped with a thin layer of a heavy metal such as palladium (Pd), platinum (Pt), or Iridium (Ir) [22], which acts as a catalyst for the reactions between the sensor surface and the gas sample.

The operation of a MOX sensor is based on the chemical sensitivity of the semiconducting surfaces. The semiconducting materials, when heated to a temperature in the range of 200-400 °C, adsorb the ambient oxygen as negatively charged ions: O^-, O_2^-, and O^{2-}. When this heated surface is exposed to a reducing gas (e.g. carbon monoxide (CO), or methane (CH_4)), the adsorbed oxygen atoms react with the gas to release the trapped electrons. As an example, the reactions of CO (carbon monoxide) with negatively charged oxygen ions [68] are summarized in the equations below:

$$\begin{aligned} O_2^- + 2CO &\rightarrow 2CO_2 + e^- \\ O^- + CO &\rightarrow CO_2 + e^- \\ O^{2-} + CO &\rightarrow CO_2 + 2e^- \end{aligned} \tag{1}$$

The release of these electrons causes an increase in the conductivity of the material. Contrarily, if the sensor is exposed to an oxidizing gas (e.g. nitrogen dioxide (NO_2)), the oxygen ions adsorbed to the sensor surface trap the free electrons. This results in a decrease in the conductivity of the sensing material. The changes in conductivity can be measured by observing the change in resistance/conductance between a pair of electrodes that are placed across the sensing material[1].

The MOX surface can be heated by applying a voltage across a resistive heating element that is incorporated into the sensor. There are two designs of incorporating the heating element into the sensor: (a) tubular design or (b) planar design, as shown in Figure 2. With tubular design, the sensor is comprised of an alumina ceramic tube coated with sintered semiconducting material, and the heating coil runs through the center of the ceramic tube. Taguchi type sensors, which are used for experimental work in this dissertation, use this type of tubular design.

[1] Interestingly, the layout of these electrodes affects the responses of MOX sensors and therefore influences the sensor's selectivity. A few examples of these electrode arrangements include inter-digital, transmission line measurements, and 4-proble arrangement.

However, most of the modern commercially available MOX sensors use planar design. With planar design, the metal-oxide is printed as a thin film (using techniques such as silk-screen printing, dip-coating, sputtering, or chemical vapor deposition) onto a substrate made of alumina or silica. The resistive heating element is placed under the substrate. Planar designs are more efficient than tubular designs in developing smaller sensors, sensor arrays, and for mass production.

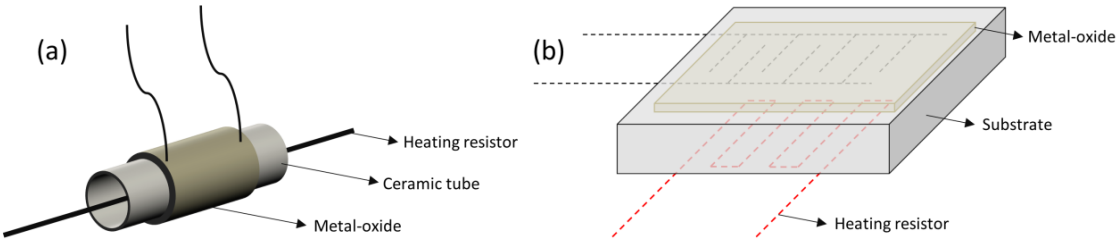

Figure 2 Two types of design used for MOX sensors: (a) tubular design, and (b) planar design. This figure was adapted from Albert et al. [67].

The changes in conductivity of a semiconductor surface can be explained by the Potential barrier theory. The MOX surface is comprised of many microscopic grains which trap the electrons, as shown in Figure 3. The trapping of the electrons builds up a potential barrier between the grains. When the surface reacts with a reducing gas, electrons are released lowering the potential barrier and increasing the conductivity of the surface. The height of the potential barrier and the conductance of the semiconductor are related by the following expression:

$$G = G_0 \exp\left(-\frac{\phi_b}{k_B T}\right) \tag{2}$$

where G_0 is a constant that depends on the semiconductor geometry, mobility of the electrons, and density of the semiconductor donors on the surface, ϕ_b is the barrier height, k_B is Boltzmann's constant, and T is the temperature.

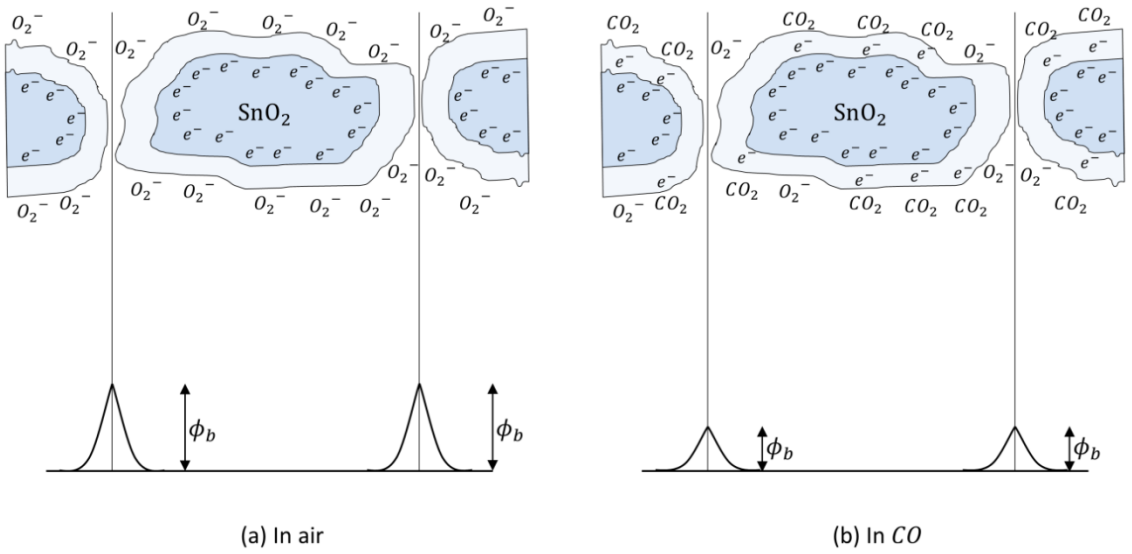

Figure 3 MOX grains at a microscopic level. When in the presence of CO, the potential barrier between the grains reduces causing an increase in the conductivity. This figure was adapted from Gutierrez-Galvez [69].

MOX sensors are widely used in gas sensing; some of the earliest work in the field of electronic noses and chemical sensing has been conducted on MOX sensors [57, 70]. MOX sensors are robust, inexpensive, and commercially the most readily available set of chemical sensors. Also, they have low detection thresholds (as low as a few ppm, depending on the chemical target), but they are cross-sensitive. Thus, MOX sensors are highly sensitive but lack specificity. To address this issue of low specificity, researchers have been using the idea of temperature modulation, which is reviewed in the following section.

2.4.1 Temperature modulation of MOX sensors

Temperature modulation leverages the fact that the reaction rates of different volatile compounds and the stability of surface-adsorbed oxygen ions depends on the operating temperature of the MOX sensor [71]. As a result, modulating the operating temperature can give rise to gas-specific temporal signatures, which provides a wealth of discriminatory and quantitative information [52]. This idea of exploiting variations in the conductance of a MOX sensor under temperature modulation has been studied for more than two decades. Also, the advent of micro-hot plates [72, 73], which have low thermal masses allowing for millisecond-scale temperature modulations that makes the temperature variations faster than the chemical processes occurring on the surface, has further increased the interest in temperature programming.

The waveforms used for temperature modulation could broadly be classified into two categories: temperature cycling and thermal transients (see Figure 4). In temperature cycling, the sensor is driven with a periodic waveform such as a sinusoid or a triangle. These waveforms are typically chosen to be slow-varying (of low frequency) to avoid sudden changes in the temperature of the sensor. To our knowledge, the work of Sears et al. [74] is one of the earliest studies on temperature cycling for MOX sensors. In this study, the authors used sinusoidal signals to drive the heater voltage of Figaro (TGS 812) sensors, when exposed to various gases such as propane, carbon monoxide, acetone, and methane. The authors then analyzed the variation in the conductance of the MOX sensor as a function of time, and found that it showed distinct peaks. More importantly, they found that the positions of the peaks and the number of peaks were dependent not only on the applied sinusoidal signals but also on the identity and concentration of the gas. The peaks in the conductance are caused when the sensor reaches an optimum catalytic oxidation temperature; this temperature is dependent on the analyte.

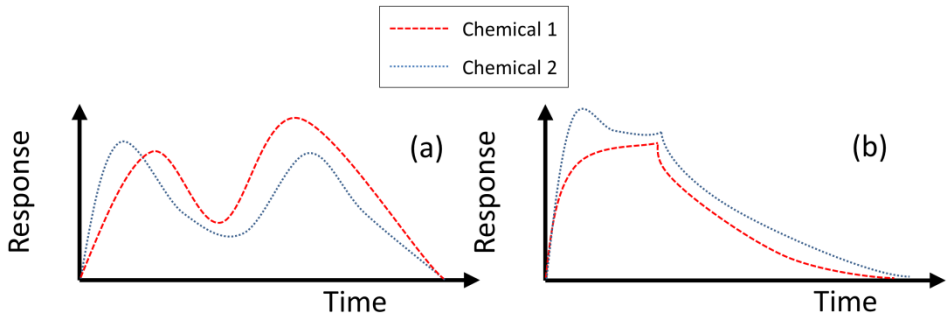

Figure 4 Examples of temperature modulated responses of a MOX chemical sensor: (a) temperature cycling, (b) thermal transients.

The signals resulting from temperature cycling may be analyzed using simple methods such as sampling the signal at different points to form a feature vector, or extracting properties such as maxima, minima, and time constants. However, due to periodic nature of the temperature cycling waveforms, the conductivity of MOX sensors also changes in a periodic manner. Therefore, the signals can be better analyzed using frequency-analysis methods such as Fast Fourier Transform (FFT) and Discrete Wavelet Transforms (DWT). Nakata and co-workers [75-77] have used FFT extensively to analyze temperature cycled responses. Their work has demonstrated that the coefficients of higher harmonics can be used for classification of gases and also for quantitative analysis of mixtures. Similar results were presented by Heilig et al. [78] and Fort et al. [79]. Ionescu et al. [80] applied DWTs to extract features from sensor responses, and showed that DWT features are more informative than those provided by the FFT. The more recent works in temperature cycling [81-84] were conducted on micro-hotplate MOX sensors. As mentioned before, due the low thermal masses of these sensors, micro-hotplate sensors can be modulated at much faster rates than the traditional Taguchi type sensors.

With thermal transients, the sensor is driven with a sudden change in heater voltage (for example a step or a pulse), which induces a fast change in the temperature of the sensor. To our

knowledge, this idea was first studied by Hiranaka et al. [85]. In this paper, the authors exposed SnO_2 gas sensors to various chemicals (methanol, ethanol, acetic acid, benzene, hexane etc.) and drove the sensor with a fixed voltage. Once the sensor's response had reached a steady-state, the applied voltage was switched off, which resulted in a sudden change in the temperature of the sensor and therefore yielded a transient response. The authors analyzed the transient signals and found that the peak times were mostly dependent on the identities of the samples and to some extent on their concentrations. Similar results were presented by Yea et al. [86], and Kato et al. [87]. More recent examples include the works of Huang et al. [88], and Ngo et al. [89].

As with temperature cycling, there are a wide variety of methods used to extract information from the transient responses of MOX sensor. The simplest methods include extracting the area under the transient, rise-time, derivate, the steady-state value, or the response could be sampled at various points to create a feature vector. However, researchers have explored many other sophisticated methods to analyze sensor transients, mostly derived from signal processing. In these methods, the transients are fitted to an analytical model and the resulting model coefficients are used as features. A few examples of such analytical models include: dynamic moments [90], exponential models [91, 92], auto-regressive models [93], and dynamic Bayesian networks [94].

In a recent paper, Martinelli et al. [95] proposed an adaptive pulse modulation system for MOX sensors. Here, the authors reorganized sensor's electronic circuit such that the sensor resistor was connected in a closed-loop with the sensor's heating resistor. Therefore, any changes in the sensor resistor are reflected as changes in the operating temperature. The heater was then modulated with a series of pulses and the authors found that the sensor's resistance also varied as a sequence of pulses but of shorter and varying durations. More importantly, the authors found the pulse durations were dependent on the analyte the sensor was exposed to.

2.4.2 Temperature program optimization

Though researchers have used a wide array of waveforms for temperature modulation, as detailed in section 2.4.1, most of these studies have been empirical. To our knowledge, there are only two studies [82, 96] that have proposed systematic approaches for optimizing the temperature waveforms applied to MOX sensors. The basic strategy behind these works is to first create a dynamic model of the sensor's response from experimental data, and then use this model in combination with an optimization procedure to identify the temperature waveform that best distinguishes between a given set of chemicals.

Kunt et al. [96] used a wavelet network to learn the sensor dynamics from experimental data. The model predicted the next conductance value of the sensor (y_{i+1}) from its previous values $\{y_k\}_{k=i}^{i-n_y+1}$ as well as from the next and previous values of the operating temperatures $\{u_k\}_{k=i+1}^{i-n_u+2}$

$$y_{i+1} = F\left(y_i, y_{i-1}, , \dots, y_{i-n_y+1}, u_{i+1}, u_i, \dots, u_{i-n_u+2}\right) \tag{3}$$

where n_y and n_u represent the model order. The authors then applied an optimization routine to find the optimal program $\{u_i\}_{i=1}^{T}$ that maximizes the distance between the temperature-modulated sensor responses to two target gases (A and B), as predicted by the wavelet model:

$$\{u_i\}_{i=1}^{T} = \underset{u_1, u_2, \dots u_T}{\arg\max}[dist(y^A, y^B)] \tag{4}$$

The optimization was subjected to a smoothness constraint to avoid drastic changes between consecutive temperatures. When applied to the discrimination of methanol and ethanol using a micro-hotplate MOX sensor, the optimization method returned a temperature program such that the sensor response to the two analytes was nearly orthogonal.

Vergara et al. [82] proposed a system-identification method for determining optimal temperature modulation frequencies. Their method was based on pseudorandom binary sequences (PRBS) and maximum length sequences (MLS). PRBS and MLS are square-wave signals with several interesting properties: they are repeatable, which ensures reproducibility; they have a flat power spectrum over a large frequency range, which makes them suitable for system identification; and they have maximum length, so the impulse response of the system can be estimated from the cross-correlation. In this method, a PRBS-MLS is used to drive the sensor heater while the sensors are exposed to various target compounds. For each individual target, the impulse response $h(t)$ is computed as the cross-correlation between the excitation signal (PRBS) and the sensor response, and the spectral components are computed from the Fast Fourier Transform of $h(t)$. Finally, each individual frequency is ranked based on its information content (measured as the ratio of between to within-class scatter), and a subset of the most informative frequencies is selected. Using this procedure, a single sensor at three modulating frequencies was sufficient to discriminate and quantify various gases and their mixtures.

2.5 Fabry-Perot interferometers

The FP filter was first developed by Charles Fabry and Alfred Perot in 1897 [97]. Over the last century, FPIs have been widely used as laboratory instruments in various research areas like metrology, spectroscopy, astrophysics, and chemical detection. An FP filter is a simple optical device comprised of two partially reflective mirrors placed parallel and separated by a distance, thus forming an optical resonance cavity. When a light beam is projected onto the outer surface of one of the mirrors, a part of the beam transmits through and undergoes several reflections back and forth between the two mirrors. However, at each reflection, a small portion of the light transmits through the second mirror. As a result, as shown in Figure 5, a large number of parallel beams (of decreasing amplitudes) emerge from the second mirror directed at

the same angle as the original incident beam, but with a phase difference induced due to the distinct lengths travelled by each beam. When the distance between the mirrors is adjusted aptly, the parallel beams can exhibit constructive interference[2] causing sharp interference fringes.

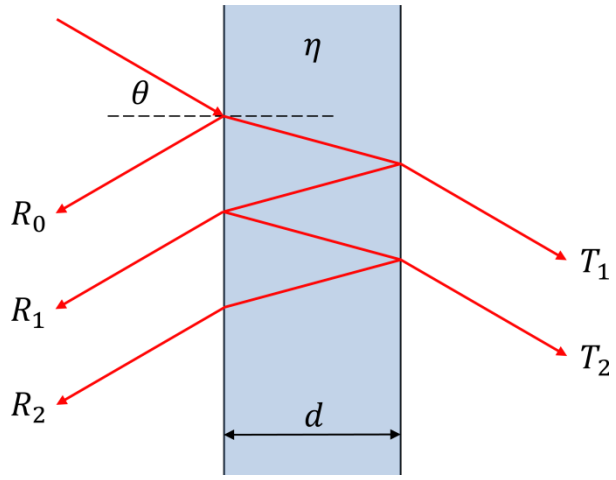

Figure 5 The incident beam undergoes multiple reflections in the optical cavity resulting in a number of parallel beams (of decreasing amplitudes) emitted from the second mirror.

2.5.1 Optical properties of FP filters

Let the two mirrors comprising the FP filter in Figure 5 be separated by a distance d, and the refractive index of the material in the optical cavity be η. When this filter is exposed to a light beam of wavelength λ at an incident angle θ, the beam goes through many reflections resulting in a series of parallel transmitted beams: $T_1, T_2, \dots, T_\infty$, where the phase difference between two consecutive transmitted beams is:

[2] Constructive interference is an optical phenomenon, where parallel light waves (with a phase difference of 2π) superimpose resulting in a wave of higher amplitude.

$$\phi = \frac{4\pi\eta d cos(\theta)}{\lambda} \tag{5}$$

The reflections, refractions, and transmissions of the waves in an FP filter are characterized by a series of optical equations called the Airy formulas of transmission [98]. These equations characterize the energy transmission coefficient of an FP filter as a function the optical properties of the mirrors, and the material in the cavity. Based on the Airy formulas, the amplitude of the k^{th} transmitted light beam (T_k) is given as:

$$A_k = T\left(1 + R^{k-1}e^{i(k-1)\phi}\right) \tag{6}$$

where $T = t_1 t_2$ is the product of the transmission coefficients of the two mirrors, and $R = r_1 r_2$ is the product of the reflection coefficients of the inner sides of the two mirrors. The cumulative amplitude of all waves is:

$$\sum_{k=1}^{\infty} A(T_k) = \frac{T}{1 - Re^{i\phi}} \tag{7}$$

Likewise, the cumulative energy of all waves is:

$$E_T = \frac{T^2}{\left(1 + R^2 - 2Rcos(\phi)\right)} \tag{8}$$

By replacing $\cos(\phi)$ with $1 - \sin^2\left(\frac{\phi}{2}\right)$, the above equation could be rewritten as:

$$E_T = \frac{T^2}{(1-R)^2}\left(\frac{1}{1 + \frac{4R}{(1-R)^2}\sin^2(\phi/2)}\right) = \frac{\left(\frac{T}{1-R}\right)^2}{1 + Gsin^2\left(\frac{\phi}{2}\right)} \tag{9}$$

where $G = \frac{4R}{(1-R)^2}$ is called coefficient of finesse. If the mirrors do not absorb any radiation, i.e. $T + R = 1$, then the transmission energy equation reduces to $E_T = \frac{1}{1+Gsin^2\left(\frac{\phi}{2}\right)}$. This energy is the highest when ϕ is an integer multiple of 2π: $\phi = 2m\pi$ (m is an integer).

$$\phi = \frac{4\pi\eta d cos(\theta)}{\lambda} = 2m\pi \tag{10}$$

Assuming the incident light is almost normal (i.e., $\theta \to 0$ and $\cos(\theta) \to 1$), and the medium in the optical cavity is air ($\eta = 1$), eq. (10) reduces to $d = \frac{m\lambda}{2}$. In other words, an FP filter demonstrates constructive interference when the distance between the mirrors is an integer multiple of half the wavelength of incident waves.

Eq. (10) can be rewritten in the frequency domain as:

$$f_m = m\left(\frac{c}{2\eta d cos(\theta)}\right) = m(\Delta\nu) \tag{11}$$

where, c is the speed of light, and $\Delta\nu$ is the free spectral range (which is the distance between two adjacent transmission peaks). Again with $\theta \to 0$ and $\eta = 1$, $\Delta\nu = \frac{c}{2d}$. $\Delta\nu$, called the free spectral range of the filter, is a very important property because it is required to estimate the finesse (F) of the filter. Finesse characterizes the resolving power of the filter or the ability to distinguish between two adjacent fringes. Finesse is estimated as the ratio of the free spectral range ($\Delta\nu$) to the full-width at half maximum (FWHM[3] or $\delta\nu$) of the transmission peaks. FWHM ($\delta\nu$) can be estimated using the energy equation (eq. (7)). Namely, the difference between the two solutions to the equation $E_T = 0.5$ is the FWHM:

[3] The full width at half maximum (FWHM) is a parameter commonly used to describe the width of a "bump" on a function. It is given by the distance between points of the function at which the function reaches half its maximum value.

$$\delta\nu = \frac{c(1-R)}{2\pi d\sqrt{R}} = \frac{c}{2\pi d\sqrt{G}} \tag{12}$$

Thus the finesse F is given as:

$$F = \frac{\Delta\nu}{\delta\nu} = \pi\sqrt{G} \tag{13}$$

Figure 6 demonstrates the effects of finesse using two simulated Gaussian transmission peaks. With increasing finesse, the overlap between the neighboring fringes reduces thus increasing the resolving power of the filter. Eq. (13) shows that finesse is dependent only on the reflection coefficients of the mirrors. Therefore, the resolving power of an FP filter is highest when the reflection coefficients are close to 1.

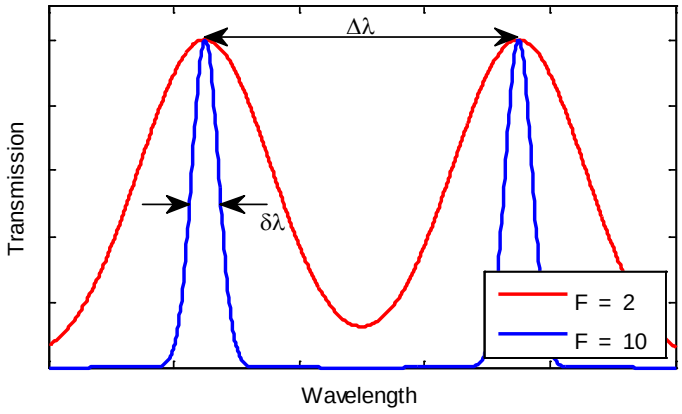

Figure 6 The effect of finesse on the resolving power of an FPI.

2.5.2 Design and actuation

The Airy formulas show that an FP filter can be tuned to transmit different wavelengths by changing the distance between the mirrors. If the distance between the mirrors of an FP filter

is tunable, it is called an interferometer (FPI) and otherwise an etalon. Typically in an FPI, one of the two mirrors is fixed and the other mirror is moved to adjust the transmitted wavelengths. This tuning is achieved using electrostatic actuation [99, 100], where a voltage is applied between the two mirrors inducing an electric field in the optical cavity and creating an electrostatic force between the mirrors. The electrostatic force is counteracted with another force induced by mechanical springs that are attached to the adjustable mirror. Thus, by changing the applied voltage, the distance between the mirrors can be adjusted.

As shown in Figure 7, FPIs can be used for gas sensing when combined with three other components: (a) an infrared source (which generates the infrared radiation of a fixed spectral range), (b) a gas chamber (where the sample is placed for analysis), and (c) a pyro-electric detector (which detects the radiation after it had passed through the gas chamber). If the distance between the mirrors is tuned to transmit different wavelengths, while a sample placed in the chamber, then the signal collected by the detector constitutes the transmission spectrum of the sample. The IR source (emitter) is combined with a collimator to ensure that the emitted radiation is parallel. The radiation after passing through the filter is focused onto the detector with a focusing lens. This increases the energy (or strength) of the signal at the detector's end.

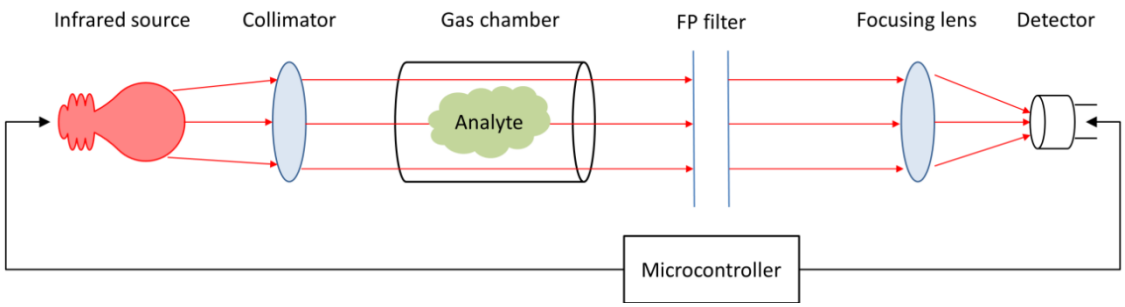

Figure 7 Design of an FPI for chemical sensing.

FPIs provide cheap and portable alternative to Fourier Transform Infrared Spectroscopy (FTIR). FTIR is one of the most popular IR spectroscopic technique. FTIR provides high resolution spectra (down to 0.001 cm^{-1}) covering a wide spectral range (200−10000 cm^{-1} at 2 cm^{-1} resolution). However, FTIR is mostly limited to laboratory settings due to its cost, size, and power requirements. FPI, though an alternative, comes at the price of limited spectral range and resolution.

3. ACTIVE SENSING

This chapter presents a short history on active sensing, a survey of its applications in computer vision, robotics, and target tracking. It also describes four mathematical frameworks generally used for active sensing: Markov decision processes, partially observable Markov decision processes, multi-armed bandit processes, and recursive state estimation. The chapter concludes with reviews on related work in machine learning and chemical sensing.

The term 'active sensor' generally refers to a device that transmits energy to make measurements (e.g. sonar, radar). Active sensors are inspired from biological sensing systems, where animals use self-generated energy to probe the environment [101]. For example, bats generate ultrasound waves (of frequencies ranging from 12 to 200 KHz) through their larynx, and then interpret the interference patterns and time delays of the echoed waves to measure distances, directions, and elevations of their surroundings [102]. This phenomenon is called echolocation and is observed in many other animals such as dolphins, electric fish, porpoises, sperm whales, shrews, and tenrecs.

A second, less common connotation of the term active sensing comes from the robotics literature, where Bajcsy [103] defines it as '*controlling strategies applied to the data acquisition process which will depend on the current state of the data interpretation and the goal or the task of the process*'. The main difference between these two connotations is that, in the second, the sensors do not project energy on the environment; instead they use energy to reconfigure their parameters to maximize the information gathered from the environment. This dissertation pertains to this latter view of active sensing (Figure 8).

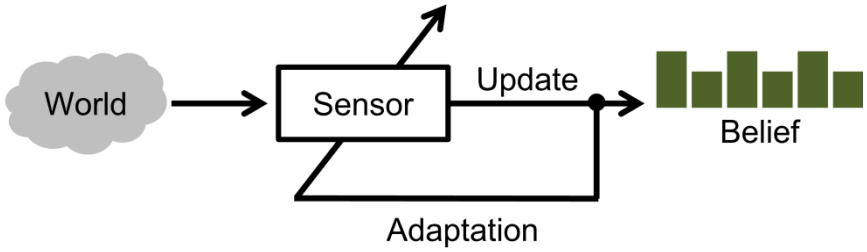

Figure 8 In active sensing, the sensor adapts its operating parameters based on the belief about the world.

Active sensing strategies (as defined by Bajcsy) were inspired from the neurobiological concept of 'active perception', where an organism actively probes the environment to enhance its ability to extract behaviorally relevant information. In his seminal work, Gibson [104] conducted a series of experiments to demonstrate that human perception of object shapes is an active exploratory process. In Gibson's famous cookie-cutter experiments, twenty human subjects were presented with six metallic objects of different shapes in two experimental conditions: (1) a passive condition, where the object was pressed down on the subject's palm with a mechanical lever system, (2) an active condition, where the subject was allowed to move his/her index finger around the edge of the object. In both conditions, the subjects were not allowed to look at the objects. In the passive condition, the identification accuracy was only 49%, as compared to 95% accuracy in the active condition. This result shows that structural properties of objects are best perceived when subjects actively explored the objects. Similar studies in visual perception were later published by Gibson [105].

3.1 History and applications of active sensing

This section presents a brief overview on the history of active sensing which spans more than two decades. It also presents applications of active sensing mainly in three areas: computer vision, robotics, and target tracking.

3.1.1 Active vision

The earliest works on active sensing were in the field of computer vision. In a seminal paper by Aloimonos et al. [106], the authors proved that several computer vision problems that are ill-posed and non-linear with passive observers would become well-posed and linear with an active observer, where an active observer is one that can control the geometrical parameters of its apparatus. For example, the problem of estimating structure from motion is unstable with a passive observer, but with an active observer it reduces to a set of linear equations that can be solved with a closed-from solution. The authors have also shown that the problems of estimating shape from shading, shape from contour, and shape from texture are well-posed with an active observer, which were otherwise ill-posed with a passive observer. A few years after Aloimonos's paper, Ballard et al. [107] developed one of the first practical active vision systems. This system had anthropomorphic features such as binocularity, fovea, and more importantly high speed gaze control. The gaze control mechanisms reduce computational complexity of many computer vision problems because they allow for embedding vision computations into the sensing process. The authors have validated this active vision system on object search and object identification tasks.

Over the next twenty years, combined with more advanced cameras and sophisticated actuators, active vision systems have been applied to solve various problems such as motion tracking [108], scene exploration and reconstruction [109], face recognition [110], vision-based localization and mapping [111], scene segmentation [112], etc. For a more thorough and recent review on active vision, please refer to the paper by Chen et al. [113].

3.1.2 Active sensing in robotics

In robotics, active sensing strategies are commonly used to address problems such as 'where to position the sensors' and 'what to do next'. The latter problem of choosing the next

action involves two sets of decisions: location decisions (where to move the robot to), and sensing decisions (how to reconfigure the sensors). Some researchers distinguish between these two problems and refer to the location decisions as active localization, and sensing decisions as active sensing. However, the difference between the two is rather subtle. In either case, the decision making process involves a trade-off between the immediate rewards of actions (e.g. bringing the robot closer to its goal), and long-term effects (e.g. gathering information through landmarks, surrounding obstacles, or reading signs). A few applications of active sensing in robotics include navigation [114], localization [115, 116], simultaneous localization and mapping [117], off-road driving [118], and robotic exploration [119].

Besides localization and navigations tasks, active sensing is also used in biomorphic robotics. For example, Pearson et al. [120] developed an active whisker system consisting of an array of artificial whiskers mounted on metal-alloy actuators. When these whiskers were subjected to an external physical force, their orientations changed and the changes were measured using optical shaft encoders. The information obtained from the whiskers was used to drive future whisking actions and the motion of the robot. Other examples of active sensing in biomorphic robotic systems include wing flapping mechanisms [121], underwater electro-location systems [122], antenna for contact sensing [123], etc. For a more thorough review on active sensing in robotics, please refer to the paper by Mihaylova et al. [124].

3.1.3 Target tracking

Apart from computer vision and robotics, active sensing has also been widely used in military applications, specifically target tracking. The target tracking problem involves estimating locations and velocities of multiple moving targets (for e.g. ground vehicles) using surveillance sensors such as radars, sonars, or electro-optical sensors. One of the central challenges in target tracking problem is selecting the next sensing action, where an action

involves choosing sensors and setting their configurations (such as pointing angles, dwell lengths, etc.). Each action has a certain cost associated with it such as the time or the energy required to reconfigure the sensor. The sensing actions are selected to minimize uncertainty in the estimates of target locations and velocities. In some target tracking applications, the sensors not only have to track the targets, but also acquire information sufficient to identify (or classify) the targets.

Kreucher et al. [125] presented an active sensing approach for tracking positions and velocities of multiple vehicles using radars working in moving target indication (MTI) mode. In this method, the controlling agent maintains a joint multi-target probability distribution. The distribution models the number of targets in the surveillance region, their estimated positions, and their velocities. The distribution is updated at each time step based on the measurements acquired from the radars. The radars are reconfigured at every time step to minimize the expected entropy (measured as Renyi divergence) of the joint probability distribution. Chung et al. [126] have presented a decentralized active sensing algorithm to track dynamic targets with a network of mobile sensors. In this algorithm, each sensor maintains local estimates on the positions of different targets and the corresponding error covariance matrices. The sensors transmit their estimates to the neighboring sensors in the network, and likewise receive estimates from other sensors. Each sensor then fuses the information to estimate a global error covariance matrix. While choosing the next action (which involves moving to a different location); each sensor estimates future global error covariance by incorporating expected changes in its local error covariance (assuming the other sensors remain stationary), and selects the action that would minimize the expected global covariance. Though sub-optimal, the authors have demonstrated on simulated scenarios that their distributed active sensing approach can obtain close-to-optimal

solutions. Other applications of active sensing in target tracking include the works of Yang et al. [127, 128], and Mukai et al. [129].

3.2 Mathematical formulation of active sensing

There are three main decision-making frameworks used to formulate active sensing problems: Markov decision processes, partially observable Markov decision processes, and multi-arm bandit processes. A fourth method used for active sensing, though inherently not a decision-making method, is recursive state estimation. These methods allow the sensing agents to address the two main issues in active sensing: (1) how to interpret the sensor responses, and (2) how to select an optimized sensing configuration for the next step? In all these methods, stochastic modeling and analysis play a very important and central role. This is because, most real-world sensing systems operate in noisy environments and their observations cannot always be interpreted with high certainty. Consequently, any useful theory and practice of active sensing must be based on probabilistic frameworks [2].

3.2.1 Markov decision processes

Markov decision processes (MDPs) are mathematical frameworks used for formulating decision-making problems in environments that are completely observable but non-deterministic. In a completely observable environment, the agent can directly observe the state of the world. An MDP is formally defined as a 4-tuple (S, A, T, C), where S is a finite set of states, A is a finite set of actions, T is a transition probability function, and C is a cost function. The transition probability function T provides probabilities of transitioning from state s to s' upon taking action a: $T(s'|s,a) = p(s_{t+1} = s'|s_t = s, a_t = a)$. The cost function C provides the costs of executing an action a at a state s: $C(s,a)$. A defining property of an MDP is that the state transitions

follow the Markov property: the future state is independent of the past states if current state is known.

When an active sensing problem is formulated as an MDP, the states represent the sensor's understanding of the world, actions correspond to adjusting the operating configurations of the sensor, and cost function represents a combination of the sensing costs and information gains. A solution to an MDP is called as a policy ($\pi^*: S \rightarrow A$): it provides a mapping from states to actions such that the expected cost is minimized. In an active sensing scenario, the sensing agent can use this policy as a lookup table and determine how to reconfigure the sensor, because the agent is completely aware of the current state of the environment.

MDPs can be solved (i.e. an optimal policy can be found) with either linear programming or dynamic programming methods, though the latter is more widely used. The two most commonly used dynamic programming algorithms are policy iteration algorithm, and value iteration algorithm [130]. A policy iteration algorithm begins with a randomly initialized policy π' and refines it repeatedly until it converges to the optimal policy. At each iteration, the algorithm goes through two steps: value determination step, and improvement step. In the value determination step, the algorithm computes the estimated cost of the current policy π' with respect to each state s:

$$E_{\pi'}(s) = C\big(s, \pi'(s)\big) + \gamma \sum_{s' \in S} T\big(s' \big| s, \pi'(s)\big) E_{\pi'}(s') \tag{14}$$

In eq. (14), the term γ is called discounting factor. This parameter determines the present value of future costs: a cost incurred t steps in the future is γ^t times what it would cost if it were incurred immediately. Note that eq. (14) is actually a system of $|S|$ linear equations with $|S|$ unknowns, one for each state s. These equations can be solved using various methods such as

Gauss-Jordan elimination, Cramer's rule, or matrix inversion. In the improvement step, the current policy π' is refined such that it has a smaller expected cost. Namely, for each state, if there is an action a that would incur a smaller expected cost $C(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) E_{\pi'}(s')$ than the current expected cost $C\big(s,\pi'(s)\big) + \gamma \sum_{s' \in S} T\big(s'\big|s,\pi'(s)\big) E_{\pi'}(s')$, then the policy is updated as $\pi'(s) = a$. The policy iteration algorithm terminates when the solution converges, i.e. there is no change in the policy from the previous iteration. Bellman's equations [131] show that the policy iteration algorithm always converges in a finite number of iterations, and the policy obtained at the final iteration is optimal.

Value Iteration algorithm beings with randomly initialized expected costs for each state $E_0(s)$. The values are refined iteratively by selecting actions that minimize the expected costs. Namely, at each step k, the algorithm computes the expected cost $E_k(s)$ using the previous estimate $E_{k-1}(s)$ as:

$$\begin{aligned} E_k(s|a) &= C(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) E_{k-1}(s') \\ E_k(s) &= \min_{a \in A} E_k(s|a) \end{aligned} \tag{15}$$

The policy is then defined as the action that generates the minimum expected cost $\pi(s) = \text{argmin}_{a \in A} E_k(s|a)$. The value iteration algorithm terminates when the reductions in expected costs converge: $\max_{s \in S} |E_n(s) - E_{n-1}(s)| < \epsilon$. This stopping criterion guarantees that the returned policy is an ϵ-optimal, i.e., it has an expected cost that does not differ from the optimum by more than ϵ, where ϵ is typically chosen to be a very small number.

3.2.2 Partially observable Markov decision processes

In most real world scenarios, it is unlikely to find a completely observable environment due to the unreliability of sensors. Partially Observable Markov Decision Processes (POMDPs)

[132] are a generalization of MDPs for decision making in such environments. These environments are called 'partially observable' because at a given point the agent does not know the exact state of the environment. The differences between MDPs and POMDPs are very neatly summarized in the following statements borrowed from Monahan [133]: '*Howard* [134] *described a transition in MDP as a frog in a pond jumping from lily pad to lily pad. In a POMDP environment, the lily pond is covered with mist, therefore the frog is no longer certain about the pad it is currently on. Before jumping, the frog can observe information about its current location.'*

A POMDP is formally defined as a 7-tuple $(S, A, O, b_0, T, \Omega, C)$, where S, A, and O are the finite set of states, actions, and observations respectively, b_0 is the initial belief across all states, $T(s'|s,a)$ is the probability of transitioning from state s to state s' given action a, $\Omega(o|s,a)$ is the probability of making observation o at state s upon taking action a, and $C(s,a)$ is the cost of executing action a at state s. Since the agent is unaware of the current state of the environment, it maintains a belief distribution b_t, which assigns a probability to each one of the possible states. Since the state transitions are Markovian, the agent can update the belief distribution based on the previous belief b_{t-1}, the action taken a_t, and the resulting observation o_t as:

$$b_t(s') = \eta\Omega(o|s',a)\sum_{s\in S} T(s'|s,a)b_{t-1}(s) \tag{16}$$

In eq. (16), η is a normalization constant that ensures the belief distribution sums to unity, i.e. $\sum_{s\in S} b_t(s) = 1$.

As with MDPs, a solution to a POMDP is also called a policy π. However, unlike an MDP's policy which provides a mapping from states to actions, a POMDP's policy provides a mapping from a belief distribution b_t to an action, i.e. $\pi: b_t \rightarrow A$, such that the expected cost is

minimized. Note that though S is a finite set, the 'belief-space' is continuous and infinite because belief is a probability distribution. Given the initial belief distribution b_0, the expected cost of a policy π is defined as:

$$J^{\pi}(b_0) = \sum_{t=0}^{\infty} \gamma^t \, c\big(b_t, \pi(b_t)\big) \tag{17}$$

where γ is the discounting factor, and c is an expected cost model derived from the original cost function C as $c(b,a) = \sum_{s \in S} b_t(s) C(s,a)$. An optimal policy for a POMDP (π^*) is the one with minimum expected cost:

$$\pi^* = \underset{\pi}{\operatorname{argmin}} J^{\pi}(b_0) \tag{18}$$

Finding the optimal policy π^* is a very challenging problem: finite-horizon POMDPs are PSPACE-complete [135], and infinite-horizon POMDPs are undecidable [136]. The currently known algorithms that return optimal policies for a POMDP can only be applied to problems of small size. However, there are a number of algorithms [137-139] that compute approximate solutions to POMDPs in polynomial time, and are therefore more practical.

Castanon [140] presented one of the earliest applications of POMDPs for active sensing. In this paper, the authors used POMDPs to schedule three radars for classification of objects into one of three known classes. The radars work in two modes: (1) an inexpensive mode that measures only the radial dimensions of the objects, and (2) an expensive mode that images the object. Their results show that POMDPs generate near-optimal sensing strategies for the classification problem. Krishnamurthy [141] used POMDPs to schedule a set of noisy sensors for aircraft identification problem. Ji et al. have used POMPDs for cost-sensitive feature acquisition and classification [142] and multi-aspect sensing [143]. More recently, Miller et al. [144] have

presented a POMDP based framework for coordinated guidance of autonomous UAVs for target tracking.

3.2.3 Multi-armed bandit processes

The third decision making framework commonly used for active sensing is the multi-armed bandit (MAB) process. The name 'multi-armed bandit' comes from a scenario in which a gambler faces several slot machines (one-armed bandits), that look identical at first but produce different expected winnings. The crucial issue here is the trade-off between acquiring new information (exploration) and capitalizing on the information available so far (exploitation). A single-armed bandit process is characterized by the two random sequences $\{X(0), X(1), \dots,\}, \{R(X(0)), R(X(1)), \dots,\}$ where $X(n)$ denotes the state of the machine after it has been operated n times, and $R(X(n))$ is the reward obtained when the machine is operated for the n^{th} time. When the machine is operated for the n^{th} time its state changes according to $X(n) = f_{n-1}(X(0), \dots, X(n-1), W(n-1))$, where $f_{n-1}(\cdot)$ is given and $\{W(n); n = 0,1, \dots\}$ is a sequence of independent real-valued random variables that are independent of $X(0)$ and have known statistical description[4]. A multi-arm bandit process is a collection of k single arm bandit processes. The i^{th} process in a MAB process is characterized by two sequences $\{X_i(N_i(t)), R_i(X_i(N_i(t))); N_i(t) = 0,1,2, \dots, t; t = 0,1,2, \dots\}$, where $N_i(t)$ denotes the number of times i^{th} arm has been operated until time t. The agent controlling a MAB process can operate only one arm at a given time. The MAB problem involves identifying a scheduling policy $\pi = \{a_1, a_2, \dots, a_t\}; 1 \leq a_i \leq k$ which maximizes the expected reward over a horizon of time. This problem was originally formulated in the 1940s and was typically solved using dynamic programming methods. However, in 1979 Gittins and Jones [145] showed that an

[4] The main difference between a single-arm bandit process and a MDP is that the state transitions in the former do not have to be Markovian.

optimal solution to the MAB problem can be obtained using an 'index type' method. In their method each bandit process was assigned a dynamic allocation index which depends only on that process. At a given time, the agent can select the bandit process with the highest index thus finding an optimal scheduling policy. This method provides a polynomial time solution to a problem which was then otherwise deemed exponential.

Though not used as widely as POMDPs, there have been a few applications of MAB processes for active sensing. Krishnamurty et al. [146] have used MAB processes to derive beam scheduling algorithms for electronically scanned array tracking systems. Song et al. [147] have also used MAB processes for scheduling multiple sensors to search a discretized grid-like environment to locate stationary objects. Keqin et al. [148] have approached the problem of sequential channel probing as a restless MAB process (a generalization of the MAB process). Their problem deals with an opportunistic communication system consisting of multiple independent channels, where the user can only access and sense with a subset of channels for a given time slot. More recently, Pavlidis et al. [149] presented a review on the prospects of MAB processes in active sensing.

3.2.4 Recursive state estimation

The fourth mathematical framework used for active sensing is recursive state estimation. Recursive state estimation provides a framework to estimate quantities that are not directly observable but can be inferred from sensor responses. It is widely used in robotics to estimate the state of the robot or its environment from sensor measurements. In active sensing, it provides a framework for the agents to interpret sensor responses. The most generic algorithm used for recursive state estimation is known as a Bayes filter, which is only applicable in discrete domains. For continuous domains, there are two families of filters used for recursive estimation:

(1) Gaussian filters (e.g. Kalman filter, Extended Kalman filter, Unscented Kalman filter), and (2) Non-parametric filters (Particle filter).

Unlike MDPs, POMDPs, or MABs, recursive state estimation is inherently not a decision making framework. However, it can be combined with a utility function or a decision criterion for selecting optimal sensor configurations. For example, Zhao et al. [150] used entropy and Mahalanobis distance as objective functions combined with a particle filter for collaborative tracking of targets in ad hoc sensor networks. Chhetri et al. [151] proposed a combination of particle filters and extended Kalman filters for scheduling multiple sensors in target tracking problems. More recently, Ryan et al. [152] have used a particle filter combined with a conditional entropy measure to track targets with a camera mounted on an unmanned air vehicle.

3.3 Related work in machine learning

This section reviews two problems from machine learning that are very similar to active sensing from a computational perspective: dynamic feature acquisition and adaptive sampling.

3.3.1 Dynamic feature acquisition

In machine learning literature, there is a problem referred to with different names such as dynamic feature acquisition, test-time feature acquisition, or cost-sensitive feature acquisition. In this problem, the classification agent is provided with a pre-trained classifier, and a test sample with missing features. The agent can obtain any of the missing features at predefined acquisition costs. The target of the agent is to determine an optimal sequence of features to acquire, and also determine when to stop acquiring features and classify the test sample. Though more relevant to machine learning research, this problem is very similar to active sensing, especially from a computation standpoint.

There is a handful of research papers published over the last decade that specifically addresses the dynamic feature acquisition problem [142, 153-156]. These papers, when summarized from a broader viewpoint, follow a similar theme. They formulate feature acquisition as a decision-theoretic process (typically a POMDP), where the states correspond to the class labels, actions correspond to acquiring features, observations correspond to the feature values, and the beliefs are the class-posteriors. The features are sequentially acquired to balance between acquisition costs and expected misclassification rates (or expected information gains). The main challenge for this problem lies in how to estimate the expected information gains.

3.3.2 Adaptive sampling

The adaptive sampling problem has its origins in statistics, signal processing, and machine learning, but form a computational perspective is closely related to active sensing. To better define this problem, we borrow an example from [2]. Consider the task of constructing a topographic map of a large unknown area (e.g. battlefield) with an airborne scanning laser. The laser can measure the distance between itself and an observed object. Due to safety concerns and fuel costs, one would like to construct the topographic map with minimal amount of flight-time. This problem can be addressed with a passive sampling approach, where the entire area is scanned at high resolution and the acquired measurements used to construct the map. However, if some regions of the area were flat (they did not have any ridges or peaks), then it would be sufficient to scan these flat regions at a low resolution. In such a scenario, it would be optimal to use an adaptive sampling approach, where the scanner is adaptively scheduled to scan the flatter regions at lower resolution and the other regions at a high resolution. This will minimize the overall scanning time and therefore the total flight-time.

Adaptive sampling has been widely used in imaging applications and sensor networks. Nayar et al. [157] have constructed an adaptive dynamic range camera, which consists of a color

CCD detector and a controllable liquid crystal light modulator. The light modulator allows the camera to adapt the light exposure levels of each pixel and therefore increases the dynamic range of the captured images. Other examples of such adaptive imaging systems include the works of Mannami et al. [158], Christensen et al. [159], and Kinast et al. [160]. In sensor networks, adaptive sampling is typically used to minimize the power consumption for periodic data collection; at each time step, a different subset of sensors is chosen to make measurements such that the quality of the collected data would not deteriorate [161-163]. The measurements of the non-sampling sensors are predicted using a combination of probabilistic sensor models and the measurements of the sampling sensors.

Castro et al. [164] have conducted a series of psychological experiments to understand if humans inherently use adaptive sampling. In this study, the human subjects were presented with a task of estimating the decision boundary of a binary classification problem in a discretized one-dimensional space. For each subject, a random decision boundary $d \in [0\ 1]$ was chosen such that all samples in the range $[0, d]$ were of class 0, and the samples in the range $[d, 1]$ belonged to class 1. To determine the boundary d, the subject was allowed to query the class labels of any samples in the range [0, 1]. The results showed that the subjects were able to select queries that allowed them to arrive at the decision boundaries faster, almost like a binary search.

3.4 Active chemical sensing

We conclude this chapter with a review of previous works that have applied active or adaptive sensing strategies with chemical sensors. These works are organized based on the sensing problem they address, which include chemical classification, odor generation, drift compensation, and data collection.

Chemical classification: To our knowledge, there have only been two works that employ active sensing strategies for classification of chemicals: Priebe et al. [165] and Dinakarababu et

al. [166]. Priebe's methodology (termed integrated sensing and decision trees (ISPDT)) is based on the concept of integrated sensing and processing [167]. Given a feature vector, the method builds a decision tree that partitions feature space hierarchically; nodes close to the root of the tree select features based on their ability to provide good clustering of examples regardless of class labels, whereas nodes at the leaves select sensors based on their ability to discriminate examples from different classes. The authors evaluated the model on an experimental dataset from an array of 19 optical sensors exposed to trichloroethylene (a carcinogenic industrial solvent) in complex backgrounds. Their results show that ISPDT can reduce misclassification rates by 50%, while requiring only 20% of all sensors to make any individual classification. Compared to the work presented in this dissertation, the ISPDT methodology has two main shortcomings: (1) it cannot handle the dynamics in sensor responses, and (2) it does not account for the data acquisition costs. Also, ISPDT is more of a methodology rather than an algorithm, and therefore leaves open many aspects of the methodology such as partition criteria and classifier selection. In addition, methods based on decision trees are highly sensitive to noise in training data, and tend to perform poorly if there is strong interaction among the features [168].

Dinakarababu et al. [166] proposed an adaptive spectroscopic architecture called adaptive feature specific spectrometer (AFSS). Unlike a traditional IR spectrometer, AFSS has a tunable spectral filter (a digital micro-mirror device), which aids in multiplexing certain spectral bands and directing them onto a photo-detector. The system thus measures the projection of the incoming spectral density onto a set of basis vectors, rather than measuring the spectral density directly. The basis vectors are changed over time based on the information obtained from previous measurements. In contrast to the computational methods proposed in this dissertation, AFSS is a hardware-based approach for active chemical sensing.

Odor genration: Nakamoto et al. [169, 170] have proposed an active method for odor generation, where the goal was to reproduce an odor blend by creating a mixture from its individual components. To our knowledge, this work is the earliest application of active sensing in the domain of chemical sensors. 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. Unlike the goal of this dissertation, which is to determine the identity and/or concentration of volatile chemicals, Nakamoto's work was aimed at generating odors.

Drift compensation: Wenzel et al. [171] have used recursive state estimation theory for online compensation of baseline drift in chemical sensors. The authors modeled the response of the chemical sensors y_t as a linear function of concentration of the absorbed analyte C_t, the current baseline drift λ_t, and a process noise ω_t as: $y_t = f(C_t, \lambda_t, \omega_t)$. The concentration C_t of the sorbed analyte at time t is modeled as linear function of the absorbed concentration C_{t-1} at time $t-1$: $C_{t+1} = g(C_t)$. When the sensor is exposed to an unknown analyte, the responses are used to predict the analyte's concentration and the sensor's baseline drift using a Kalman filter. The estimated baseline drift is then subtracted from the sensor's output. The authors validated this technique on experimental data collected on shear horizontal surface acoustic wave (SH-SAW) sensors exposed to organophosphate pesticides.

Data collection: Lomasky et al. [172] have presented an active class selection method for the data collection with chemical sensors. Active class selection addresses the question: if one can collect n additional training instances, how should they be distributed with respect to the chemical classes? To this end, the authors proposed an approach where the training instances are generated in proportion to the instability of class boundaries. The instability is measured in terms of the number of instances whose classification labels change upon inclusion of new training instances. The authors have validated this approach on an experimental dataset from an array of

fluorescent microbead sensors exposed to six different organic chemicals and their mixtures. The results show that active class selection can minimize the number of new training instances needed to obtain the maximal classification performance. Active sensing has also been investigated for data collection in hyperspectral imaging [173]; a domain where acquiring the complete data cube is neither required nor desirable in most applications. In these cases, adaptive techniques may be used in library-based spectral identification tasks to match a partial chemical spectrum to a known spectrum or to a mixture of spectra.

4. ACTIVE CLASSIFICATION OF CHEMICALS

In chemical sensing, the classification problem involves determining the identity of a given sample from a list of known possible targets. It is one of the most widely studied computational problems in chemosensing, and has gained a lot of importance with the growing interest in machine olfaction and electronic nose research. To address this problem, researchers have experimented with a wide array of statistical techniques such as *k*-nearest neighbor classifier, support vector machines, quadratic classifiers, decision trees, and neural network-based methods like multi-layer perceptron, radial basis functions, probabilistic neural networks, etc. For a more thorough review on pattern recognition methods for chemical sensing, please refer to reviews by Gutierrez-Osuna [174], and Marco and Gutierrez-Galvez. [175].

The classification problem with a single chemical sensor could be formally stated as follows: given a chemical sensor that operates at D different configurations: $\boldsymbol{\rho} = \langle \rho_1, \rho_2, \dots, \rho_D \rangle$, and a gas sample, determine the identity of the sample from one of M known classes: $\boldsymbol{\omega} = \langle \omega_1, \omega_2, \dots, \omega_M \rangle$. The traditional approach to this problem is to expose the chemical sensor to the gas sample, and collect the sensor's response at all D configurations to create a feature vector $\langle o_1, o_2, \dots, o_D \rangle$. The feature vector is then passed through a pre-trained classifier to predict the identity of the sample.

Though straightforward, this form of sensing is unlikely to be cost-effective because only a subset of the features is typically sufficient to classify the given sample, thus rendering a significant portion of the acquired information expendable. This issue can be addressed with passive methods such as feature subset selection; however, we seek to develop an active sensing method for classification of chemicals. The method will optimize the operating configurations of the sensor on-the-fly, based on evidence accumulated from previous measurements. Once

sufficient evidence is acquired, the sensing process can be halted to reduce future costs of data acquisition and processing.

4.1 Methods

To develop an active sensing method, the classification problem needs to be cast as a decision-making problem. For this purpose, we follow Ji and Carin's [142] work on cost-sensitive feature acquisition for classification, where the authors have formulated active sensing as a POMDP. In this formulation, each class is represented as a set of interconnected states in a POMDP. When sensing actions are executed and corresponding observations obtained, the belief distribution over these states changes based on a set of pre-trained sensors models.

The method[5] consists of three steps as demonstrated in Figure 9. The first step involves training class-dependent probabilistic sensor models using Input-Output Hidden Markov Models (IOHMMs). These models create probabilistic representation of the sensor's responses to different chemicals when driven with a sequence of operating configurations. The probabilistic models are necessary for the POMDP to interpret and predict the sensor responses. In the second step, the IOHMM parameters (states, transition probabilities, and observation probabilities) are mapped into a POMDP, which casts the classification problem as a decision-theoretic process. The third step comes into play during the sensing stage, i.e. when the sensor is exposed to an unknown analyte. In this step, a myopic approach is used to select the next operating configuration for the sensor such that the expected risk of misclassifying the sample is minimized. The myopic algorithm continues until a predefined stopping criterion is met. A detailed description of these three steps is presented in following sections.

[5] We will refer to this method as AC, an abbreviation for active classification.

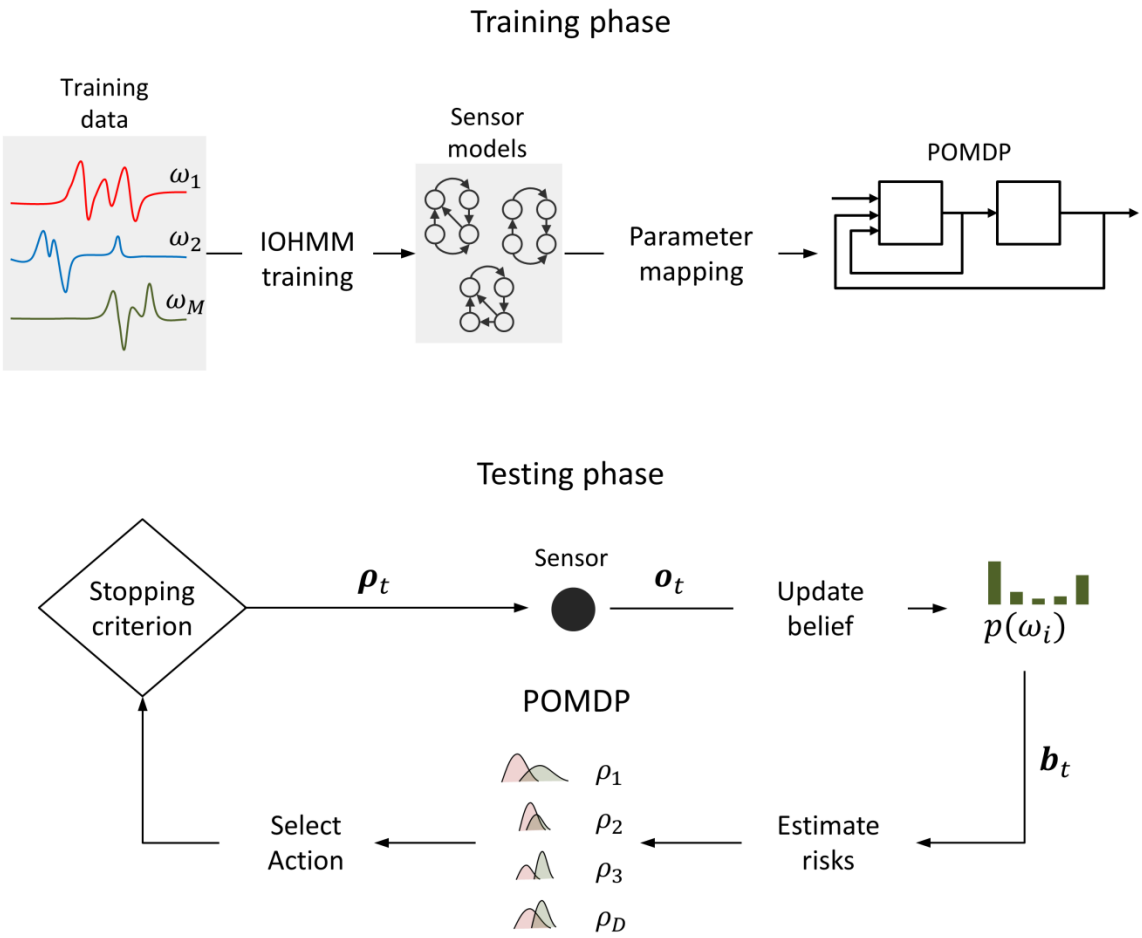

Figure 9 Overview of the active classification method. During the training phase, a separate IOHMM is trained for each class. Then, using the IOHMM parameters, the M-class classification problem is formulated as a POMDP. During the testing phase, the POMDP selects the sensing action with the least expected Bayesian risk at each step. The sensor is operated accordingly and the belief distribution updated based on the sensor's response. The sensing process continues until a stopping criterion is met.

4.1.1 Sensor modeling

The steady-state response of the chemical sensor operating at configuration ρ_i when exposed to chemical ω_j is modeled as a Gaussian mixture [176]:

$$p(x_{i,j}|\rho_i, \omega_j) = \sum_{k=1}^{K} \alpha_{i,j,k}\, \mathcal{N}(x_{i,j}|\mu_{i,j,k}, \sigma_{i,j,k}) \tag{19}$$

where $x_{i,j}$ is the sensor's response, K is the number of mixture components, $\alpha_{i,j,k}$, $\mu_{i,j,k}$, and $\sigma_{i,j,k}$ are k^{th} mixing coefficient, mean, and variance respectively. In this model, the sensor configurations ρ_i and the classes ω_j are discrete variables but the sensor's response $x_{i,j}$ is a continuous variable. For notational convenience, all components in this GMM are represented as $G_{(i,j)}$. Given training data, we use expectation maximization algorithm [177] to learn the model parameters such that the likelihood of the training data is maximized. GMMs are suited for developing these sensor models because they are proven universal approximators for continuous distributions: given a sufficient number of mixture components, GMMs can model any continuous distribution [178].

Though GMMs can capture the steady-state responses of the sensors they cannot capture the dynamics introduced due to changes in the sensor configurations. To model the sensor dynamics, we use IOHMMs [179]. IOHMMs are a generalization of the traditional Hidden Markov Model (HMMs) [180]. HMMs and IOHMMs belong to a class of probabilistic models known as DBNs [181]. DBNs model the temporal relations between a given set of random variables. In an HMM, the future state of the system is assumed independent of the past states given the present state, whereas, in an IOHMM the future state of the system not only depends on the current state but also on the future input to the system. Figure 10 compares the temporal architectures of a discrete-state HMM and a discrete-state IOHMMs. This figure shows three time slices of an HMM and an IOHMM, where the rectangular nodes represent discrete variables and circular nodes represent continuous variables. Likewise, the output of an HMM depends only on the current state of the system, but the output of an IOHMM depends on both the current

state and the current input to the system. Thus IOHMMs create a dynamic mapping between an input data stream and an output data stream. In the current context, the input data stream corresponds to the operating configurations of the chemical sensor and the output stream corresponds to the sensor responses.

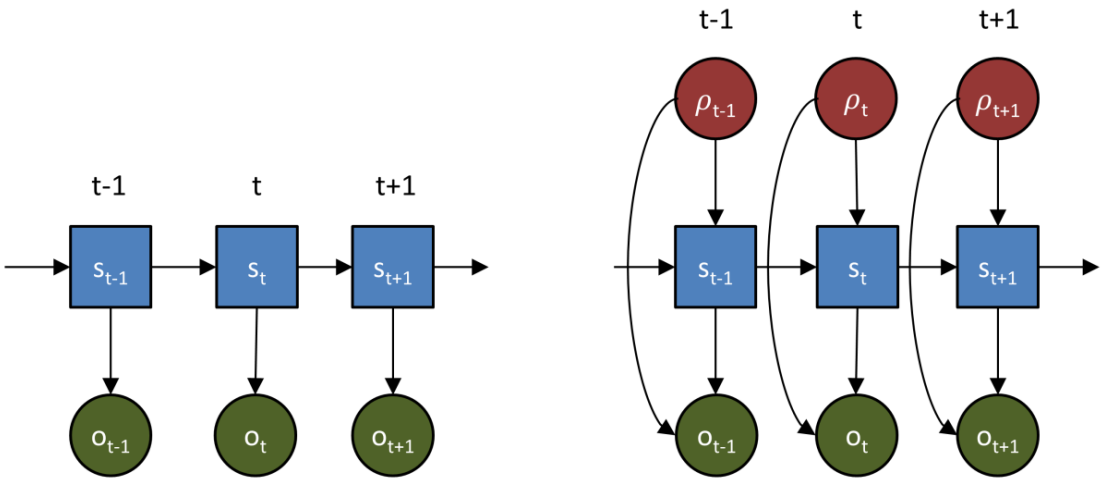

Figure 10 Temporal architectures of HMMs and IOHMMs.

Formally, an IOHMM is defined as a 6–tuple $(H, I, O, \psi, \tau, \varphi)$ where: H is a finite set of states, I is a finite set of discrete inputs, O is a set of observations, ψ is the initial state distribution, $\tau(h'|h, i)$ is the state transition function, which describes the probability of transitioning from state h to state h' upon getting an input i, and $\varphi(o|h, i)$ is the observation probability function, which describes the probability of making observation o at state h upon getting an input i.

We model the response of the sensor to each chemical as a separate IOHMM. To do this, we first train D GMMs, one for each sensor configuration as described in section 4.1.1. We assume that each mixture component in the GMMs acts as a state in the IOHMM and is therefore

hidden. Thus, the IOHMM for class ω_j has $D \times K$ different states, i.e. $H_{(j)} = G_{(1,j)} \cup G_{(2,j)} \cup \ldots \cup G_{(D,j)}$. I has D entries corresponding to the D operating configurations of the sensor, i.e. $I = \boldsymbol{\rho}$. The observation probability function φ is directly obtained from the GMMs. However, to learn the transition probabilities τ, we use expectation-maximization algorithm on the training data. The training data for IOHMMs is obtained by driving the chemical sensor with random sequences of operating parameters when in the presence of the chemical, and recording the corresponding responses. During the testing phase, the sensor will be driven with a sequence that is most likely not part of the training set. Thus, it is important that the training set captures as much of the variance introduced due to the dynamic changes in sensor configurations, and for this reason randomized sequences are a better choice than fixed sequences.

For chemical sensors that have larger time constants (e.g. a MOX chemical sensor), the current sensor response is heavily influenced by previous configurations of the sensors. For such sensors, learning the transition probabilities is essential to capture the sensor mechanics completely. For sensors with shorter time constants (e.g. an FPI sensor), the transition probabilities could be approximated to a 0-1 model: $\tau(h'|h,i) = \begin{cases} 1, h = h' \\ 0, h \neq h' \end{cases}$.

Once trained, this family of IOHMMs can be used for classification purposes. Given an input sequence $(\rho_1, \rho_2, \ldots, \rho_t)$ and the corresponding observation sequence $(o_1, o_2, \ldots o_t)$, we can estimate the class posteriors as:

$$p(\omega_i|o_1, o_2, \ldots, o_t, \rho_1, \rho_2, \ldots, \rho_t) = \frac{p(o_1, o_2, \ldots, o_t|\rho_1, \rho_2, \ldots, \rho_t)p(\omega_i)}{\sum_k p(o_1, o_2, \ldots, o_t|\rho_1, \rho_2, \ldots, \rho_t)p(\omega_k)} \quad (20)$$

then classify the observation sequence based on maximum a posteriori (MAP) condition. However, to determine an optimal sequence of actions, we need to formulate the classification problem as a POMDP.

4.1.2 Parameter mapping

As defined in section 3.2.2, a POMDP is a seven-tuple $(S, A, O, b_0, T, \Omega, C)$ and they can be obtained from the IOHMM parameters as follows. The state space S of the POMDP is the union of the states of the M IOHMMs: $S = H_{(1)} \cup H_{(2)} \cup \ldots \cup H_{(M)}$, where $H_{(u)}$ is the set of states of the IOHMM corresponding to class ω_u. The initial belief for each state is defined as $b_0(s) = p(\omega_u)\psi_{(u)}(s)$, for $s \in H_{(u)}$. The action space A is a union of sensing and classification actions: $= A_\rho \cup A_\omega$ $(|A| = D + M)$. $A_\rho = (\rho_1, \rho_2, \ldots, \rho_D)$ is a set of sensing actions containing D items, where each item corresponds to operating the sensor at one of the D configurations. Similarly $A_\omega = (\omega_1, \omega_2, \ldots, \omega_M)$ is the set of classification actions containing M items, where each item corresponds to classifying the sample as one of the M classes and terminating the sensing process.

The state transition probabilities are defined such that transitions are restricted to states within a single class. This restriction is driven by the assumption that the sensing actions cannot change the underlying identity of the test sample. Thus T can be derived from the IOHMMs as:

$$T(s'|s,a) = \begin{cases} \tau_{(u)}(s'|s,a), & s, s' \in H_{(u)} \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

The observation probabilities are also obtained directly from the IOHMMs: $\Omega(o|s,a) = \varphi_{(u)}(o|s,a)$, for $s \in H_{(u)}$. The cost function is considered independent of the state and only dependent on the actions, i.e. $C(s,a) = C(a)$. The cost of a sensing action $C(\rho_i) = c_i$ reflects the amount of energy, time, or power used to reconfigure the sensor. For example, in a MOX sensor the sensing costs correspond to the energy required to heat the sensor to a particular temperature. The cost of classification actions $C(\omega_u, \omega_v) = c_{uv}$ represent the price of misclassifying a sample. For example, consider a classification problem of distinguishing

between a toxic chemical ω_a form another non-toxic chemical ω_b. Then the cost of misclassifying ω_a as ω_b is much higher than its counterpart.

The POMDP maintains a belief distribution $b_t(s)$ across all states. The belief distribution assigns a probability to each state and is defined as follows:

$$b_t(s) = p(s|o_1, o_2, \dots, o_t, \rho_1, \rho_2, \dots, \rho_t) = p(s|o_t, \rho_t, b_{t-1})$$
$$\sum_{\forall s \in S} b_t(s) = 1 \tag{22}$$

where $(o_1, o_2, \dots, o_t)$ and $(\rho_1, \rho_2, \dots, \rho_t)$ is the history of observations and sensing actions respectively. The above equation reflects the assumption that $b_t(s)$ is a sufficient statistic and it is updated incrementally by incorporating the latest action a_t and observation o_t:

$$b_t(s) = \frac{p(o_t|s, \rho_t) \sum_{s'} p(s|\rho_t, s') b_{t-1}(s)}{p(o_t|\rho_t, b_{t-1})} \tag{23}$$

In eq. (23), the denominator is a normalization term, which ensures that $b_t(s)$ sums to 1.

4.1.3 Solving POMDP

Once the POMDP has been formulated as described above, the next step is to solve the POMDP to find an optimal policy. As discussed in section 3.2.2, this problem is computationally very expensive and is therefore only restricted to POMDPs of small size (the number of states). However, we could use approximation algorithms to solve the POMDP. Though the idea of using an approximate solution is appealing, it comes with the following issues. First, an approximate POMDP policy would allow for repeated actions, i.e. operating the sensor at the same configuration multiple times. This is not a desired outcome in the current classification scenario because we do not want to operate the sensor at the same configuration multiple times, since it would not provide any extra information towards classification. Second, in a

classification problem, one wishes to use only a few actions to classify the sample. Therefore, employing a non-myopic approach (one that optimizes over a long horizon) is not the appropriate choice. Therefore, we use a myopic approach to solve the POMPDs. In a myopic approach, the objective is to optimize on a per-decision (action) basis.

In our myopic approach, a sensing action is taken only when the cost of sensing (c_i) is lower than the expected reduction in Bayes risk. The expected risk of a sensing action ρ_i can be estimated as:

$$R_s(\rho_i, b_t) = \sum_{\forall o} \min_u \left(\sum_v c_{uv} \sum_{s' \in H_{(v)}} \sum_s p(o|s', \rho_i) p(s'|s, \rho_i) b_t(s) \right) \tag{24}$$

which averages the minimum Bayes risk over all observations that may result from a sensing action. Similarly, given belief state b_t, the expected risk of a classification can be computed as:

$$R_c(b_t) = \min_u \sum_v c_{uv} \sum_{s \in H_{(v)}} b_t(s) \tag{25}$$

Using equations 24 and 25, the utility of a sensing action ρ_i is computed as:

$$U(\rho_i, b_t) = [R_c(b_t) - R_s(\rho_i, b_t)] - c_i \tag{26}$$

At a given time step, the myopic algorithm selects the sensing action that has the maximum utility. The algorithm tracks all sensing actions that have been used before to avoid repetition. If the utilities of all actions are negative, i.e. the sensing costs exceed the expected reductions in Bayes risk, then a classification action is executed. The final class label of the sample ω_{out} is determined as:

$$\omega_{out} = \underset{\omega_u \in \boldsymbol{\omega}}{\operatorname{argmin}} \left(\sum_{\omega_v \in \boldsymbol{\omega}} c_{uv} \sum_{s \in H_{(v)}} b_t(s) \right) \tag{27}$$

The myopic approach is summarized in Table 2.

Table 2 Pseudocode for active classification.

Step 1: Initialization	• Initialize beliefs: $b_0(s) = p(\omega_u)\psi_{(u)}(s)$, for $s \in H_{(u)}$ • Initialize available sensing actions $A_\rho = \{\rho_1, \rho_2, \dots, \rho_D\}$ • $t = 1$
Step 2: Select action	• Estimate utilities of all available sensing actions o $U(\rho_i, b_t) = [R_c(b_t) - R_s(\rho_i, b_t)] - c_i$ • Select best action o $a_t = \operatorname{argmax}_{\rho_i \in A_\rho} U(\rho_i, b_t)$
Step 3: Sensing	• If $U(a_t) > 0$ o $U(\rho_i, b_t) = [R_c(b_t) - R_s(\rho_i, b_t)] - c_i$ o Operate sensor at a_t to obtain observation o_t o Remove a_t from A_ρ: $A_\rho = A_\rho - a_t$ • else o Classify the sample using eq. (27).
Step 4: Update belief	• Update belief distribution • $t = t + 1$
Step 5: Termination	• If $t < D$ o Go to step 2 • else o Classify the sample using eq. (27).

4.1.4 Discretizing observation spaces

The equation used for calculating expected risk (eq. (24)) is only applicable for discrete observational spaces. For continuous spaces, the summation in eq. (24) becomes an indefinite integral. To address this issue, during the training stage, we discretize the observation space for each operating configuration using *k*-means clustering. The clustering results in a finite set of (sorted) cluster centers $\{-\infty, o_{\rho_i,1}, o_{\rho_i,2}, \dots, o_{\rho_i,k}, \infty\}$ for each operating configuration ρ_i. The likelihoods of these clusters centers at state $s \in \boldsymbol{S}$ are estimate as:

$$p(o_{\rho_i,j}|s,\rho_i) = \int_{(o_{\rho_i,j-1}+o_{\rho_i,j})/2}^{(o_{\rho_i,j}+o_{\rho_i,j+1})/2} \mathcal{N}(\mu_s,\sigma_s) \tag{28}$$

where μ_s and σ_s are the mean and variance of the Gaussian distribution of state s. The resulting probabilities are used in eq. (24) to estimate the expected risk. The integration is performed numerically using the trapezoidal rule. We chose *k*-means clustering for the discretization process because it provides a more accurate representation of the distribution in continuous feature space than uniform discretization [182]. Also, unlike uniform discretization *k*-means clustering is robust to outliers [183].

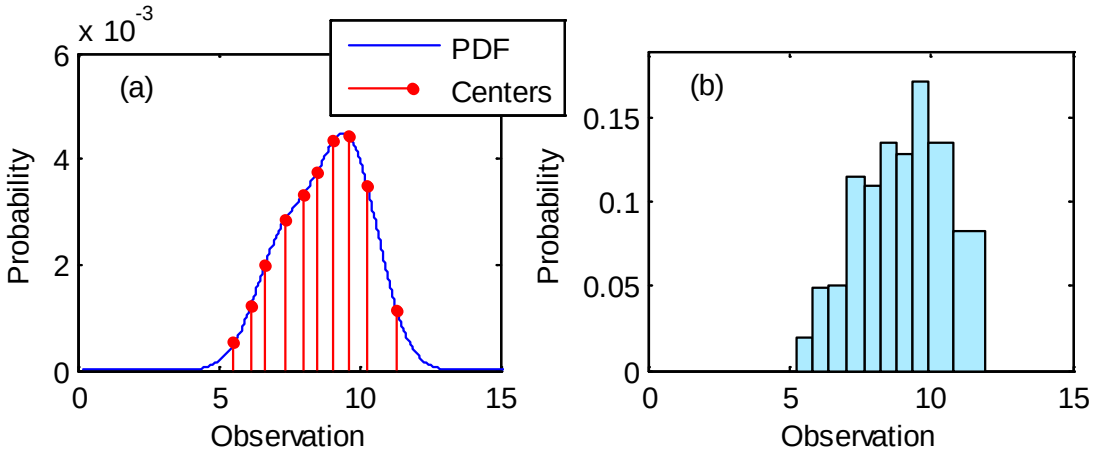

Figure 11 (a) The probability distribution function (PDF) from which the training data was generated, and the centers resulting from *k*-means clustering. (b) The discretized observation space and the corresponding probabilities.

The discretization process is demonstrated with an example in Figure 11. In this example, the training data was generated from a probability distribution, which is a sum of three Normal distributions with randomly generated centers at 7.30, 9.17, and 9.98 (Figure 11 (a)). The data was then discretized into 10 bins using *k*-means clustering. The likelihoods of the cluster centers (estimated using eq. (28)) are shown in Figure 11 (b).

4.2 Validation on synthetic data

We validated AC through a series of experiments. Our experimental work is divided into two sections based on the type of data used: synthetic and experimental. The first set of experiments performed on synthetic data allowed us to test AC extensively. Such comprehensive evaluation on experimental data is impractical due to the prolonged time required for data collection. The second set of experiments establishes an experimental proof-of-concept.

We conducted four experiments on synthetic data. The first experiment characterizes the classification performance of AC with varying sensing costs. The second experiment compares

the performance of AC and sequential forward selection (SFS) with the number of sensing steps. The third experiment also compares performance of AC and SFS, but with varying levels of signal-to-noise ratio (SNR) in test data.

4.2.1 Data

We simulated the response of an FPI to different chemicals using infrared absorption data obtained from NIST chemistry WebBook (http://webbook.nist.gov/chemistry/). This database provides high resolution (250 points per μm) FTIR spectra in the range 3-21 μm for over 16,000 chemicals in gas phase. We chose 50 chemicals from this database that had strong absorption peaks in the range 3-4.3 μm, which is the typical operating range of commercially available FPI sensors. To simulate the spectral resolution of the FPI sensors, the FTIR spectra were down-sampled to 44 values each. This is the typical resolution or number of unique wavelength tunings in commercially available FPI. The resulting absorption spectra of the 50 compounds are shown in Figure 12, and their names are provided in APPENDIX B.

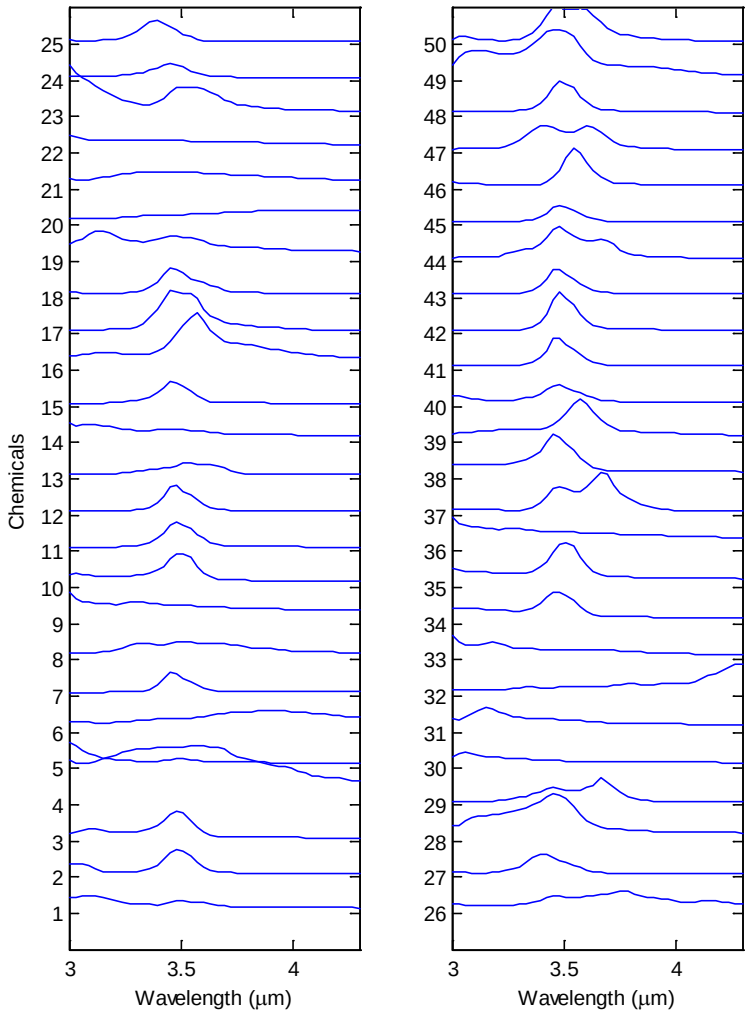

Figure 12 Absorption spectra of the fifty chemicals as a function of the wavelength. For visualization purposes, the spectra are plotted in two columns with an offset along the y-axis.

Using these spectra, we generated 10 samples for each chemical by adding Gaussian noise of standard deviation 0.03 at each wavelength. The resulting dataset containing 500 (50×10) samples was used for training purposes. Similarly, we generated another dataset containing 1000 samples (20 per chemical) for testing purposes, but with an additive noise of standard deviation 0.05.

4.2.2 Classification performance vs. sensing costs

In the first experiment, we tested the performance of AC with varying sensing costs. For this experiment, we first developed sensor models using GMMs (as described in section 4.1.1) with $K = 2$ components per each mixture. The misclassification costs were set to be a 0-1 loss function $c_{uv} = \begin{cases} 0 \; if \; u = v \\ 1 \; if \; u \neq v \end{cases}$, and the sensing costs to be uniform, i.e. c_i is the same for all D configurations. We then tested the performance of AC with varying the values of c_i, from $c_i = 0$ to $c_i = 1$ in steps of 0. 1.

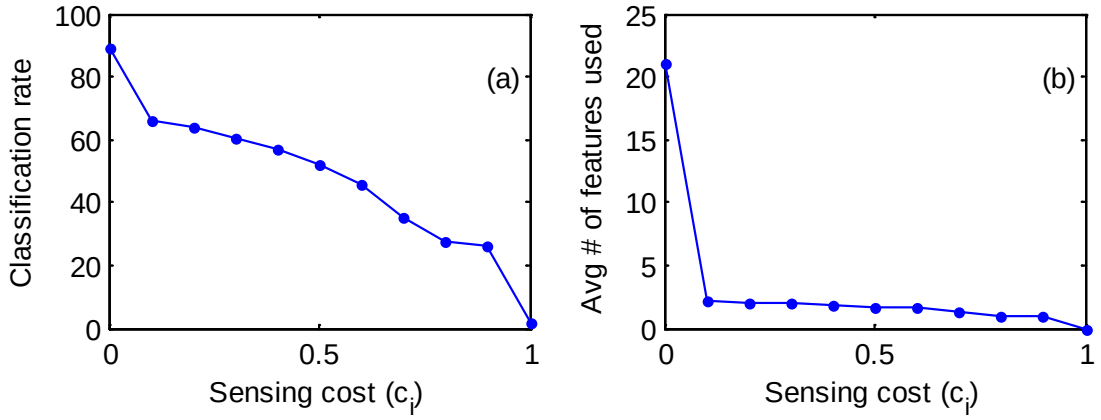

Figure 13 (a) The classification performance of AC, and (b) the average number of features used by AC as a function of c_i.

The results are summarized in Figure 13. The classification performance of AC deteriorates from 89.2% at $c_i = 0$ to 2% at $c_i = 1$. Likewise, the average number of features used by AC also reduces from 21.13 at $c_i = 0$ to 0 at $c_i = 1$. Since, the misclassification costs are fixed and the sensing costs are varied, the expected sensing risks (eq. (24)) and classification risks (eq. (25)) remain the same. However, with increasing sensing costs (c_i), the utilities of the

sensing actions reduce (eq. (26)) and therefore become negative in the earlier sensing steps, which forces AC to halt the sensing process. This is reflected in the average number of features used and therefore the classification performance of AC.

When the sensing costs are increased from $c_i = 0$ to $c_i = 0.1$, there is a large drop in the classification rates: form 89.2% to 66%. At $c_i = 0$, AC will acquire features uninhibitedly and would only stop acquiring features if the expected risks of all sensing actions are negative. This uninhibited behavior is reflected in the average number of features acquired by AC, which is 21.1 at $c_i = 0$ but only 2.22 at $c_i = 0.1$. We observed a similar drop in classification rates from 26.2% at $c_i = 0.9$ to 2% at $c_i = 1$. At $c_i = 1$, AC does not acquire any features because from the very first sensing step the expected reductions in risks of all sensing actions are negative.

To better understand the behavior of AC at lower values of c_i, we conducted another experiment where the sensing costs were varied on a logarithmic scale: $c_i = 10^{-10}$ to $c_i = 0.1$ in steps of 10^{-1}.

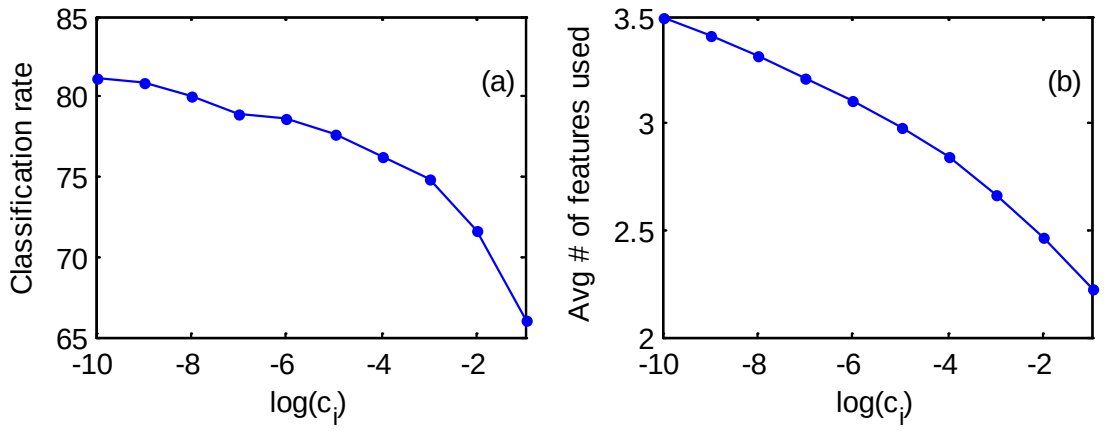

Figure 14 (a) The classification performance of AC, and (b) the average number of features used vs. $\log(c_i)$.

The results are summarized in Figure 14. The classification performance of AC deteriorates from 81.1% at $c_{uv} = 10^{-10}$ to 66% at $c_{uv} = 0.1$. Similarly, the average number of features reduces from 3.49 at $c_{uv} = 10^{-10}$ to 2.22 at $c_{uv} = 0.1$. This suggests that the AC method is very sensitive to the sensing costs, especially at the lower ranges.

4.2.3 Comparison with SFS

In the second experiment, we compared the classification performance of AC and SFS with varying number of sensing steps. In contrast to AC where the features are selected online, SFS is a passive strategy that selects an 'optimal' feature subset off-line using training data. SFS operates in a greedy fashion: it starts with an empty feature set and sequentially adds the feature that maximizes the value of an objective function. In our case, the objective function was the classification performance of a naïve Bayes classifier with 10-fold cross-validation. We applied SFS on training data to generate feature subsets of 20 different cardinalities ($1 \leq f \leq 20$). For each cardinality f, we trained a naïve Bayes classifier on the corresponding training data. To ensure a fair comparison between the AC and SFS, we modified the stopping criterion of AC such that it would not acquire more than f features for a given test sample. Namely, in step 5 of Table 2, we changed the first line to be $t \leq f$. We set misclassification costs to be a 0-1 loss function, i.e. $c_{uv} = \begin{cases} 0 \; if \; u = v \\ 1 \; if \; u \neq v \end{cases}$, and $c_i = 0$ for the 44 sensor configurations. We tested the performance of both methods at different values of f ranging from $f = 1$ to $f = 20$.

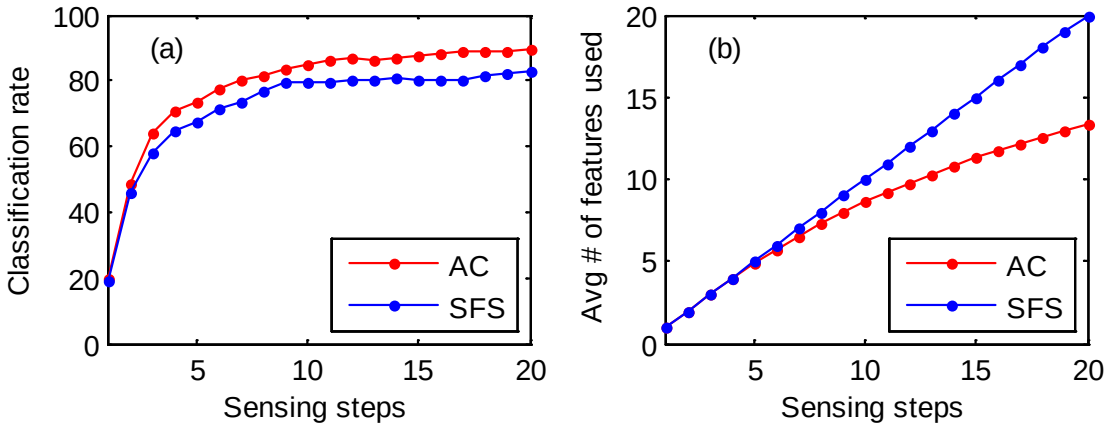

Figure 15 (a) The classification performance of AC and SFS, and (b) the average number of features used by AC and SFS as a function of the number of sensing steps f.

The results are summarized in Figure 15. The classification performance of AC increases monotonically from 19.4% at $f = 1$ to 89.5% at $f = 20$. Likewise, the performance of SFS increases from 19.4% at $f = 1$ to 84.6% at $f = 20$. AC outperforms SFS at all settings; the average difference between the two methods across all values of f is 5.8%. This superior performance of AC could be attributed to its: adaptive nature, where it acquires features during the test time, and stopping criteria, which allows for it to stop acquiring features when they are no longer expected to reduce Bayes risk. Since SFS is forced to use one feature at each sensing step, the average number of features increases linearly with the number of sensing steps. On the other hand, AC stops the sensing process when there is no expected reward from acquiring more features. Therefore, the average number of features used by AC is always less than that of SFS (Figure 15 (b)).

We further analyzed how the average entropy of the belief distributions changed with the number of sensing steps (for both AC and SFS). To estimate the entropies for AC, we compressed each belief distribution b_t into 50 values, where the values correspond to the

posterior probabilities of the 50 classes: $p_t(\omega_v) = \sum_{s \in H_{(v)}} b_t(s)$. We then calculated the entropy of the posterior distribution p_t as $-\sum_v p_t(\omega_v) \log_2\big(p_t(\omega_v)\big)$. We repeated this process for all test cases and for the first ten sensing steps. Then, we estimated the average entropy (across the 1000 test cases) for each sensing step. For test cases, where AC took a classification action before the tenth sensing step, the entropy of the last belief distribution was used as a proxy for the latter sensing steps. We repeated the same process for SFS, but in this case the posteriors at each time step were obtained from the naïve Bayes classifiers used by SFS.

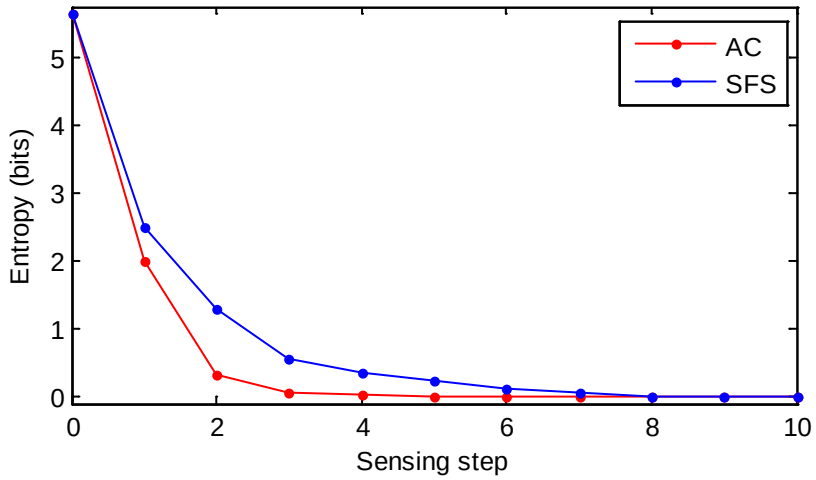

Figure 16 Average entropy (across all test samples) as a function of the sensing steps for AC and SFS.

The results are summarized in the Figure 16. At $t = 0$, since the belief distribution is initialized to be uniform, the average entropy for both AC and SFS is 5.64 bits. This value corresponds to the entropy of a uniform probability distribution of 50 values. As more observations are obtained, the average entropy for both methods decreases monotonically approaching values close to zero by the eight sensing step. However, the average entropy values

decrease faster for AC than for SFS. In fact, for AC the average entropy approaches zero by the fourth sensing step. This faster reduction in entropy can be attributed to the fact that AC selects features based on the beliefs generated by the test instance at hand, as opposed to SFS which uses the same sequence of features for all test cases.

4.2.4 Performance vs. SNR

In the third experiment, we compared the classification performance of AC and SFS at different levels of SNR in test data. We first generated ten test datasets with varying levels of additive Gaussian noise ranging from a standard deviation of 0.01 to 0.1 in steps of 0.01. Once again, the stopping criterion of AC was changed to ensure a fair comparison between SFS and AC. The misclassification costs were set as $c_{uv} = \begin{cases} 0 \; if \; u = v \\ 1 \; if \; u \neq v \end{cases}$, and the sensing costs were set as $c_i = 0.1$ for all configurations. We then tested the two methods for three different values of f: $f = 5$, $f = 8$, and $f = 10$.

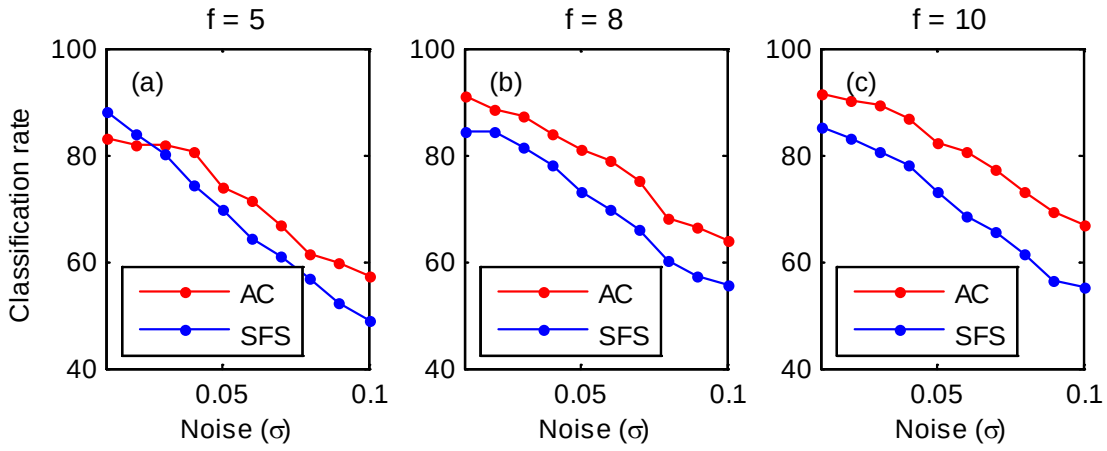

Figure 17 The classification performance of AC and SFS vs. the noise in test data for different number of sensing steps (a) $f = 5$, (b) $f = 8$, and (c) $f = 10$.

The results are summarized in the Figure 17. For $f = 5$, the classification performance of both methods deteriorate monotonically with increasing levels of noise in test data from 83.1% (AC) and 88% (SFS) at $\sigma = 0.01$ to 57.2% (AC) and 49% (SFS) at $\sigma = 0.1$. For $f = 5$, AC outperforms SFS at all noise levels expect for $\sigma = 0.01$ and $\sigma = 0.02$. We observed similar behavior for $f = 8$ and $f = 10$, however, for these settings AC outperforms SFS for all SNRs. Once again, this superior performance of AC could be attributed to its adaptive nature where it selects features at measurement time, which allows it to adapt to noise. On the other hand, SFS uses a pre-specified set of features that were selected off-line using noise-free training data.

4.3 Experimental validation

We validated AC on a commercially available MOX gas sensor (TGS 2620) manufactured by Figaro. We conducted three experiments on this sensor. The first experiment compares the performance AC with SFS for various number of sensing steps (as in section 4.2.2). The second experiment characterizes the performance of AC with using non-uniform sensing costs. In the third experiment, to establish a proof-of-concept, AC was tested in an online fashion.

4.3.1 Data

We collected a dataset of temperature-modulated responses of the MOX sensor to the following five household chemicals: xylene, denatured alcohol, mineral spirits, turpentine, and ammonia. This data was collected using the experimental apparatus described in APPENDIX A. Before the data collection, we conducted a preliminary study to determine concentrations for each chemical at which they had similar isothermal responses. For this study, we collected the isothermal responses of the sensor to all analytes at 5 different concentrations (4% to 20%, in steps of 4%, concentrations are expressed in volume/volume), and identified concentrations at

which the sensor's isothermal responses were similar. To obtain the isothermal response, the sensor was exposed to each analyte for 100 seconds under a constant heater voltage of 5 V. Our goal here was to ensure the samples could not be distinguished based solely on the amplitude of the responses and that temperature modulation would be needed. The final concentrations are specified in Table 3.

Table 3 Concentrations at which the chemicals had similar isothermal responses.

Chemical	Concentration
Xylene	12%
Denatured alcohol	4%
Mineral spirits	4%
Turpentine	8%
Ammonia	16%

The sensor was operated at 10 different heater voltages ranging from $3 \leq V_H \leq 7.5$ in steps of 0.5 V. At each operating voltage, the sensor was pulsed for duration of 20 seconds. Between two consecutive pulses, the sensor was reset to a baseline voltage (0V) for duration of 10 seconds. This form of temperature programming helped reduce variance in the sensor response due to thermal dynamics. Figure 18 shows the transient response of the sensor to all five chemicals when driven with a random sequence of five voltage pulses. For each chemical, the sensor was operated with 20 random sequences, each sequence containing the 10 operating voltages; this resulted in a dataset with 100 samples (5 chemicals × 20 sequences). The chemicals were presented in a randomized order to avoid systematic errors. The information from each transient was reduced to a single observation by computing its integral response.

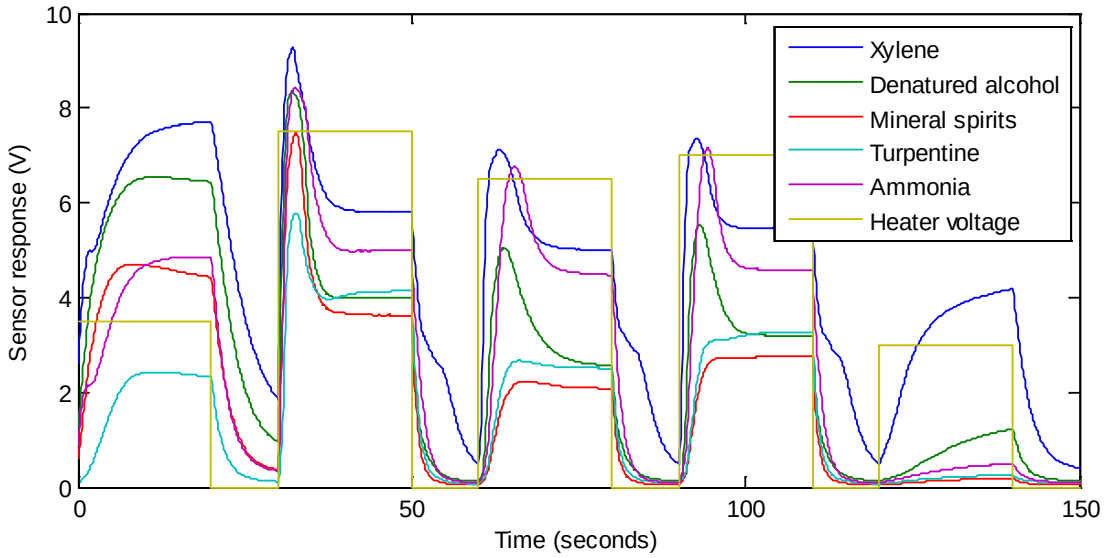

Figure 18 Transient sensor responses to the five chemicals. The sensor was driven with a sequence of five voltage pulses, each 20 second long.

4.3.2 Offline testing

In the first experiment, we compared the performance of AC against SFS with varying number of sensing steps. This experiment was conducted with a 10-fold cross-validation loop. For each fold, the experimental data (100 samples) was randomly divided into two subsets: a training dataset containing 40 samples (8 per chemical), and a test dataset containing 60 samples (12 per chemical). We ran SFS on the training datasets to generate optimal feature subsets of 10 different cardinalities $1 \leq f \leq 10$. A naïve Bayes classifier, wrapped in a 10-fold cross-validation loop, was used as the objective function for SFS. For each of these feature subsets, we trained a naïve Bayes classifier on the training data and estimated the classification performance on the corresponding test datasets. We used the same cross-validation loop to estimate the performance of AC. We trained the sensor models (IOHMMs) on each of the training sets (one IOHMM per each class) with $K = 2$ components per each Gaussian mixture. This resulted in a

POMDP with $5\times10\times2=100$ states. We then estimated the performance of AC on the corresponding test sets. Our implementation of the IOHMMs was based on Kevin Murphy's Dynamic Bayesian network toolbox [184], which is an open-source Matlab toolbox for directed graphical models. To ensure a fair comparison, the stopping criterion of AC was modified such that it would not acquire more than f observations for a given test case.

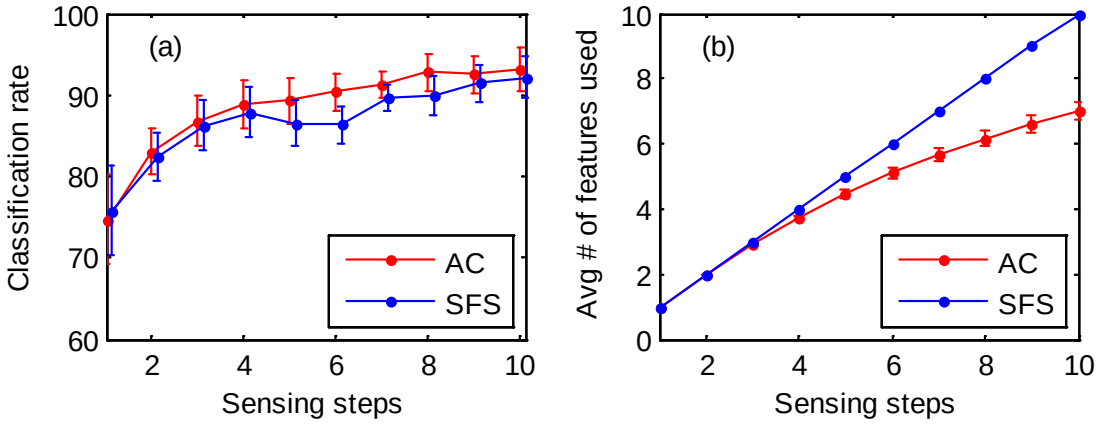

Figure 19 (a) Classification performance of AC and SFS vs. the number of sensing steps. For visualization purposes, the SFS curve was shifted slightly along the x-axis. (b) The average number of features used by AC and SFS vs. the number of sensing steps.

The results are summarized in Figure 19. The classification performance of AC increases from 74.6% at $f=1$ to 93.2% at $f=10$. Similarly, the classification performance of SFS increases from 75.1% at $f=1$ to 92.1% at $f=10$. However, unlike with AC, the classification rates of SFS do not increase monotonically showing a small drop between $f=4$ to $f=6$. AC consistently outperformed SFS at all values of f except for $f=1$. Once again, the average number of features used by SFS increases linearly with f since SFS is forced to acquire one feature at each time step, whereas, the average number of features used by AC is always less

than SFS because AC stops the sensing process when there is no expected reward from acquiring more features. These results are consistent with our observations on synthetic data presented in section 4.2.2.

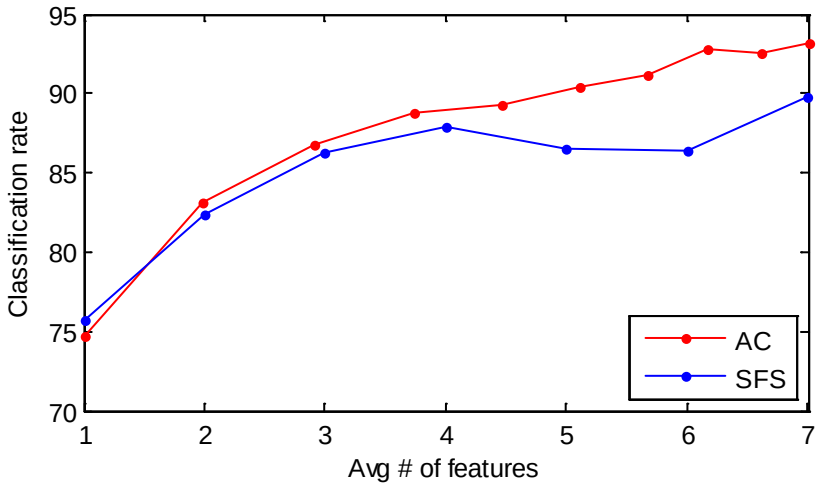

Figure 20 Classification performance of AC and SFS vs. the average number of features used.

Figure 20 combines the results from Figure 19 (a) and Figure 19 (b). It plots the mean classification performance of AC and SFS against the average number of features used. This representation of the results is fairer towards AC because the x-axis represents the average number of features used as opposed to the number of sensing steps (as in Figure 19 (a)). Though AC and SFS are set to run for the same number of sensing steps, AC does not necessarily go through all the steps because for some test cases AC stops when features do not offer any expected reduction in risk. Therefore, Figure 20 offers a more convincing perspective that AC obtains better classification performance than SFS using fewer features.

4.3.3 Energy-aware active sensing

In the previous experiment, the feature acquisition costs were set uniform, i.e. all temperature pulses had the same costs. In this experiment, we consider an experimental scenario with non-uniform acquisition costs, i.e., the temperature pulses have different costs. Using the dataset described in section 4.3.1, we created such a sensing scenario with 30 voltage pulses (sensing actions) of different amplitudes, durations, and costs. We compressed the information from the transient sensor responses to each 20-second pulse as three different observations. The first observation is the integral of the first five seconds of the transient. The second observation is the integral over the first 10 seconds of the transient, and similarly the third observation is the integral of the entire 20 second transient. Thus, by using the early parts of the transients as proxies for pulses of shorter durations, we created a dataset containing responses to 30 pulses (10 voltages $\times$ 3 durations). Then, we defined the sensing cost of each pulse as a product the amplitude of the pulse (in Volts), the duration (in sec), and a constant $\xi = 10^{-3}$. For example, the sensing cost of a 5V pulse of duration 10 seconds would be 0.05 units. These sensing costs reflect the energy required to heat the sensor. The constant ξ allows us to keep the sensing risks and sensing costs in a similar range.

With the sensing costs defined as above, we tested the performance of AC with varying values of misclassification costs $c_{uv} = 0.1$ to $c_{uv} = 1$ in steps of 0.1. As in section 4.3.2, we ran this experiment with a 10-fold cross-validation loop. For each fold, we randomly divided the experimental data (100 samples) into two subsets: a training dataset containing 40 samples (8 per chemical), and a test dataset containing 60 samples (12 per chemical). We trained a separate IOHMM for each class with $K = 2$ components per each Gaussian mixture resulting in a POMDP with $6 \times 30 \times 2 = 360$ states.

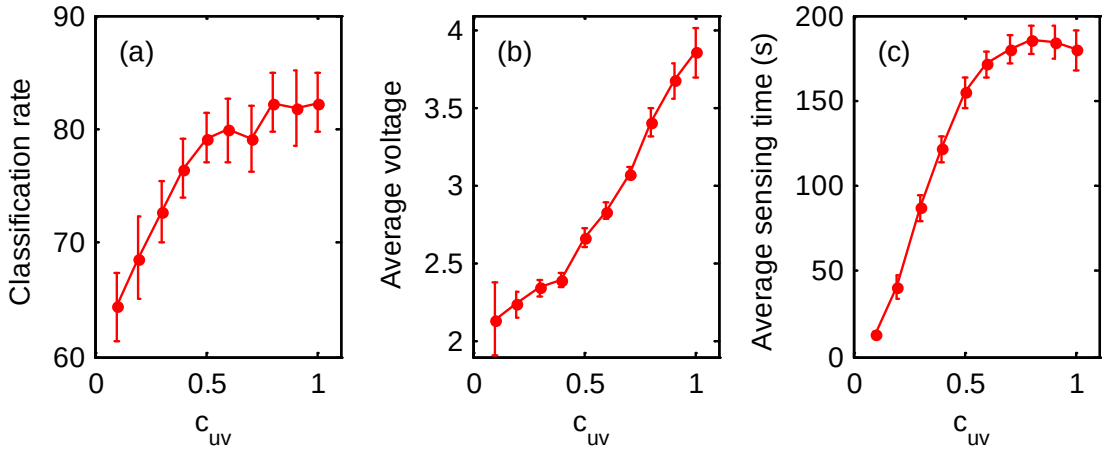

Figure 21 (a) The classification rate, (b) average sensing cost, and (c) average sensing time as a function of the misclassification costs c_{uv}.

The results are summarized in Figure 21. As shown in Figure 21(a), the classification performance of AC increases from 64.4% at $c_{uv} = 0.1$ to 82.3% at $c_{uv} = 1$. Likewise, the average heater voltage (Figure 21(b)) increases from 2.1V at $c_{uv} = 0.1$ to 3.86V at $c_{uv} = 1$. We estimate the average heater voltage as weighted average of all pulses selected by AC, the weights being the durations of the pulses. The result shows that, with increasing misclassification costs, AC tends to use pulses of higher amplitude. Similarly, the average sensing time (estimated as the average duration required to classify each sample) increases with misclassification costs (Figure 21(c)), showing that AC selects pulses of longer duration to counter the increasing misclassification costs. These results are consistent with our observations on synthetic data presented in section 4.2.2.

We also analyzed the frequency at which AC selects each voltage pulse (across all test cases) for misclassification costs of $c_{uv} = 0.5$ and $c_{uv} = 1$; the corresponding classification rates were 79.2% and 82.2% respectively.

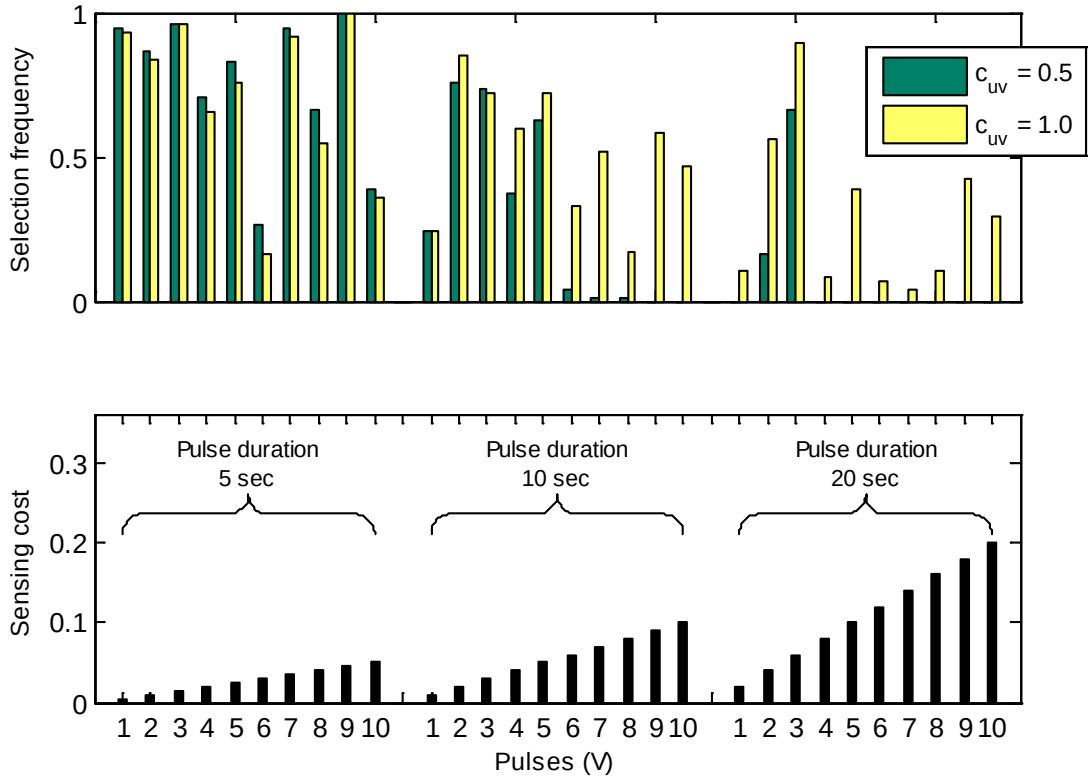

Figure 22 (top) Frequency of selecting each feature for two misclassification costs c_{uv} = (0.5, 1.0). (bottom) Sensing costs associated with each pulse organized by heater voltage and pulse durations.

The results are summarized in Figure 22. At $c_{uv} = 0.5$, AC avoids using any of the costlier pulses. In fact, at $c_{uv} = 0.5$, AC did not use any of the pulses with sensing costs greater 0.08, most of which are of 20-second duration. On the other hand, AC tends to use the costlier pulses more frequently at $c_{uv} = 1$ to balance the sensing costs against the risk of misclassifying a given sample.

4.3.4 Online testing

The last two experiments evaluated AC in an offline fashion: the algorithm was trained and tested on pre-collected data. This type of offline testing allowed us to adjust various

experimental parameters such as the number of components used in the GMMs (K), misclassification costs (c_{uv}), and sensing costs (c_i). It also, allowed us to test the method at various experimental settings such as the number of sensing steps (f), and pulse durations.

As the final step, to establish a proof-of-concept, we validated AC in an online fashion. Using the protocols described in section 4.3.1, we collected a dataset of temperature-modulated sensor responses for three chemicals: denatured alcohol, mineral spirits, and ammonia. For each chemical, we collected sensor's response to 10 random pulse sequences, with each sequence containing the 10 operating voltages ($3 \leq V_H \leq 7.5$); this resulted in a dataset with 30 samples (3 chemicals × 10 sequences). The information from each transient was reduced to a single observation by computing its integral response. Using this data, we trained 3 IOHMMs (one per chemical), with $K = 2$ components per each Gaussian mixture resulting in a POMDP with $3{\times}10{\times}2 = 60$ states. Then, we tested AC 60 times (20 times per chemical) in an online manner with $f = 10$ sensing steps. The misclassification costs were set to be $c_{uv} = \begin{cases} 0 \; if \; u = v \\ 1 \; if \; u \neq v \end{cases}$, sensing costs $c_i = 0$ (for all ten voltages). For each test case, the sensor's heater voltage was first set to 0V, then the chemical was introduced into the sensor chamber for duration of 30 seconds, and then AC algorithm was commenced to drive the sensor. At each sensing step, the AC algorithm selected a voltage pulse, then the sensor was operated accordingly, and features extracted from the resulting transient responses were used to update the belief distribution. The AC algorithm continued until one of the following two stopping criteria was met: (1) 10 pulses were used, or (2) the expected reduction in Bayes risk for all the unused pulses was negative. For test cases where AC ran into the second stopping criterion, it was forced to execute the remaining unused pulses since the data was necessary to validate SFS. We kept track of all the intermediary belief distributions that were generated by AC because they were required to estimate the classification performance for $f = 1$ to $f = 9$. After each test case, the sensor chamber was flushed with

desiccated air for duration of two minutes to remove any remnants from the previous test case. The chemicals were presented in a randomized order.

We also evaluated SFS with varying number of sensing steps ($f = 1$ to $f = 10$). We ran SFS on the training data, with a naïve Bayes classifier in 5-fold cross-validation as the objective function, to select feature subsets of ten different cardinalities. We then estimated the performance of SFS on the data obtained from the test cases of AC; this ensured that both AC and SFS were tested on the same dataset.

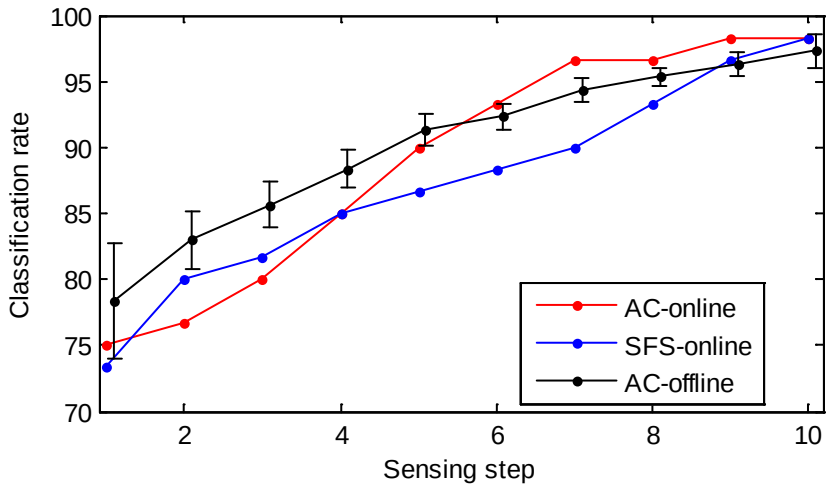

Figure 23 Classification performance of AC-online, AC-offline, and SFS-online vs. number of sensing steps.

Figure 23 summarizes the results of the experiment. The classification performance of AC (red curve) increases from 73.3% at $f = 1$ to 97.3% at $f = 10$, reaching its peak performance by the 9th sensing step. The performance of SFS (blue curve) also increased from 75% at $f = 1$ to 97.3% at $f = 10$. AC obtained better classification performance than SFS at most settings, except for at $f = 2$, and $f = 3$, where SFS outperformed AC by 3.33% and 1.67%

respectively. In general, this result follows the trends of the previous results, except for the two inconsistencies as mentioned above. We believe this was most likely caused due to over-fitting: AC selects features based solely on its sensors models but SFS's selection is driven by a cross-validation step, which addresses some of the issues that arise due to over-fitting. Though the cross-validation step helps SFS select useful features in the preliminary sensing steps, as AC starts learning more about the sample its begins to outperforms SFS.

To check if this explanation was accurate, we conducted another offline experiment, where we generated 10 training datasets with each dataset containing 80% (24 samples, 8 per chemical) of the samples randomly selected from the original training data. Then, we trained sensor models on these ten training sets and tested AC on the same 60 samples. The results are presented in terms of average classification performance (the black curve termed AC-offline) in Figure 23. As shown in the figure, AC-offline outperforms SFS at most settings except for $f = 9$ and $f = 10$, where the difference is below 1%; less than the standard deviation of AC-offline. More importantly, AC-offline outperforms SFS-online by 3% and 4% at $f = 2$ and $f = 3$. Putting all these observations together, we believe that this result is consistent with our hypothesis, and we also conclude that this experiment serves as a proof-of-concept and adds to the completeness of the dissertation.

4.4 Discussion

Our results in sections 4.2.2 suggest that the AC method is very sensitive to the cost settings, especially at the lower ranges of c_i. Upon further analysis, we found that this issue was due to machine precision. The belief update equation (eq. (23)) uses the term $p(o_t|s, \rho_t)$, which is the probability of making observation o_t at state s, given the sensing action was ρ_t. Here, the state s corresponds to a Gaussian distribution with mean μ_s and standard deviation σ_s. Since this is a continuous probability density function, the probability $p(o_t|s, \rho_t)$ is estimated as

proportional to $\frac{1}{\sqrt{2\pi}}\exp-\frac{(o_t-\mu_s)^2}{2{\sigma_s}^2}$. Therefore, even when an observation o_t is relatively far away from the mean of a given state, it would still have a non-zero probability. For example, consider a state s with parameters $\mu_s = 0,$ and $\sigma_s = 1$. If the observation was $o_t = 5$, which is five standard deviations away from the mean, then $p(o_t|s,\rho_t) = 1.48\times10^{-6}$. Similarly if the observation was $o_t = 7$, which is seven standard deviations away from the mean, then $p(o_t|s,\rho_t) = 9.1\times10^{-12}$. When these probabilities are used in belief update equation (eq. (23)) and then eventually in the risk and utility estimations (eq. (24), eq. (25), and eq. (26)), then some of the sensing actions end up with utilities very close to zero but still non-zero (sometimes even in the orders of 10^{-10}). These non-zero utilities force AC to acquire more features and that obviously improves the classification performance (as shown in section 4.2.2). Therefore, AC is quite sensitive to cost settings. One way to address this issue is to round off the low probability values to zeros. Another option would be to use either the total sensing cost, or the total number of features acquired as a stopping criterion, as in sections 4.2.3.

The experimental results in sections 4.3.2 and 4.3.3 show a large standard deviation in classification rates (in the order of 5%). This suggests that there is significant amount of variation in the data (even among the samples of the same class). We believe this due to the experimental setup: the chemicals are delivered to the sensor chamber from headspaces of 30 ml glass vials connected through dilutors, so as more data samples are collected the vapor concentrations the headspaces reduce because of reduction in the volumes of the chemicals. This problem could be addressed by using better experimental equipment such as: pressurized cylinders, which can maintain fixed vapor pressures and therefore deliver chemicals at fixed concentrations, and mass flow controllers, which can maintain steady flow rate in the delivery process. Another challenge with our experimental setup is that we have to constraint the total number of samples, because collecting each sample takes in the order of 5 minutes. This is a

limitation of the particular sensors we used for validation purposes (Figaro TGS sensors have long time-constants). If tested on newer metal-oxide sensor technologies, such as micro-hotplate arrays that have very low thermal constants, then we can collect larger datasets.

In the experimental work, we only used the integral response of the sensor transients as features because our focus is on establishing an experimental proof-of-concept. However, any of the more advanced feature extraction techniques (as detailed in section 2.4.1) could easily be incorporated into the overall framework: the extracted features would constitute the observation o_t. We do not expect that the choice of the feature extraction technique would change the main results of the chapter, which is that AC outperforms SFS with fewer measurements, and is robust to noise. The absolute performance of both approaches may differ with other feature extraction techniques, but not their relative performance.

5. ACTIVE MULTICOMPONENT ANALYSIS

In a multicomponent analysis problem, the given chemical sample is assumed to be a mixture of known non-interacting gases, and the goal is estimating the concentrations of the constituent gases. Unlike classification problem, where the hypothesis is a categorical variable, in multicomponent analysis, the hypothesis space is continuous and multi-dimensional. Therefore, classification problem could be considered as a special case of the multicomponent analysis problem. In practice, multicomponent analysis is limited to a few components (2 to 4) due to chemical sensors' cross-selectivity, and due to the exponential growth in the number of required training samples with the number of components in the mixture.

Researchers have been studying multicomponent analysis for more than two decades [185, 186]. The typical approach to this problem is to build a mapping from the independent variables (sensor responses) to the dependent variables (concentrations). Both linear (e.g. ordinary least squares) and nonlinear (e.g. neural networks) models have been used for building these maps. In linear models, the concentrations are modeled as a weighted linear combination of the sensor responses, and the weights are estimated in a least-squares fashion by minimizing the sum-squared prediction error over training data. These weights could be further stabilized by using methods such as principal components regression [187, 188], ridge regression [189], or partial least squares [190-192]. Nonlinear models have also been used based on several artificial neural networks architectures, including multi-layer perceptrons [186, 193], self-organizing maps [194], and time delay neural networks [195, 196]. However, in all these methods the sensor responses are generated by driving the sensor with a pre-determined program. To our knowledge, there is one study by Vergara et al. [84] that has proposed a systematic approach to optimizing temperature programs applied to micro-hotplate metal-oxide gas sensors for

multicomponent analysis. In this paper, the authors used a multi-sinusoidal signal for temperature modulation and trained a PLS regression model to predict the constituent concentrations from sensor responses. Then, they selected the best operating frequencies using 5-fold cross-validation to quantify different mixtures of ethylene, ammonia, and acetaldehyde. Vergara's study was based on feature subset selection, which is an offline technique. Here, we present an approach that actively optimizes the sensor program to estimate the concentrations.

5.1 Methods

Given a sample that is a mixture of n known non-interacting gases, and a chemical sensor that operates D different settings $\boldsymbol{\rho} = \langle \rho_1, \rho_2, \dots, \rho_D \rangle$, we seek to find an optimized sequence of actions to estimate the concentrations $\theta = (c_1, c_2, \dots, c_n)$ of the n gases, where an action corresponds to operating the sensor at one of the D configurations. For simplicity, we assume that each gas concentration c_i could have k_i different values[6]. As a result, the state of the system can be represented by a vector θ with $r = k_1 \times k_2 \times \dots \times k_n$ different admissible values: $\theta \in \boldsymbol{S} = \{s_1, s_2, \dots, s_r\}$, where s_i is an n-dimensional vector containing the discrete concentrations of the n gases.

We solve this problem through probabilistic state estimation by maintaining a belief distribution over the r possible concentration profiles[7]. The belief b_t at time t assigns a probability to each one of the r states:

$$b_t : \boldsymbol{S} \rightarrow [0,1]; \sum_{s_i \in \boldsymbol{S}} b_t(s_i) = 1 \tag{29}$$

[6] This assumption allows us to constrict the representation to a discrete state-space, which greatly simplifies the optimization problem from a computational standpoint.
[7] We will refer to this method as AMA, an abbreviation for active multicomponent analysis.

Figure 24 shows an example of a hypothetical belief distribution over $r = 10 \times 10$ profiles for a mixture of two chemicals. Given an initial belief $b_0(s)$, a sequence of operating configurations $(\rho_1, \rho_2, \dots, \rho_t)$ and the corresponding observations $(o_1, o_2, \dots, o_t)$, the current belief b_t is defined as:

$$b_t(s) = p(s|o_1, o_2, \dots, o_t, \rho_1, \rho_2, \dots, \rho_t) \tag{30}$$

We assume that the belief $b_t(s)$ acts as a sufficient statistic, and can be updated based on the previous estimate $b_{t-1}(s)$:

$$b_t(s) = p(s|o_t, \rho_t, b_{t-1}) \tag{31}$$

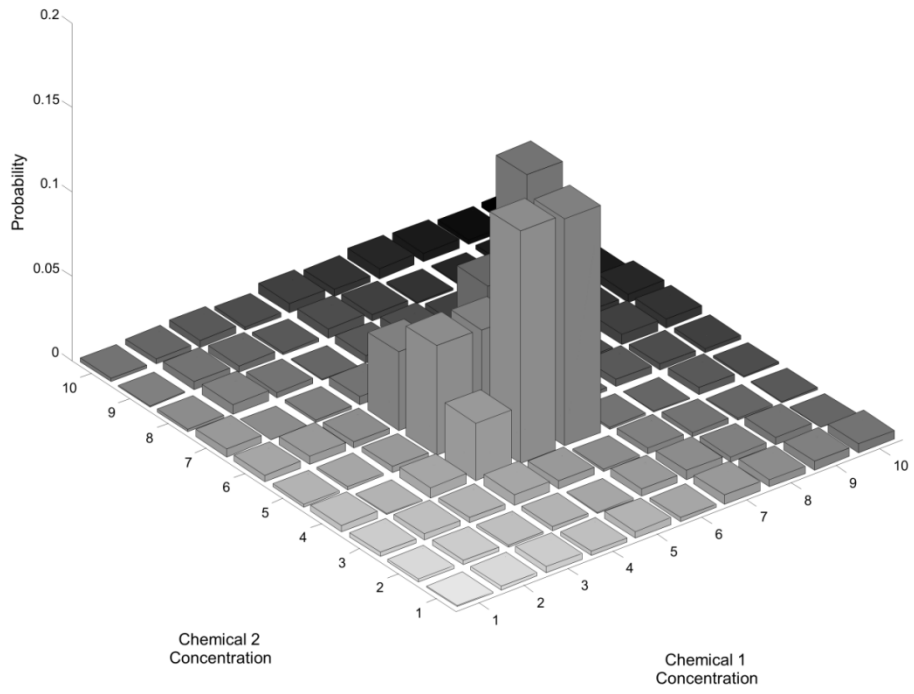

Figure 24 Example of a belief distribution over the 'concentration space' of a binary mixture of chemicals.

Our goal is to select a sequence of operating configurations that minimizes uncertainty in θ, which can be estimated as:

$$H(\theta) = -\sum_{s_i \in \boldsymbol{S}} p(\theta = s_i) \log\big(p(\theta = s_i)\big) \tag{32}$$

where $H(\theta)$ is the Shanon entropy. In what follows, we will use the notation $H(\theta_t)$ to represent the entropy of θ at time t, estimated as:

$$H(\theta_t) = -\sum_{s_i \in \boldsymbol{S}} p(\theta_t = s_i) \log\big(p(\theta_t = s_i)\big) = -\sum_{s_i \in \boldsymbol{S}} b_t(s_i) \log\big(b_t(s_i)\big) \tag{33}$$

As discussed in (section 3.2.2), an exact solution to this problem (called a policy π) is a mapping from a belief distribution b_t to an operating configuration $\pi: b_t \rightarrow \boldsymbol{\rho}$. However, finding an exact solution is P-SPACE complete [135] and thus computationally expensive. Therefore, we use a myopic approach that optimizes the sensing program on a per-configuration basis. Our myopic algorithm selects the next operating configuration ρ_{sel} as the one with the largest expected reduction in entropy:

$$\rho_{sel} = \underset{\rho_{t+1}}{\operatorname{argmax}}\big(H(\theta_t) - H(\theta_{t+1}|\rho_{t+1})\big) \tag{34}$$

where $H(\theta_{t+1}|\rho_{t+1})$ is the expected entropy of θ after the sensor is operated at ρ_{t+1}. After selecting the operating configuration, the system transitions to the next time step (i.e. $t = t + 1$). Then, we operate the sensor at $\rho_t = \rho_{sel}$ and measure the resulting response o_t; this observation o_t is used to update the belief distribution $b_t(s) = p(s|\rho_t, o_t, b_{t-1})$. In the final step, we estimate the entropy of the updated belief distribution. If the entropy is below a pre-determined threshold ψ, the concentration is declared as $\theta = \operatorname{argmax}_s b_t(s)$; this halts the sensing process

and avoids any further data acquisition. If on the other hand the entropy exceeds the threshold ψ, a new operating configuration is selected, and the process is repeated. A pseudo-code of the myopic algorithm is shown in Table 4. This pseudo-code leaves two questions unanswered: (1) How do we update the belief distribution, and (2) how do we estimate the expected entropy $H(\theta_{t+1}|\rho_{t+1})$?

Table 4 Pseudocode of the active multicomponent analysis algorithm.

Step 1: Initialization	• Initialize uniform belief distribution $b_0(s)$ • Initialize set of available configurations $\boldsymbol{\rho}$ • $t = 1$
Step 2: Select action	• Select best operating configuration $\rho_{sel} = \underset{\rho_{t+1}\in\boldsymbol{\rho}}{\operatorname{argmax}}\big(H(\theta_t) - H(\theta_{t+1}|\rho_{t+1})\big)$
Step 3: Sensing	• Operate the sensor at $\rho_t = \rho_{sel}$ o Obtain sensor response o_t o remove ρ_{sel} from $\boldsymbol{\rho}$
Step 4: Update belief	• Update belief distribution • $t = t + 1$
Step 5: Termination	• Measure the entropy $H(\theta_t)$ • If $H(\theta_t) > \psi$ o Go to step 2 • Else o Declare $\theta = \operatorname{argmax}_s(b_t(s))$

5.1.1 Belief update

Given the belief b_{t-1}, and action ρ_t leading to observation o_t, we obtain the updated belief distribution b_t using a sequential Bayesian update [197]:

$$b_t(s) = \frac{p(o_t|s,\rho_t)\sum_{s' \in \boldsymbol{S}} b_{t-1}(s')p(s|s',\rho_t)}{p(o_t|\rho_t, b_{t-1})} \tag{35}$$

where $p(o_t|s,\rho_t)$ is obtained from a probabilistic model of the sensor, $p(s|s',\rho_t)$ is the probability that θ changes from s to s' upon taking action ρ_t and $p(o_t|\rho_t, b_{t-1})$ acts as a normalization term, which ensures the belief distribution sums to 1. Since the concentration profile θ does not change over time, the transition model becomes:

$$p(s|s') = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{matrix} if\ s = s' \\ otherwise \end{matrix} \tag{36}$$

As a result, the belief update simplifies to:

$$b_t(s) = \frac{p(o_t|s,\rho_t)b_{t-1}(s)}{p(o_t|\rho_t, b_{t-1})} \tag{37}$$

At time $t = 0$, when no sensor measurements are obtained, we assume that all concentration profiles are equally likely, that is, $b_0(s) = \frac{1}{r}, \forall s \in \boldsymbol{S}$.

The term $p(o_t|s,\rho_t)$ denotes the probability of observing sensor response o_t at concentration setting s when sensor is operated at configuration ρ_t. To enable real-time estimation of concentrations, it is crucial that this likelihood value be computed efficiently. For this purpose, as described in section 4.1.1, we model the sensor responses using GMMs.

5.1.2 Estimating expected entropy

The term $H(\theta_{t+1}|\rho_{t+1})$ denotes the expected entropy in the state variable θ if the sensor were to be operated at ρ_{t+1}. We estimate the expected entropy as the average over all possible sensor observations weighted by the probability of each observation:

$$H(\theta_{t+1}|\rho_{t+1}) = \sum_{\forall o_{t+1}} p(o_{t+1}|\rho_{t+1}) H(\theta_{t+1}|\rho_{t+1}, o_{t+1}) \tag{38}$$

where $p(o_{t+1}|\rho_{t+1})$ is the probability of making observation o_{t+1} at configuration ρ_{t+1} irrespective of concentration, and $H(\theta_{t+1}|\rho_{t+1}, o_{t+1})$ is the entropy of θ at time $t+1$ following observation o_{t+1} at configuration ρ_{t+1}. Estimating $H(\theta_{t+1}|\rho_{t+1})$ is described in a greater detail in section 6.1.2.

The expected entropy calculation in eq. (38) is only applicable for discrete observations; for continuous spaces, the summation in eq. (38) becomes an indefinite integral. To address this issue, we discretize the observation space for each temperature using *k*-means clustering. The discretization process is similar to the one described in section 4.1.4.

5.1.3 Incorporating sensing costs

The above formulation does not take sensing costs into consideration. However, each sensing action ρ_{t+1} could have an associated sensing cost q_{t+1}. These sensing costs can be incorporated into the formulation by modifying the temperature selection criterion of eq. (34) as follows:

$$\rho_{sel} = \underset{\rho_{t+1}}{\operatorname{argmax}}\, U(\rho_{t+1}) = \underset{\rho_{t+1}}{\operatorname{argmax}}((H(\theta_t) - H(\theta_{t+1}|\rho_{t+1}) - \alpha \times q_{t+1}) \tag{39}$$

where $U(\rho_{t+1})$ is the utility of operating configuration ρ_{t+1}, defined as a weighted sum of the sensing cost q_{t+1} and the expected reduction in entropy. The parameter α acts as a constant that balances between two units of measurement: entropy and sensing cost, and ensures that the entropy measures and sensing costs have comparable magnitudes. If the utility of all operating configurations is negative, there is no incentive in any further data acquisition. This happens when the sensing costs of all operating configurations exceed their expected reductions in

entropy. Under this condition, we halt the sensing process and declare the concentration profile as $\theta = \text{argmax}_s(b_t(s))$.

5.2 Validation on synthetic data

As in the previous chapter, our experimental work is divided into two sections based on the data used: synthetic and experimental. Unlike chapter 4, where the active sensing methods were validated on infrared data, here we used a simulated model of MOX sensor to make the problem more challenging. According to Beer-Lambert law [198], the infrared absorption/transmission spectra of a chemical mixture is linearly proportional their concentrations. In contrast, the relation between MOX sensor responses and analyte concentrations follows an exponential model. In addition, MOX sensor responses are far less selective than infrared spectra making multicomponent analysis more challenging with MOX sensors. After weighing between the choices of maintaining consistency across the chapters against that of testing the methods on a more complex problem, we chose the latter.

We conducted two sets of experiments on synthetic data. The first experiment compares the performance of AMA against SFS with varying number of features used. The second experiment compares AMA against SFS with varying levels of SNR. In our implementation, SFS uses the classification performance of a naïve Bayes classifier with 10-fold cross-validation as the objective function.

5.2.1 Simulated sensor model

We used a simulated sensor model to generate a dataset consisting of sensor responses to mixtures of two non-interacting chemicals. Following Clifford and Tuma [199], we modeled the sensor response as:

$$o_t = \sum_i G_i(\rho_t) c_i^{\beta_i} + \gamma o_{t-1} \tag{40}$$

where ρ_t denotes temperature at time t, o_t is the sensor response, $G_i(\rho_t)$ is the sensor sensitivity to gas i at temperature ρ_t, c_i is the concentration of gas i, β_i is a gas-dependent parameter, and γ is a parameter that captures sensor dynamics (history effects). Following Wlodek et al. [200], we defined the sensitivity G_i of the sensor to a gas i as a sum of a univariate Gaussian and a linear function over the operating temperatures:

$$G_i(\rho_t) = l_i \rho_t + \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp - \frac{(\rho_t - \tau_i)^2}{2\sigma_i^2} \tag{41}$$

where l_i, σ_i, and τ_i are gas-dependent parameters.

We created an experimental scenario with 20 different temperatures (1,2,…,20) and two hypothetical chemicals: $(l_1, \sigma_1, \tau_1) = (0.1, 0.4, 12)$ and $(l_2, \sigma_2, \tau_2) = (0.08, 0.45, 6)$; parameters in the Clifford-Tuma model were set to $(\beta_1, \beta_2, \gamma) = (0.45, 0.5, 0.1)$. Figure 25 (a) shows the temperature-sensitivity profile of the simulated MOX sensor for two hypothetical chemicals. Figure 25 (b) shows the sensor's response to two different binary mixtures when the sensor temperature is ramped from 1 to 20. We assumed each chemical could be present at one out of five discrete concentrations ($c_1, c_2 \in (20,40,60,80,100)$), resulting in $r = 5 \times 5 = 25$ different concentration profiles.

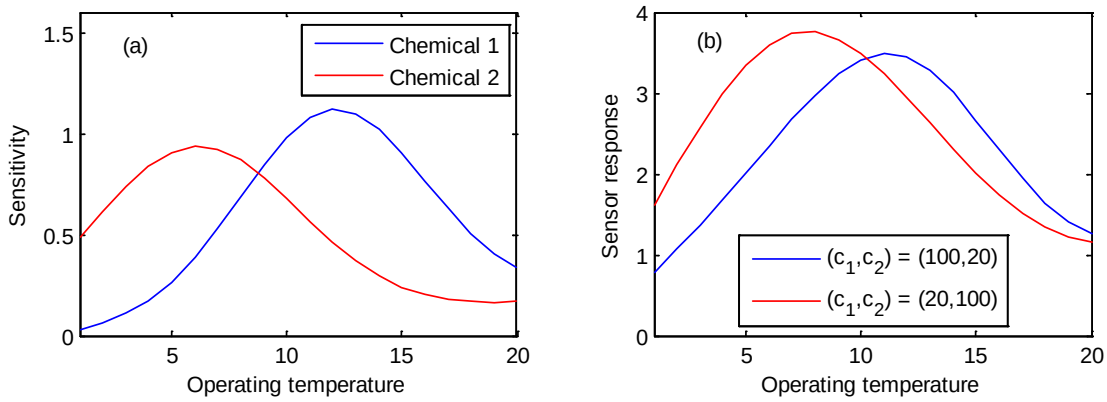

Figure 25 (a) Temperature-dependent sensitivity of the simulated MOX sensor to two hypothetical chemicals. (b) Response to two different binary mixtures when the sensor temperature was ramped from 1 to 20.

5.2.2 Comparison with SFS

In the first experiment, we compared the two methods (AMA and SFS) as a function of the number of operating temperatures used. For this purpose, we created a training dataset containing 125 samples. To create this dataset, for each one of the 25 concentration profiles, we generated five random permutations of the 20 operating temperatures and then obtained sensor responses using eq. (40); o_0 was initialized to a random value. Using this dataset, we trained GMM sensor models as described in section 5.1.1, and selected SFS feature subsets of different cardinalities f ($1 \leq f \leq 10$). At each cardinality f, we tested both methods on a dataset of 250 samples (10 per concentration profile). To ensure a fair comparison between both methods, we modified AMA such that the algorithm stopped sensing when either f observations were utilized or $H(\theta_t) < 0.65$[8]; in addition, we did not allow AMA to use the same feature twice.

[8] We chose the value of 0.65 for ψ because it reflects a situation with 90% certainty in the classification. In other words, consider a belief distribution that assigns a probability of 0.90 to one of the concentration profiles and 0.004 = 0.1/24 to each of the 24 remaining profiles. This distribution has entropy of 0.65 nat.

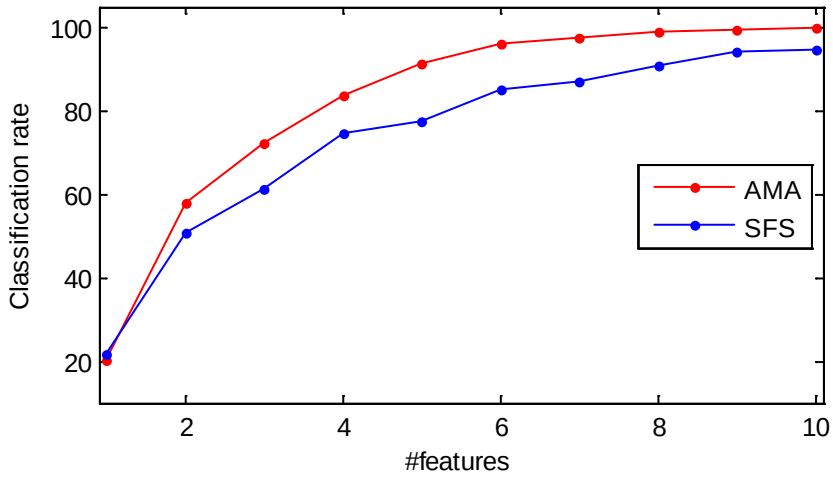

Figure 26 Classification performance of AMA and SFS as a function of the number of features used.

Results are summarized in Figure 26. The classification performance for both methods improved with increasing number of features. Interestingly, when only one temperature is used ($f = 1$), SFS outperformed AMA: 21.9% vs. 20.3%, respectively. This is likely due to the fact that SFS uses 10-fold cross-validation to identify the initial temperature, whereas AMA selects the feature based solely on the sensor models and has no provision to cross-validate the selection. However, for $f > 1$ AMA consistently outperforms SFS, and achieves perfect classification at $f = 7$. The superior performance of AMA may be attributed to its adaptive nature, which allows it to change the temperature program based on information obtained thus far from the test sample, whereas SFS always uses a pre-determined set of operating temperatures that were optimized off-line.

5.2.3 Classification performance vs. SNR

In the second experiment, we compared the two methods as a function of measurement noise. As before, we generated a training dataset containing 125 samples using the simulated sensor model, and trained the GMM sensor models. We then used SFS to identify the best

feature subsets of cardinality $f = 5$. Following the previous experiment, the AS algorithm was stopped sensing when either 5 observations were obtained or $H(\theta_t) < 0.65$; AS was also prevented from using the same feature twice. To test the two methods, we generated 11 different test sets, each having additive Gaussian noise with variance ranging from $0.0 \leq \sigma^2 \leq 0.5$ in steps of 0.05. Each test dataset contained 250 test samples, 10 per concentration profile.

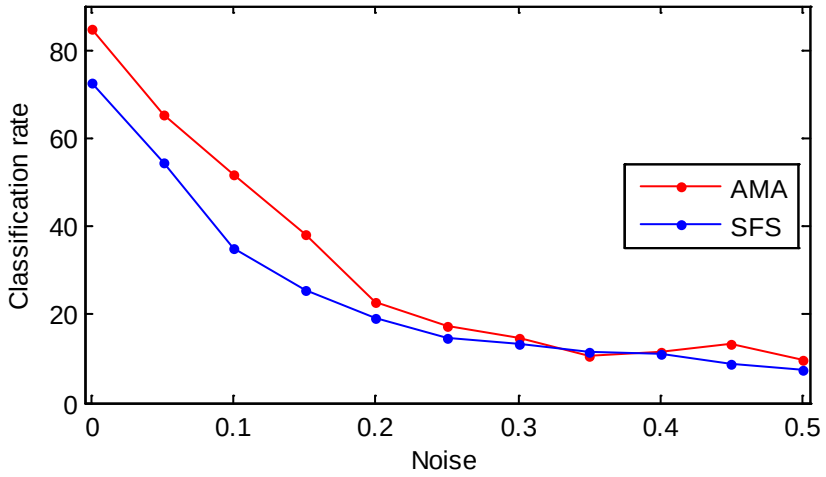

Figure 27 Classification performance of AMA and SFS vs. variance of additive Gaussian noise.

Classification results are summarized in Figure 27. As expected, classification performance for both methods degrades with increasing levels of noise in the test data. However, in the noise range $0.0 \leq \sigma^2 \leq 0.3$, AMA consistently outperforms SFS. This is because AMA selects features at measurement time, which allows it to adapt to noise, whereas SFS uses a pre-specified set of temperatures that was computed off-line using noise-free training data.

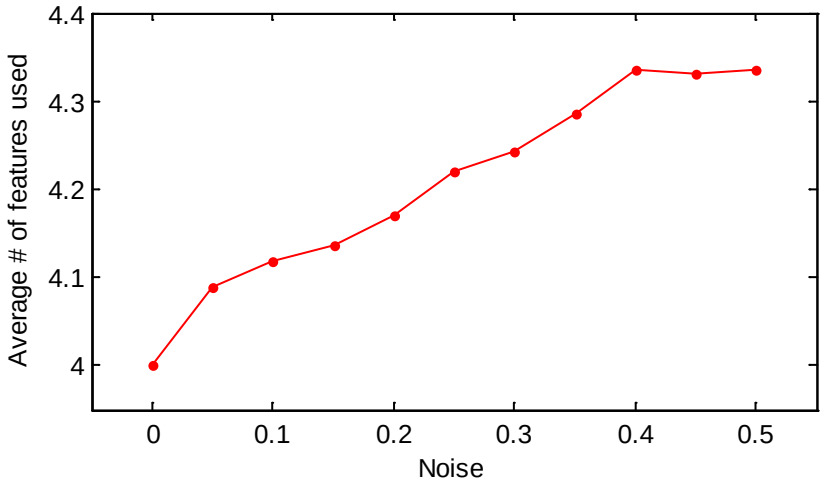

Figure 28 Average number of features used by AMA vs. variance of the additive Gaussian noise.

The average number of features used by AMA (as shown in Figure 28) increase with decreasing SNR because, the increasing levels of noise do not allow the beliefs to reach the certainty levels required for AMA to stop the sensing process. Therefore, AMA acquires more features to 'counter' the noise, whereas SFS does not have this capability. In addition, with increasing levels of noise, the observations are more likely to fall in areas where the class distributions overlap. In such situations, the beliefs associated with these overlapping classes would have similar values. Then, for the subsequent sensing steps, AMA will give preference to features that can strongly distinguish between the 'overlapping classes'. In contrast, SFS works with a predefined sequence of features. At higher levels of measurement noise ($0.35 \leq \sigma^2 \leq 0.5$) both methods perform comparably, indicating that at these settings the amount of noise in the data prevents AMA from making accurate predictions.

5.3 Experimental validation

We also validated the AMA algorithm experimentally using a commercial MOX sensor (TGS 2620) with six binary mixtures of xylene and ethyl-alcohol (EtOH).

5.3.1 Experimental setup

For the experimental validation presented here, we used the apparatus described in APPENDIX A. However, we reorganized this delivery system (with the diluters connected in parallel) so that we could collect sensor responses to binary mixtures. A schematic of the modified experimental setup is shown in Figure 29.

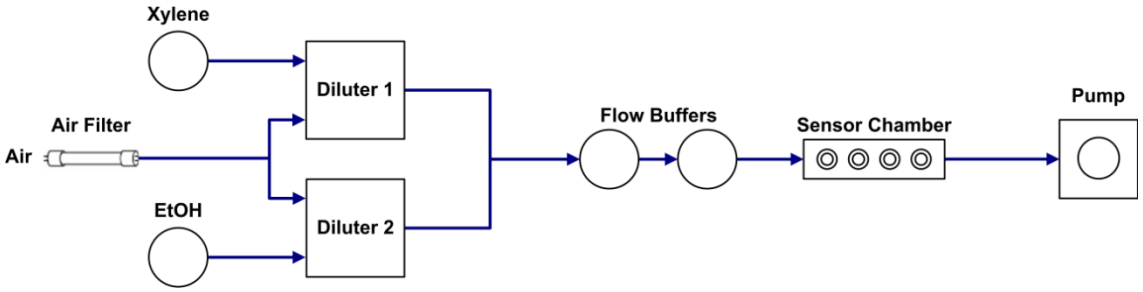

Figure 29 A schematic of the delivery system for generating binary mixtures.

We conducted a preliminary study (similar to the one described in section 4.3.1) to determine a suitable range for the concentrations of the chemicals and sensor heater voltage; our goal was to ensure the quantification problem was not trivial so that temperature modulation would be needed to discriminate the mixtures. The resulting binary mixtures are summarized in Table 5.

Table 5 Concentrations of the two chemicals in the mixtures, expressed in volume/volume.

Mixture #	1	2	3	4	5	6
Xylene concentration	5%	10%	15%	5%	10%	15%
EtOH concentration	5%	5%	5%	15%	15%	15%

5.3.2 Data

The sensor was operated at 10 different heater voltages ranging from $4.5 \leq V_H \leq 7.5$ in steps of 0.33V. At each operating voltage, we pulsed the sensor for duration of 30 seconds. Between consecutive pulses, the sensor was reset to a baseline voltage (0V), also for duration of 30 seconds. This form of temperature programming helped reduce variance in the sensor response due to thermal dynamics.

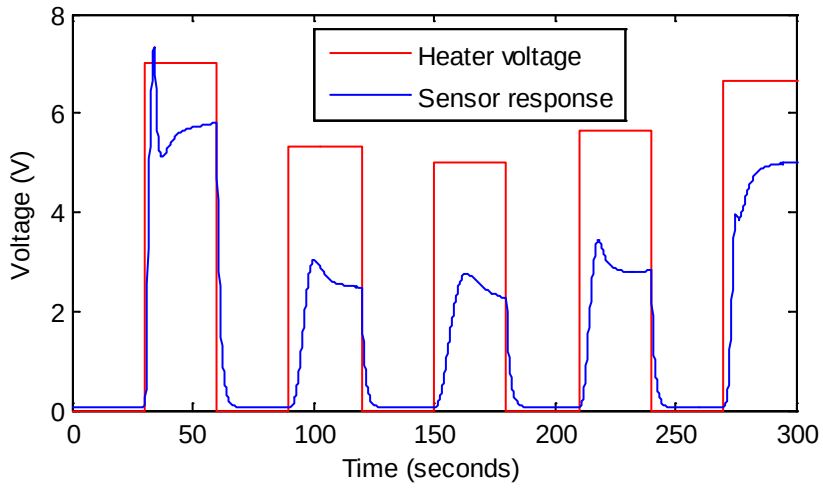

Figure 30 Transient sensor response to a mixture of 5% Xylene and 5% EtOH. The sensor was driven with a sequence of 5 voltage pulses, each 30 seconds long.

Figure 30 shows a sample sensor response to a mixture of 5% xylene and 5% EtOH for a random sequence of five voltage pulses. For each binary mixture, we operated the sensor with 12 random sequences, each sequence containing the 10 operating voltages; this resulted in a dataset with 72 samples (6 mixtures × 12 sequences). The mixtures were presented in a randomized order to avoid systematic errors.

We used Principal Component Analysis (PCA) to extract features from the sensor transients. Namely, for each of the 10 voltage settings and for each sensor, we collected the transient responses to all chemicals and then applied PCA to obtain the loadings (eigenvectors) and scores. Figure 31 (a) shows 30 sensor transient responses to the six binary mixtures (5 transients per mixture) to a 30-second 4.83V pulse; Figure 31 (b) shows the first three loadings. This process is repeated for each sensor and voltage setting to obtain the corresponding loadings and scores. In all cases, the first three principal components were sufficient to capture more than 99% of the variance. Thus, PCA allowed us to compress the transient response down to three features (i.e., the PCA scores). Then, we used three one-dimensional GMMs (one per principal component) to create the probabilistic sensor models; details are available in section 4.1.1. We chose to use three 1-D GMMs as opposed to using 3-D GMMs due to the limited number of training samples available. During the testing stage, when the sensor is driven with a particular pulse, the resulting transient response is multiplied with the PCA loadings corresponding to the same pulse. The resulting scores are treated as the observations.

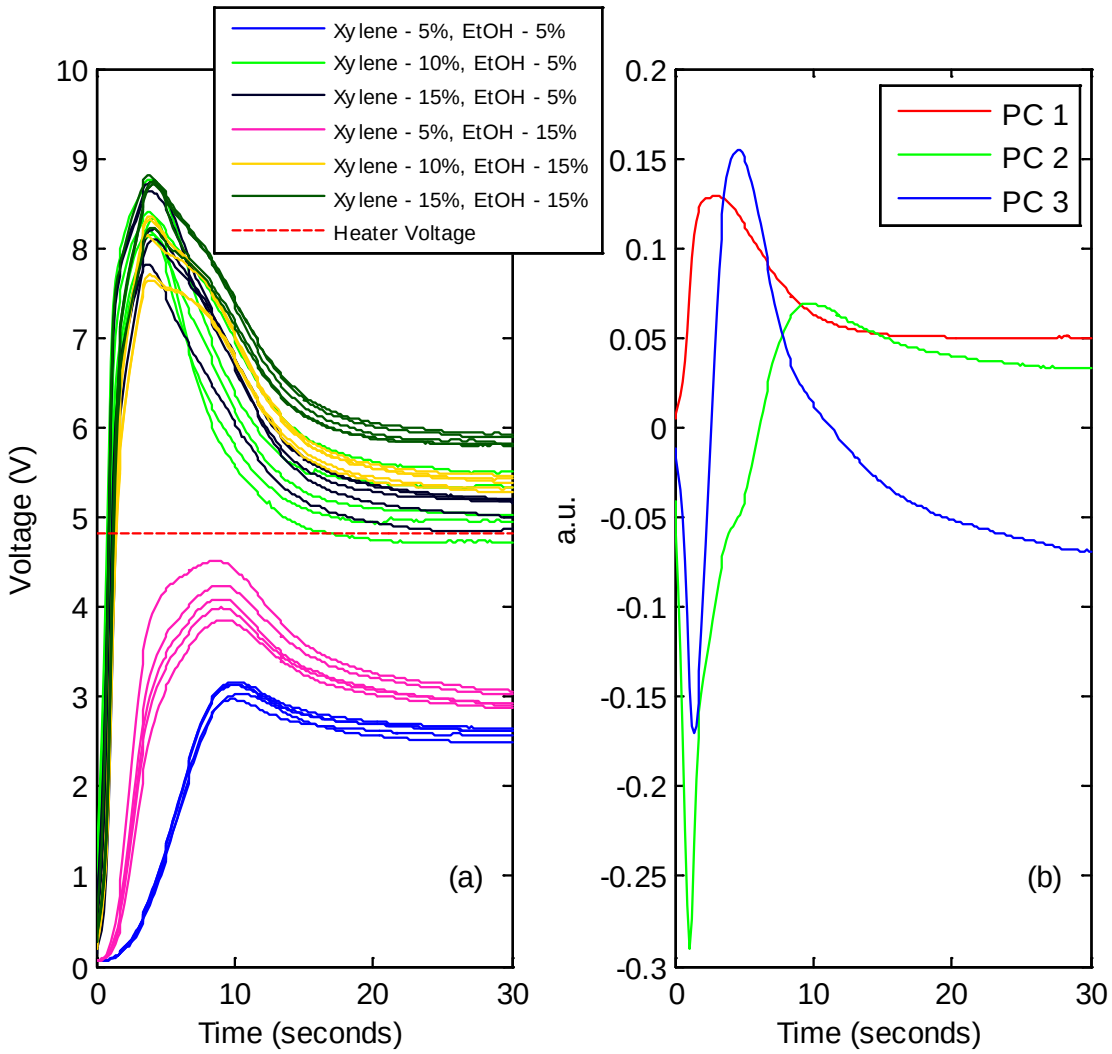

Figure 31 (a) Sensor transient to the six gas mixtures (5 repetitions per mixture) in response to a 4.83V pulse in heater voltage. This pulse was preceded and succeeded by a 0V pulse of 30-second duration. (b) The first three principal components extracted from the transients in (a).

5.3.3 Offline testing

Following procedures described in section 5.2, we compared the performance of AMA against sequential forward selection (SFS). To avoid over-fitting, we used a naïve Bayes classifier wrapped in a 10-fold cross-validation loop as the objective function for SFS. For each

fold, we randomly divided the sensor data (72 samples) into two subsets: a training dataset containing 30 samples (5 per mixture), and a test dataset containing 42 samples (7 per mixture). Then, we ran SFS on the training datasets to generate optimal feature subsets of cardinalities $2 \leq f \leq 7$. For each of these subsets, we estimated classification performance on the corresponding test datasets. We used the same 10-fold cross-validation loop to estimate the performance of AMA. Namely, we trained GMM sensor models on each of the 10 training sets and estimated the performance of AMA on the corresponding test sets. To ensure a fair comparison between the two methods, the AMA algorithm was stopped when either f observations were obtained or $H(\theta_t) < 0.1$[9]; as earlier, AMA was not allowed to choose the same feature twice.

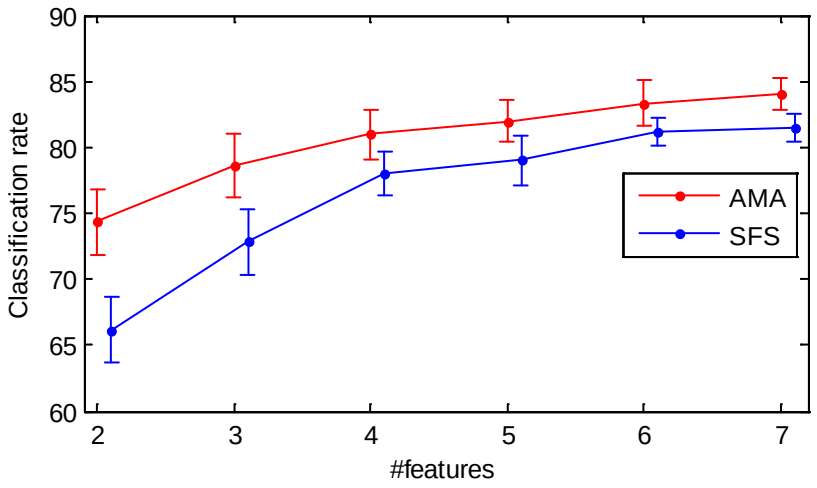

Figure 32 Classification performance of AMA and SFS as a function of number of features used.

Results are summarized in Figure 32. Classification performance for both methods improved with increasing number of features. However, AMA consistently outperformed SFS

[9] The value of 0.1 for ψ reflects a situation with 90% certainty in the classification.

regardless of feature set size. The disparity between the two methods is more evident at lower cardinalities; as the number of features increases, there is a greater overlap between the cumulative feature sets selected by the two methods. These results are consistent with those obtained on simulated data (see 5.2.2).

We calculated the average number of pulses used by AMA for all mixtures (across all rounds of cross-validation and for $f = 7$). The results are summarized in Table 6. For mixtures #1 and #4, AMA requires an average of 2 pulses to identify the samples. Interestingly, these are the two mixtures with 5% Xylene. For mixture #6 (15% both chemicals), AMA requires an average of 3 pulses to identify the sample. For the remaining mixtures (#2, #4, and #5), AMA uses around 6 to 7 pulses. The low standard deviations in Table 6 suggest that the average number of pulses used by AMA is very consistent with the composition of the test samples.

Table 6 The average number of pulses used by AMA and the corresponding deviations (at $f = 7$) for the six mixtures.

Mixture	1	2	3	4	5	6
Average # of pulses used	2.12	6.36	6.34	2.04	6.67	2.88
Standard deviation	0.16	0.53	0.38	0.08	0.24	0.68

To illustrate the adaptive nature of AMA, we also analyzed the probability that each pulse is selected as a function of the sensing step. Results for the case $f = 7$ (first round of cross-validation) are summarized in Figure 33. In the first sensing step, AMA always selects the same pulse ($5.17\ V$), regardless of the test sample; since we assume a uniform belief distribution at $t = 0$, the selection in the first step is based solely on the sensor models. The same trend continues in the second sensing step where AMA selects the same pulse (7.17 V) for all test

samples. At the third sensing step, however, AMA selects the pulse (5.5 V) for mixtures #2, #3, and #5, but selects a different pulse (7.5 V) for mixture #6. We observe similar behavior in the fourth sensing step: AMA selects the pulse (4.5 V) for mixtures #2, #3, and #5, but the pulse (6.2 V) for mixture #6. In the latter sensing steps ($f \geq 5$), the sequences selected for mixtures #3 and #6 tend to converge to a single temperature, but for mixtures #2 and #5 AMA select from among several of the higher temperatures. Thus, the results from Table 6 and Figure 33 clearly illustrate the adaptive nature of AMA, which allows it to select pulse sequences that are highly dependent on the identity of each sample.

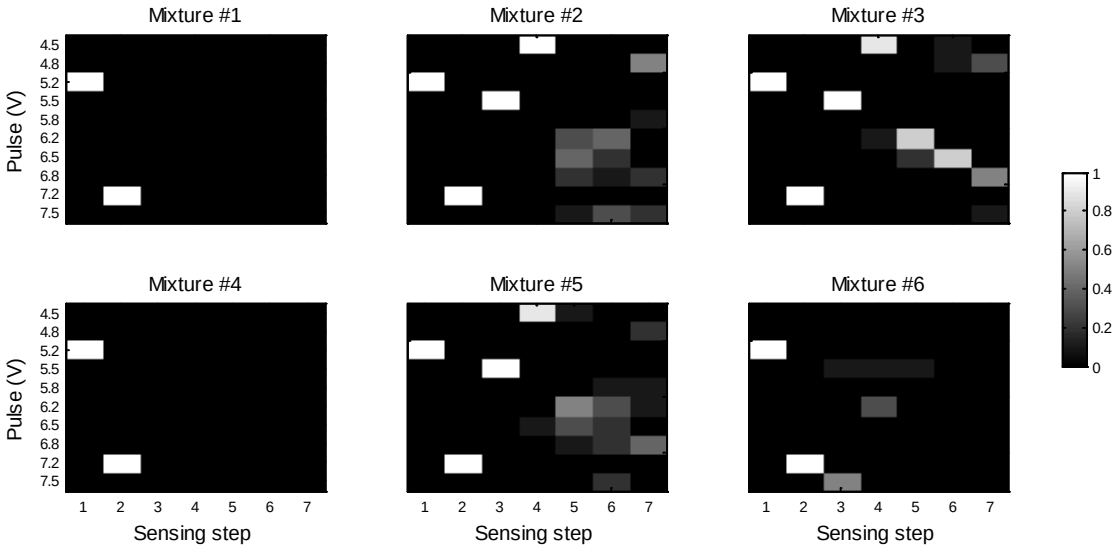

Figure 33 Probability of selecting each temperature pulse vs. sensing steps for each mixture.

5.3.4 Classification performance vs. pulse duration

We also investigated whether shorter pulse durations would affect the classification performance of the active sensing algorithm. For this purpose, we evaluated AMA using only the first 15 and 25 seconds of the transients as well as the entire 30-second transient. Following

procedures described in section 5.3.2, for each transient duration we first applied PCA to extract features, then used GMMs to create the sensor models, and finally estimated classification performance of AMA using 10-fold cross-validation at different cardinalities f.

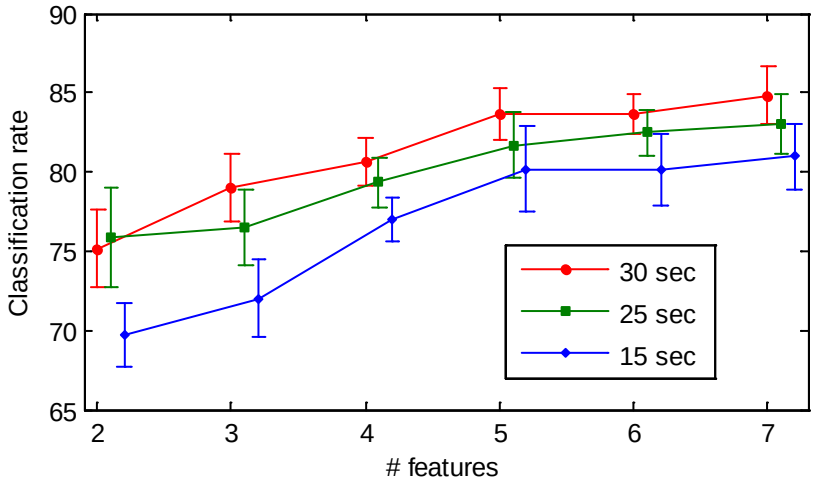

Figure 34 Classification performance of AMA vs. number of features for three pulse durations – 15, 25, and 30 seconds. For visualization purposes, the curves corresponding to 15 and 25 second pulses have been moved slightly along the x-axis.

The results are summarized in Figure 34. Classification rates are consistently higher for 30s pulses than for 15s or 25s pulses, with the only exception of $f = 2$, where 30s and 25s pulses are comparable. The 30s pulses improve the average classification performance by 1.3 and 4.5 percentage points relative to 25s and 15s pulses, respectively, which indicates that the first half of the sensor transients contains most of the discriminatory information. This result is consistent with Figure 31 (a), which shows that the sensor often converges towards its steady-states after 15 seconds, rendering the latter parts of the transients relatively less informative. According to Figure 34, differences between the short (15s) and long (30s) pulses are more marked with fewer

sensing actions (2 or 3 temperatures), which suggests that accuracy can be improved by increasing either the number of sensing actions or the duration of the temperature pulses.

5.3.5 Online testing

Finally, we validated AMA in an online fashion to establish proof-of-concept. Based on the experimental protocols described in section 5.3.2, we collected a dataset of temperature modulated responses to four mixtures of ammonia and denatured alcohol. The four mixtures were: (1) 5% ammonia and 5% denatured alcohol, (2) 5% ammonia and 10% denatured alcohol, (3) 10% ammonia and 5% denatured alcohol, and (4) 10% ammonia and 10% denatured alcohol. Once again, the sensor was operated at 10 different heater voltages ranging from $4.5 \leq V_H \leq 7.5$ in steps of 0.33V. At each operating voltage, we pulsed the sensor for 30 seconds, and reset its temperature to a baseline voltage (0V) for 20 seconds between pulses. For each binary mixture, we operated the sensor with 10 random sequences of the operating voltages resulting in a dataset with 40 samples (4 mixtures × 10 sequences). Using this data, we trained the GMM sensor models as described in section 5.1.1.

We tested AMA 60 times (15 times per mixture) in an online fashion with $f = 10$ sensing steps. We followed similar experimental protocols as described in section 4.3.4. Namely, for each test case, the sensor's heater voltage was first set to 0V, and then the chemical was introduced into the sensor chamber for 30 seconds, and then AMA was commenced to drive the sensor. AMA selected voltage pulses one after the other until one of the following two stopping criteria was met: (1) 10 pulses were used, or (2) $H(\theta_t) < 0.1$. Once again, we tracked all the intermediary belief distributions that were generated by AMA to estimate its classification performance for $f = 1$ to $f = 9$. After each test case, the sensor chamber was flushed with desiccated air for two minutes. Then, as described in section 4.3.4, we also tested the performance of SFS on the data obtained from the test cases of AC. Finally, for completeness,

following the procedures of section 4.3.4 we also evaluated the performance of AMA in an offline fashion.

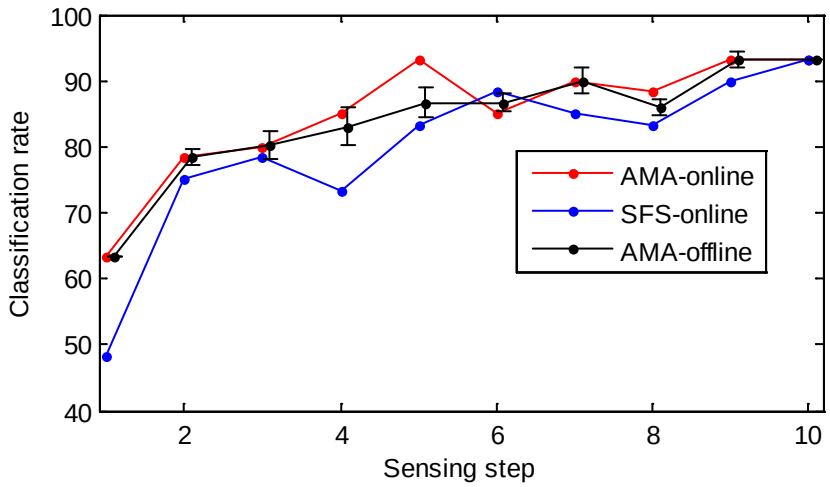

Figure 35 Classification performance of AMA and SFS for online testing vs. number of sensing steps.

The results are summarized in Figure 35. The classification performance of AMA increases from 63.3% at $f = 1$ to 93.3% at $f = 10$. Similarly, the average classification rates of SFS increase from 48.5% at $f = 1$ to 93.3% at $f = 10$. AMA outperforms SFS at all values of f expect for $f = 6$, where SFS does 1.6% better. In terms of classification rates, this result is strongly consistent with all the other results presented in the dissertation. However, unlike the results in sections 5.2.2 and 5.3.3 where the classification rates of AMA followed a smooth monotonic rise with increasing number of sensing steps, the variations here were relatively 'uneven'. This is likely due to the lack of cross-validation for AMA and also the comparatively smaller scale of the experiment. On the other hand, AMA-offline shows a relatively smooth rise in performance with the number of sensing steps. To summarize, like the experiment in section 4.3.4, we believe this experiment serves as a proof-of-concept for AMA.

5.4 Discussion

Our formulation is currently limited to discrete concentration spaces, i.e. where the concentration of each mixture component belongs to one of k possible discrete values. We are currently exploring ways in which the formulation could be extended to continuous concentration spaces, i.e. $\theta \in \mathbb{R}^n$. As an example, particle methods [201] can be used to search through a continuous concentration space. In this case, we would maintain a sample of L posterior distributions or particles $\theta_t = \left\{\theta_t^{[1]}, \theta_t^{[2]}, \ldots, \theta_t^{[L]}\right\}$, where each particle is an n dimensional vector representing a hypothesis of what the concentrations may be at time t, and each particle is assigned a weight $\omega_{t+1}^{[l]}$ which is proportional to the observational probability: $\omega_{t+1}^{[l]} \propto p\left(o_t \middle| a_t, \theta_{t-1}^{[l]}\right)$.

The framework presented here could be extended to other types of temperature modulation waveforms. As an example, if the sensor temperature were to be modulated with sinusoidal waveforms [90] instead of pulses, AS could be used to choose the next modulation sinewave (DC offset, amplitude, and frequency) based on information obtained from previous sensing steps. Examples of other temperature waveforms where this framework could be used, include staircases [202], triangular waveforms [203], and voltage spikes [204].

Our current implementation discretizes the observation space to estimate the expected entropy. This discretization step may be circumvented by using closed-form approximations of entropy. As an example, Huber et al. [205] have presented closed-form approximations of the true entropies for Gaussian mixtures, which are the exact form of our sensor models. Alternatively, one may estimate expected entropy via Monte Carlo sampling [206]. Monte Carlo sampling can provide more accurate estimates of differential entropy than discretization, but at a computational expense.

Finally, in its present implementation, the myopic algorithm does not allow the same feature to be measured multiple times. Because the sensor response is partly a function of its history, it is possible that measurements at the same temperature performed at different times could bring complementary information. Note, however, that in our experiments the sensor temperature is reset between consecutive temperature pulses (Figure 30). This removes, to a large extent, history effects on the sensor response, to where measurements of the same temperature at different times are likely to yield the same information. Multiple measurements at the same temperature may still be beneficial, e.g., as a way to increase signal-to-noise ratio [207]. However, the baseline method we used for comparison (SFS) does not allow repeated measures of the same feature, so it would have been at a disadvantage if we had allowed AC to acquire multiple measurements at the same temperature. Leaving aside the experimental considerations, AC can be easily extended to allow the same pulse multiple times.

6. ACTIVE CLASSIFICATION WITH SENSOR ARRAYS

This chapter presents an active sensing method for arrays of tunable chemical sensors. Sensor arrays offer two advantages when compared to individual sensors: they can provide with multiple features simultaneously, and they respond to a wider range of chemicals if the constituent sensors have different selectivity profiles. With active sensing, arrays can be programmed to behave in a collaborative manner, i.e., acquire multiple features that are informative but not redundant, thus reducing the overall sensing time. However, active sensing methods that are developed for single sensors cannot directly be applied to sensor arrays due to the combinatorial explosion in the number of sensing configurations. For example, an N-sensor array (with D configurations per sensor) could be operated at $O(D^N)$ different configurations.

To address this problem, this chapter presents an active sensing algorithm termed Active Belief-weighted Search (ABS) that combines concepts from feature subset selection with that of active sensing to enable real-time adaptation of sensor arrays. Previous research in sensor array optimization has focused on identifying the best subset of sensors for a given application. However, to our knowledge, there has rather been limited work on the problem of optimizing the operating configurations for arrays. The most relevant study was conducted by Pearce et al. [207], where the authors considered the problem of finding the best set of sensor tunings for a particular detection task. In this study, the authors used information theoretic measures and linear algebra analysis to characterize how different sensors, their tunings, and the noise in the measurements affect the overall performance. Similarly, Raman et al. [208] proposed a heuristic to optimize a micro-sensor array for the problem of identifying toxic industrial chemicals in varying ambient conditions and in presence of interferents. The proposed heuristic is a weighted combination of information content, array size, and material costs. The information content is

measured based on Fisher scores (discussed later in section 6.1.1), and the material costs accounts for the different films required to construct the array. More recently, Fonollosa et al. [209] have characterized how the discriminatory information of MOX sensor array varies with operating temperatures of individual sensors. The discriminatory information is measured as the mutual information between odor labels (4 organic compounds) and the array responses (at 94 different operating temperatures). By maximizing the mutual information, the authors were able to find an optimal set of operating temperatures for all sensors in the array.

6.1 Methods

Consider an array 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 $\boldsymbol{\rho} = \langle \rho_1, \rho_2, \dots, \rho_D \rangle$, and all sensors in the array are identical (same sensing materials, fabrication methods, and transduction principles). The sensor array is exposed to a chemical, and the goal is to determine the identity of this chemical from a list of M possible targets $\boldsymbol{\omega} = \langle \omega_1, \omega_2, \dots, \omega_M \rangle$. The array is operated by a centralized controller, which can adjust the configurations of each sensor and collect their responses. The goal of the controller is to find an optimized sequence of T action vectors $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \dots, \boldsymbol{a}_T\}$, where each action vector corresponds to simultaneously operating N sensors (or a subset of them), each sensor at a particular configuration[10].

This sensing problem can be addressed with a passive approach, where feature subset selection is used to identify a subset of the best TN sensing configurations. Then, the TN configurations are organized into T action vectors such that each action vector contains N configurations. During the sensing stage, the sensor array is driven with these T action vectors, irrespective of the test sample. The resulting TN measurements are collected by the controller

[10] For example, the action vector $\langle (s_1, \rho_2), (s_2, \rho_6), (s_3, \rho_5) \rangle$ corresponds to simultaneously operating sensor s_1 at ρ_2, s_2 at ρ_6, and s_3 at ρ_5.

and processed to determine the identity of the sample. However, this approach is likely to have the disadvantages that are inherent to passive sensing, as detailed in chapter 4. Likewise, this problem could also be addressed with a naïve active sensing approach. This approach would search through all possible action vectors at each time step, evaluate their expected information content, and select the best vector. Clearly, this approach will only work for small arrays with few sensor tunings.

The proposed method (ABS) circumvents the combinatorial explosion of a naïve active sensing method, and addresses the disadvantages of a passive sensing method. ABS works in two steps as illustrated in Figure 36. In the first step, the ABS algorithm selects a subset of the best N configurations and organizes them into an action vector for the next sensing step. This step is similar to feature selection because it selects N configurations from D possibilities. Unlike feature selection, however, ABS 'biases' the process to incorporate the current belief distribution. In the second step, the controller operates the sensor array with the chosen action vector, collects responses from all sensors, and updates the belief distribution. This process is continued till the allocated T sensing steps are completed and finally the class label is declared.

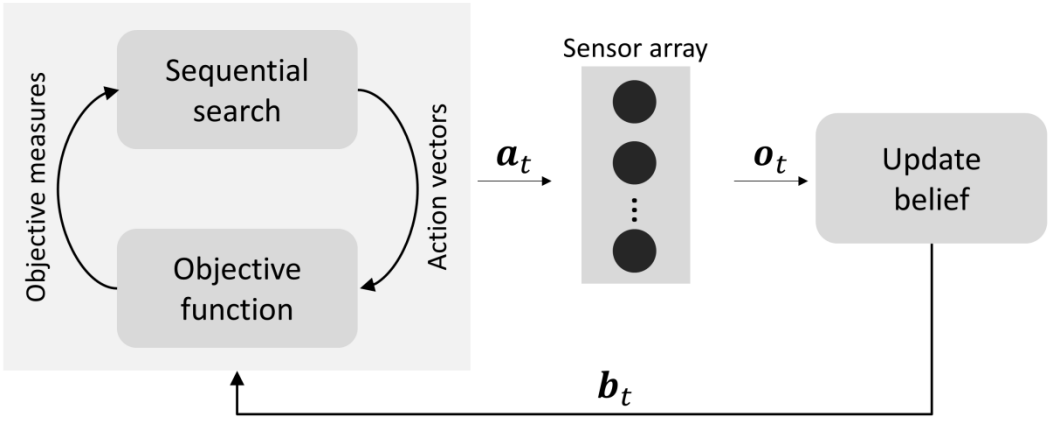

Figure 36 An overview of the ABS method.

As shown in Figure 36, there are two main components to the first step of ABS: a search algorithm and an objective function. The search algorithm generates various candidate action vectors, and the objective function evaluates the action vectors and returns measures of their 'goodness'. In feature subset selection literature, a wide variety of objective functions have been developed based on different criteria such as class separability, probabilistic similarities, correlations, error probabilities, etc. In this chapter, we chose two of the most widely used objective functions [210]: Fisher score (which is based on class separability), and mutual information (which is based on probabilistic similarities), and modified them so that they can incorporate the current belief distribution. A detailed description of all the steps is presented in the following sections.

6.1.1 Weighted Fisher score

The Fisher score has its origins in linear discriminant analysis (LDA); a supervised dimensionality reduction method that seeks to preserve class discriminatory information. Given a subset of features $\boldsymbol{\mathcal{F}} = \rho_1, \rho_2, \dots, \rho_j$, the Fisher score is defined as the determinant of the ratio of the sum of between-class variances to the sum of within-class variances corresponding to the features in $\boldsymbol{\mathcal{F}}$. The main idea of Fisher score is to find a subset of features, such that in the space spanned by these features, the distances between the training instances of different classes are maximized, and the distances between the instances of the same class are minimized.

The objective function proposed here (termed Weighted Fisher Score (WFS)) is a generalized version of Fisher score. Whereas the conventional Fisher score uses sums of variance matrices, WFS uses weighted sums of variance matrices; the weights are obtained from the current belief distribution. The belief distribution $b_t = \{b_t(\omega_1), b_t(\omega_2), \dots, b_t(\omega_M)\}$ is a collection of posterior probabilities associated with each class. $b_t(\omega_i)$ is the posterior of class ω_i at time t given all previous measurements from all sensors $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \dots \boldsymbol{o}_{t-1}\}$, and the

corresponding action vectors $\boldsymbol{A} = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \dots, \boldsymbol{a}_{t-1}\}$. In simpler terms $b_t(\omega_i) = p(\omega_i|\boldsymbol{O}, \boldsymbol{A})$. WFS uses this belief distribution to induce a bias towards features that can better distinguish between classes of higher beliefs. Similarly, it gives lower preference to features that are efficient at discriminating between classes of lower beliefs. WFS is equivalent to the standard Fisher score when the belief distribution is uniform, i.e. $\forall_i\, b_t(\omega_i) = 1/M$. Given the current belief distribution b_t, an action vector, $\boldsymbol{a}_t$, and training data $\boldsymbol{X}$, where $\boldsymbol{X}$[11] is a matrix of D columns and $(N_1 + N_2 + N_3 \dots + N_M)$ rows, where N_i is the number of training instances from class ω_i, then WFS can be calculated in four steps as described below.

In the first step, we calculate M between-class variance matrices: $\{B_1, B_2, \dots, B_M\}$, where B_i is the tensor product (or the outer product) of two vectors. The first vector is the difference between the mean of training samples from class ω_i and the mean of the entire training dataset $\boldsymbol{X}$, and the second vector is the transpose of the first vector. The tensor product of the two vectors results in a symmetrical matrix of size $D \times D$. This matrix is then multiplied with N_i, the number of training samples of class ω_i:

$$B_i = N_i(\mu - \mu_i)(\mu - \mu_i)^T \tag{42}$$

In eq. (42), μ is the mean of the entire dataset $\boldsymbol{X}$:

$$\mu = \frac{1}{(N_1 + N_2 + \cdots + N_M)} \sum_{x \in \boldsymbol{X}} x^T \tag{43}$$

and μ_i is the mean of class ω_i:

[11] In subsequent equations, x is used to represent a single row in matrix $\boldsymbol{X}$.

$$\mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x^T \tag{44}$$

In the second step, we estimate M within-class variance matrices: $\{W_1, W_2, \dots, W_M\}$, where W_i is the covariance matrix associated with class ω_i:

$$W_i = \sum_{x \in \omega_i} (x - \mu_i)^T (x - \mu_i) \tag{45}$$

In the third step, the between-class and within-class variance matrices are summed up to calculate two cumulative variance matrices B and W as:

$$\begin{aligned} B &= \sum_{i=1}^{M} b_t(\omega_i) B_i \\ W &= \sum_{i=1}^{M} b_t(\omega_i) W_i \end{aligned} \tag{46}$$

In the fourth step, from matrices B and W, we extract the rows and columns corresponding to the features in $\boldsymbol{a}_t$ to create two square matrices: $B(\boldsymbol{a}_t)$ and $W(\boldsymbol{a}_t)$, each of size$|\boldsymbol{a}_t| \times |\boldsymbol{a}_t|$. Finally, the objective score (J_F) of action vector $\boldsymbol{a}_t$ is estimated as:

$$J_F(\boldsymbol{a}_t) = \det\big(W(\boldsymbol{a}_t)^{-1} B(\boldsymbol{a}_t)\big) = \frac{\det\big(B(\boldsymbol{a}_t)\big)}{\det\big(W(\boldsymbol{a}_t)\big)} \tag{47}$$

The determinant of square matrix captures the volume of a parallelotope[12] formed using the rows of the matrix. For example, given a matrix $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, then $\det(A) = ad - bc$ is

[12] A parallelotope is generalization of parallelograms for higher-dimensional spaces. Therefore, a parallelogram is 2-parallelotope, and a parallelepiped (which is the equivalent of parallelogram in three dimensional spaces) is a 3- parallelotope.

equal to the area of a parallelogram with vertices: $[0,0], [a,b], [c,d], [a+c, b+d]$. Therefore, the larger the determinant of $B(\boldsymbol{a}_t)$, the larger the volume of the corresponding parallelotope, which means the father apart the classes are from each other. Likewise, the smaller the determinant of $W(\boldsymbol{a}_t)$, the smaller the volume of the parallelotope, which means the distances between the instances of the same classes are smaller.

6.1.2 Dynamic mutual information

Information theoretic metrics are among the most widely used set of objective filters in supervised feature subset selection. These metrics capture probabilistic dependencies among a given set of variables. Unlike correlations, which can only capture linear dependencies, information theoretic measures can also capture non-linear dependencies. Among the various information metrics[13], mutual information is the most commonly used for feature selection. Given a subset of features $\boldsymbol{\mathcal{F}} = \rho_1, \rho_2, \dots, \rho_j$ and the class labels $\boldsymbol{\omega}$, then the ideal information theoretic metric to evaluate this subset would be:

$$I(\boldsymbol{\omega}; \boldsymbol{\mathcal{F}}) = \sum_{i=1}^{M} \int \dots \int p\left(\rho_1, \rho_2, \dots, \rho_j, \omega_i\right) \log\left(\frac{p\left(\rho_1, \rho_2, \dots, \rho_j, \omega_i\right)}{p(\omega_i) p\left(\rho_1, \rho_2, \dots, \rho_j\right)}\right) d\rho_1 \dots d\rho_j \qquad (48)$$

In eq. (48), the term $p\left(\rho_1, \rho_2, \dots, \rho_j, \omega_i\right)$ is the joint probability distribution over $(j+1)$ different variables. Due to the curse of dimensionality, estimating such a high dimensional probability distribution is extremely challenging and requires a very large training dataset. To avoid using eq. (48) directly, researchers have been developing various approximations that ignore the higher-order dependencies [212-217]. These approximations evaluate two sets of metrics from the training data: (a) feature relevance (mutual information of each feature with the

[13] These include Jensen-Shannon divergence, Kullback-Leibler divergence, Renyi divergence, mutual information, etc. Lin's paper [211] describes all these metrics in detail.

class labels), and (b) feature redundancy (mutual information between every pair of features). Both feature relevance and feature redundancy can be estimated using only 2-dimensional probability distributions. The information content of a given feature subset is measured using an equation that combines feature relevance with redundancy (for e.g. the ratio of relevance to redundancy). This equation is at its highest when the relevance is high and redundancy is low; the feature subset that maximizes this equation is considered the optimized solution. The subsets thus selected will contain features that are highly relevant to the class labels, but the features themselves are not redundant.

Of the various objective functions mentioned above, Hall's objective function [217] is one of the most widely used. Hall's objective function evaluates a given feature subset as the ratio of relevance to redundancy, where relevance is defined as the sum of mutual information between each feature and class labels, and redundancy is the square root of the sum of mutual information between every pair of features. The objective function proposed in this section (termed Dynamic Mutual Information (DMI)) modifies the relevance aspect of Hall's objective function to incorporate the current belief distribution. DMI favors configurations that have a high mutual dependence with classes of higher beliefs, and low dependence with other selected configurations. Thus, DMI can screen out configurations that are irrelevant (uncorrelated with classes of high belief) or redundant (highly correlated with one of the selected configurations). When the given belief distribution is uniform, i.e. $\forall_i\ b_t(\omega_i) = 1/M$, DMI is equivalent to Hall's objective function [217]. This idea could be applied to any of the other information theoretic objective functions presented in [212-216]. Given the current belief distribution $b_t = \{b_t(\omega_1), b_t(\omega_2), \dots, b_t(\omega_M)\}$, the training data, and an action vector $\boldsymbol{a}_t$, DMI operates in three steps to evaluate $\boldsymbol{a}_t$.

The first step involves estimating the expected reductions in entropy of the current belief distribution b_t upon observing each one of the features in the action vector $\boldsymbol{a}_t$. Namely, if $\rho_i \in \boldsymbol{a}_t$, then the corresponding expected reduction in entropy is estimated as:

$$U(b_t, \rho_i) \ = 2 \times \left(\frac{H(b_t) - H(b_{t+1}|\rho_i)}{H(b_t) + H(\rho_i)} \right) \tag{49}$$

where $H(b_t)$ is the Shannon entropy of the current belief distribution estimated as: $H(b_t) = -\sum_{\omega_i \in \boldsymbol{\omega}} b_t(\omega_i) \log(b_t(\omega_i))$, and $H(b_{t+1}|\rho_i)$ is the expected entropy of the distribution upon observing ρ_i. The difference between the two entropies is the expected reduction. Estimating $H(b_{t+1}|\rho_i)$ can be interpreted as projecting uncertainty into the immediate feature, and it involves three steps. First, for each possible observation o_k from ρ_i, we calculate the projected belief distribution b_{t+1}. Then, for each projected belief, we compute the entropy $H(b_{t+1}|\rho_i, o_k)$. Finally, we sum all these entropies, weighted by the corresponding observation probabilities, to estimate $H(b_{t+1}|\rho_i)$:

$$H(b_{t+1}|\rho_i) = \sum_{o_k} p(o_k|\rho_i) H(b_{t+1}|\rho_i, o_k) \tag{50}$$

The denominator in eq. (49) acts as a normalization constant to ensure that $U(b_t, \rho_i)$ is in range [0, 1]. $U(b_t, \rho_i)$, called Symmetrical uncertainty [218], is a normalized variant of mutual information. Mutual information has an inherent bias in favor of variables with more values. Symmetrical uncertainty compensates for this bias by normalizing the values to be in the range [0, 1].

The second step involves estimating the symmetrical uncertainties between every pair of configurations (ρ_i, ρ_j) from $\boldsymbol{a}(t)$:

$$U(\rho_i, \rho_j) \ = 2 \times \left(\frac{I(\rho_i; \rho_j)}{H(\rho_i) + H(\rho_j)} \right) \tag{51}$$

In eq. (51), $I(\rho_i, \rho_j)$ is the mutual information between the configurations ρ_i and ρ_j, and $H(\rho_i)$ and $H(\rho_j)$ are the entropies of ρ_i and ρ_j respectively. $I(\rho_i; \rho_j)$ is estimate as:

$$I(\rho_i; \rho_j) = \sum_{o_i \in \rho_i} \sum_{o_j \in \rho_j} p(o_i, o_j) \log \left(\frac{p(o_i, o_j)}{p(o_i) p(o_j)} \right) \tag{52}$$

where $p(o_i, o_j)$ is the probability of jointly obtaining o_i from configuration ρ_i, and o_j from configuration ρ_j. To estimate this joint distribution, the observation spaces of all features are discretized using *k*-means clustering (as described in section 4.1.4) and the joint probability of ρ_i and ρ_j are calculated through combinatorial counting.

In the third step, all the symmetrical uncertainty measures are combined to estimate the DMI (J_M) of action vector $\boldsymbol{a}_t$ as:

$$J_M(\boldsymbol{a}_t) = \frac{\sum_{\rho_i \in \boldsymbol{a}_t} U(b_t, \rho_i)}{\sqrt{|\boldsymbol{a}_t| + \sum_{\rho_i \in \boldsymbol{a}_t} \sum_{\rho_j \in \boldsymbol{a}_t, \rho_i \neq \rho_j} U(\rho_i, \rho_j)}} \tag{53}$$

In the above equation, the numerator captures the relevance of all the features for the classification problem, and the denominator captures the redundancy amongst the selected features. The term $|\boldsymbol{a}_t|$ in the denominator acts like a regularization term.

6.1.3 Illustration

This section presents an illustrative example to demonstrate how WFS and DMI measures change with the belief distribution. Shown in Figure 37, the problem involves choosing the better of two features using WFS or DMI, when provided with training data for a three-class

problem (ω_1, ω_2, and ω_3), and a belief distribution b_t. The training data consists 3000 samples (1000 for each class) generated from three separate 2-D Gaussian distributions, with the following parameters: μ_1 =[1.26 4], μ_2 =[4.73 4], μ_3 = [3 1], and $\Sigma_1 = \Sigma_2 = \Sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Note that the three class-means form an equilateral triangle in the 2-D space, where each side of the triangle has a length of √3 units.

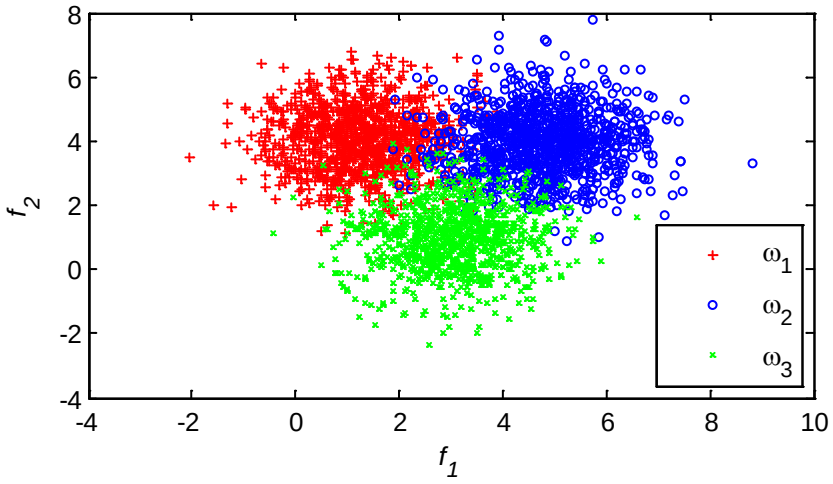

Figure 37 Training data used for the illustration.

Consider a first scenario where the three classes are equally likely: $b_t(\omega_1) = b_t(\omega_2) = b_t(\omega_3)$ =1/3. In this case, the WFSs of the two features are $J_F(f_1) = 2.03$, and $J_F(f_2) = 2.03$, and the DMIs are $J_M(f_1) = J_M(f_2) = 0.33$. This shows that the two features are equally useful towards classifying the sample, no matter what objective function is used. Since the belief distribution is uniform, WFSs are equal to the standard Fisher scores. Likewise, the DMIs of the two features are equal to the mutual information measures.

Now, consider a second scenario where the belief distribution is b_t = [0.4, 0.4, 0.2]. In this case, the WFS values become $J_F(f_1) = 2.45$ and $J_F(f_2) = 1.62$, and the DMI values are

$J_M(f_1) = 0.35$ and $J_M(f_2) = 0.29$. This suggests that feature f_1 is likely to be more useful in classifying the test sample. This is because, since the current sample is more likely to be ω_1 or ω_2, the feature that can better discriminate between those two classes is assigned a higher score. The class means ω_1 and ω_2 are farther apart along f_1 but are the same along f_2.

In the third scenario, there is a greater belief that the sample belongs to either classes ω_2 or ω_3 as compared to ω_1: $b_t = [0.2, 0.4, 0.4]$. With this belief distribution, the WFSs of the two features are $J_F(f_1) = 1.82$ and $J_F(f_2) = 2.25$, and the DMI values are $J_M(f_1) = 0.30$ and $J_M(f_2) = 0.35$. These measures follow the same explanation as above: since the current sample is more likely to be ω_2 or ω_3, f_2 is given a higher score because the class means of ω_2 and ω_3 are farther apart along f_2.

6.1.4 Search algorithm

The second component of the first step of ABS is a search algorithm that generates various candidate action vectors. Based on the literature on feature subset selection [219], these search algorithms can broadly be classified into four categories: (1) exhaustive search, (2) sequential search, (3) branch-and-bound, and (4) meta-heuristic based searches.

Exhaustive search generates every possible action vector. Though exhaustive methods are bound to find the optimal solution, their computational complexity is exponential: $O(D^N)$, and therefore not practical for active sensing. In sequential search, the sensor configurations are added or removed sequentially to maximize an objective function. There are different types of sequential search such as sequential forward search, sequential backward search, sequential floating search, bidirectional search, plus-I minus-R search, and randomized sequential search [220]. Unlike exhaustive search, sequential searches cannot guarantee optimal solutions; however, in practice they find very accurate solutions. More importantly, sequential search has polynomial $O(ND)$ time complexity, which makes them a practical choice for active sensing.

Branch-and-Bound algorithms [221] start from the full set and removes configurations using a depth-first strategy under the monotonicity assumption, which states that the addition of features can only improve the solution. Therefore, nodes whose objective measures are lower than the current best are not explored because the monotonicity assumption ensures that their children will not contain a better solution. Though branch-and-bound algorithms guarantee optimal solutions, they cannot provide bounds on computational complexity: in worst-case scenarios, their complexity could be exponential, thus rendering them impractical for active sensing.

Meta-heuristics [222] are commonly used for combinatorial optimization problems that have solutions in a discrete space. In meta-heuristic based searches, the candidate solutions are improved iteratively by intensification (searching in regions where we are likely to find the optimal solution) or diversification (adding randomness to the solutions to widen the search space). A few examples of meta-heuristic methods include genetic algorithms, tabu search, ant colony optimization, simulated annealing, etc. Meta-heuristic searches do not guarantee an optimal solution and their computational complexity is dependent on the stopping criterion used.

For the experimental work in this dissertation, we chose sequential forward search to construct the action vectors because of its polynomial time computational complexity and its deterministic nature. To check it this choice would be critical, we conducted a few preliminary studies (not reported in this dissertation), which compared the performance of ABS when using sequential forward search against that of exhaustive search. The results showed that, in our application, the classification performance with sequential forward search was comparable to that of exhaustive search.

Table 7 Pseudo-code of sequential forward search for constructing action vectors

Construct_action_vector (N, $\boldsymbol{\rho}$, b_t)

- $\boldsymbol{a}_t = \phi$
- for $i = 1$ to N
    - *cur_obj_measure* = 0
    - for $j = 1$ to $|\boldsymbol{\rho}|$
        - *j_measure* = $J_F(\boldsymbol{a}_t \cup \rho_j)$
        - if *j_measure* > *cur_obj_measure*
            - *cur_obj_measure* = *j_measure*
            - $\rho_{sel} = \rho_j$
        - end
    - end
    - $\boldsymbol{\rho} = \boldsymbol{\rho} - \rho_{sel}$
    - $\boldsymbol{a}_t = \boldsymbol{a}_t \cup \rho_{sel}$
- end
- return $\boldsymbol{a}_t$

In sequential forward search, the action vector is initialized to be a null set: $\boldsymbol{a}_t = \phi$. Then, from the set of available configurations $\boldsymbol{\rho}$, the configuration ρ_j is selected such that when combined with the action vector, maximizes the chosen objective function. For example, if WFS was chosen to be the objective function, then configuration ρ_j is selected such that $J_F(\boldsymbol{a}_t \cup \rho_j)$ is maximized. The search is executed for N iterations to construct an action vector $\boldsymbol{a}_t$ of cardinality N. While doing so, the configurations that were used in the previous sensing steps are tracked to avoid repetition. A pseudo-code of the search process is presented in Table 7.

6.1.5 Sensing and belief update

After constructing the action vector $\boldsymbol{a}_t$, the controller drives the sensor array accordingly. Then it collects the responses from all sensors $\boldsymbol{o}_t$, and updates the beliefs associated with each class using the sequential Bayesian update equation:

$$b_{t+1}(\omega_k) = \frac{b_t(\omega_k)p(\boldsymbol{o}_t|\omega_k, \boldsymbol{a}_t)}{p(\boldsymbol{o}_t)} \tag{54}$$

In eq. (54), $p(\boldsymbol{o}_t|\omega_k, \boldsymbol{a}_t)$ is the probability of obtaining observation vector $\boldsymbol{o}_t$ when the sensor array is exposed to chemical ω_k and driven with action vector $\boldsymbol{a_t}$. The denominator is a normalization constant that ensures the belief distribution sums to one. $p(\boldsymbol{o}_t|\omega_k, \boldsymbol{a}_t)$ is obtained by combining probabilities of responses obtained at each sensor[14]. For example, if $\boldsymbol{a}_t = \langle(s_1, \rho_1), (s_2, \rho_2), \dots, (s_N, \rho_N)\rangle$ and $\boldsymbol{o}_t = \langle o_1, o_2, \dots, o_N\rangle$, then $p\big(\boldsymbol{o}(t)\big|\omega_k, \boldsymbol{a}(t)\big)$ is estimated as:

$$p(\boldsymbol{o}_t|\omega_k, \boldsymbol{a}_t) = \prod_{i=1}^{N} p(o_i|\omega_k, s_i, \rho_i) \tag{55}$$

At $t = 0$, ABS initializes the belief of each class as proportional to the number of training samples $b_0(\omega_k) = \frac{N_k}{(N_1+N_2+\cdots+N_M)}$. ABS brings the sensing process to a halt when the allocated T sensing steps are completed. The final class label is declared based on maximum a posteriori (MAP) criterion as:

$$\omega_{out} = \underset{1\le k\le M}{\operatorname{argmax}}\, b_T(\omega_k) \tag{56}$$

The entire ABS algorithm is summarized below in Table 8.

[14] During the training stage, we use Gaussian mixture models (GMMs) [176] to learn the probabilistic distribution of the sensor responses to different chemicals at different operating configurations .

Table 8 Pseudo-code of the ABS algorithm

Step 1: Initialization	• Initialize belief distribution $b_0(\omega_k) = \frac{N_k}{(N_1+N_2+\cdots+N_M)}$ • Initialize set of available configurations $\boldsymbol{\rho}$ • $t = 1$
Step 2: Search	• Construct action vector $\boldsymbol{a}_t$
Step 3: Sensing	• Apply $\boldsymbol{a}_t$ to obtain observation vector $\boldsymbol{o}_t$
Step 4: Update belief	• $b_{t+1}(\omega_k) = \frac{b_t(\omega_k)p(\boldsymbol{o}_t\|\omega_k,\boldsymbol{a}_t)}{p(\boldsymbol{o}_t)}$ • $t = t + 1$

Step 5: Termination	• If $t < T$
	  o Go to step 2
	• Else
	  o Classify the sample using eq. (56).

6.1.6 Computational complexity

The computational complexity of the ABS algorithm depends on the choice of the objective function. When WFS is used as the objective function, the run-time computational cost lies in: (a) finding weighted sum of scatter matrices, and (b) calculating the determinants. The scatter matrices do not have to be calculated at every sensing step. Instead, they could all be calculated before sensing process beings to reduce a significant portion of the computational costs. At each sensing step, the forward search goes through N iterations. At each iteration, it searches through a maximum of D configurations, and for each configuration it computes the sum of M square matrices of size $N \times N$ and then it calculates the determinant of the matrix. The

sum can be computed in $O(N^2M)$ steps and the determinant can be computed in $O(N^3)$ steps using the LU decomposition algorithm. Therefore, the worst-case computation complexity of ABS, for any sensing step, is the higher of $O(N^4D)$ and $O(N^3MD)$.

When DMI is used as the objective function, the run-time computational cost comes from estimating the expected information gains in eq. (50). The pair-wise symmetrical uncertainties used in eq. (51) are completely independent of the current belief distributions; therefore, these values can be estimated before the sensing process begins, thus reducing a significant portion of the computational cost. As with WFS, the forward search would go through N iterations. For every iteration, it searches through a maximum of D configurations, and for each configuration, it calculates the information gain. Also, for every iteration, it calculates the sum of $O(N^2)$ elements to estimate denominator in eq. (53). The complexity of estimating information gains depends on the number of discrete values into which the observation spaces have been discretized. If each configuration's space was discretized into r discrete values, then estimating information gain takes $O(rM)$ steps. Therefore, the worst-case computation complexity of ABS, for any sensing step, is the higher of $O(N^3D)$ and $O(NDrM)$.

6.1.7 Heterogeneous sensor arrays

The ABS method presented above works under the assumption that all the sensors in the array are identical. In this section, we discuss how ABS method can be modified to handle arrays with non-identical sensors. With such arrays, the problem statement would have to be modified to include two new details: (1) each sensor s_i work at D_i different configuration given as $\boldsymbol{\rho}_i = \{\rho_{i,1}, \rho_{i,2}, \dots, \rho_{i,D_i}\}$, (2) the configuration spaces of the sensors are independent of each other, which means that operating sensor s_1 at $\rho_{1,1}$ is different form operating sensor s_2 at $\rho_{2,1}$. The goal of the controller is still to find an optimal sequence of T action vectors $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \dots, \boldsymbol{a}_T\}$,

where each action vector corresponds to simultaneously operating N sensors (or a subset of them), each sensor at its own particular configuration.

ABS can be extended to heterogeneous arrays simply by modifying the sequential search in Table 7. Namely, when selecting an action vector of cardinality N, the k^{th} configuration would be chosen from the configurations space of sensor s_k, which is $\boldsymbol{\rho}_k$. The resulting action vector would therefore have one configuration for each sensor in the array. However, this approach has one main disadvantage: the ordering of the sensors could significantly influence the action vectors chosen by ABS. This is an inherent disadvantage of sequential search and therefore would also be prevalent in corresponding passive sensing methods.

6.2 Validation on synthetic data

We validated ABS through a series of experiments. As in chapter 4, the experimental work is divided into two sections based on the type of the data used: synthetic and experimental. The experiments on synthetic data allowed testing the ABS method extensively, whereas the validation on experimental data helped establish proof-of-concept.

We conducted five experiments on synthetic data. The first experiment compares ABS against SFS on an array of homogenous sensors with varying number of sensing steps. The second experiment does the same comparison but with varying SNR levels in the test data. The third experiment compares the performance of ABS with SFS on an array of heterogeneous sensors. The fourth experiment studies how sensor orderings affect the performance of ABS and SFS. In the last experiment, we propose an early stopping criterion for ABS and investigate how this criterion affects the classification performance.

6.2.1 Data

The synthetic dataset used in this chapter is similar to the one described in section 4.2.1. We simulated the response of an FPI sensor array to different compounds using infrared absorption data obtained from NIST chemistry WebBook. We chose 50 compounds from this database that had strong absorption peaks in the range 8-10.5 μm. We used a wider operating range in this chapter (as compared to section 4.2.1) to allow us to test ABS and SFS on sensor arrays with non-overlapping sensing ranges. To simulate the spectral resolution of the FPI sensors, the FTIR spectra were down-sampled to 116 values each. The absorption spectra of the 50 compounds are shown in Figure 38, and their names are provided in APPENDIX B.

Using these spectra, we generated 10 samples for each chemical by adding Gaussian noise of standard deviation $\sigma = 0.05$ at each wavelength. The resulting dataset containing 500 = 50×10 samples was used for training purposes. Similarly, we generated another dataset containing 1000 samples (20 per chemical) for testing purposes. However, the test dataset contained an additive Gaussian noise of standard deviation $\sigma = 0.15$, which is three times that of the training dataset.

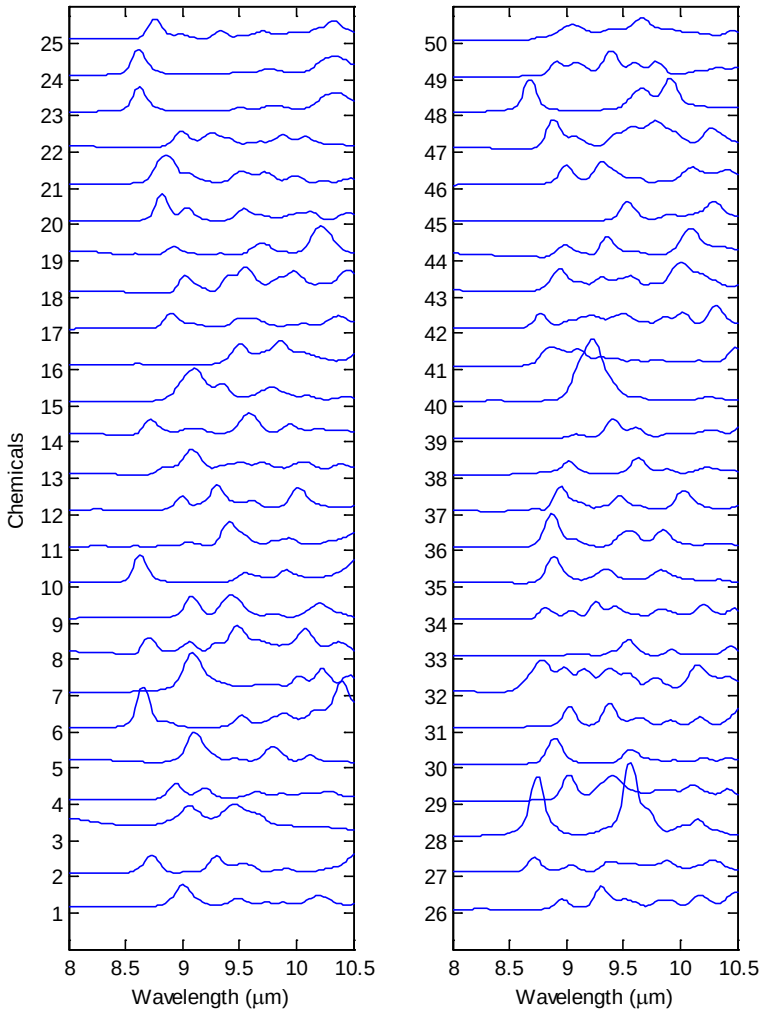

Figure 38 Absorption spectra all compounds as a function of wavelength. For visualization purposes, the spectra are plotted in two columns with an offset along the y-axis.

6.2.2 Comparison with SFS

The first experiment compares the performance of ABS and SFS with varying number of sensing steps. This experiment consists of four steps. In the first step, we applied SFS on the training data (with Fisher score as the objective function) to generate an ordered feature subset

containing 40 features. We refer to this method as SFS-FS, where FS stands for Fisher Score. We then developed 40 naïve Bayes classifiers using the training dataset, where the i^{th} classifier was trained on the first i features that were selected by SFS-FS. For each naïve Bayes classifier, the features were modeled with Gaussian distributions, and the priors initialized uniformly: $\forall_i\ b_0(\omega_i) = 1/50$. These classifiers were used to evaluate the classification performance of SFS-FS for three sensors arrays ($N = 2$, 3, and 4), and 10 sensing steps: $T = 1,2, \ldots, 10$. For example, to evaluate SFS-FS for $N = 3$ and $T = 5$, the classification performance of naïve Bayes classifier trained on the first $TN = 15$ features was used.

In the second step, we repeated the entire process described in the first step, but in this case the features were selected based on Hall's objective function [217]. This method is called SFS-MI, where MI stands for mutual information. For estimating mutual information, each feature was discretized into 10 bins[15] using *k*-means clustering. This allowed us to estimate the class-dependent probability distributions of different sensor configurations, and the joint probability distributions of every pairs of sensor configurations.

In the third step, we estimated the classification performance of the ABS method for three sensor arrays ($N = 2,3$, and 4) with varying number of sensing steps $T = 1$ to $T = 10$. In this step, WFS was used as the objective function. This method will be referred to as ABS-WFS. The features were again modeled using Gaussian distributions. In the last step, we repeated the processes described in the third step but with DMI as the objective measure; this method will be referred to as ABS-DMI.

[15] This value was chosen empirically: we tested the performance of SFS-DMI for various numbers of bins and chose the value that performed the best.

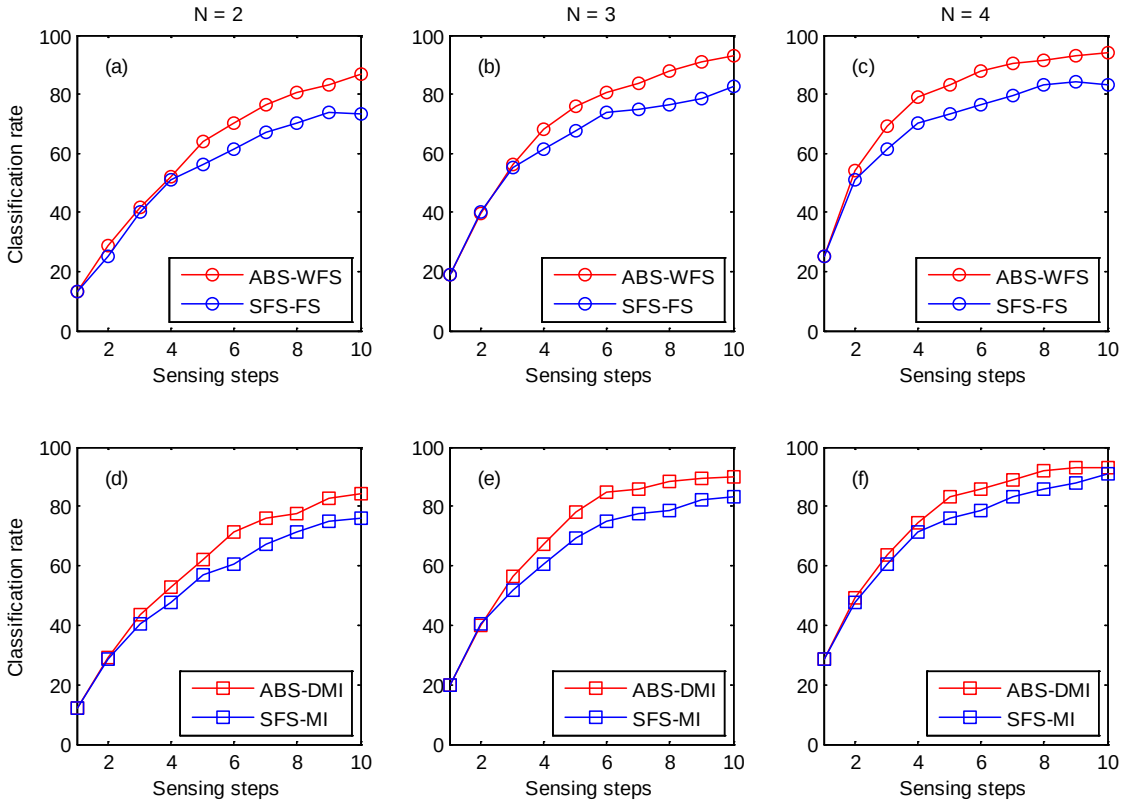

Figure 39 Classification performance of ABS compared to SFS for various number of sensors, sensing steps, and for both objective functions.

The results from these four steps are summarized in Figure 39. For $N = 2$ sensors, the classification rates of all methods increased from 13.2% (ABS-WFS and SFS-FS), and 12% (ABS-DMI and SFS-MI) after one sensing step to 86.8% (ABS-WFS), 73.2% (SFS-FS), 84.4% (ABS-DMI), and 75.8% (SFS-MI) after 10 sensing steps. At $N = 3$ and $N = 4$, the four methods demonstrated similar behavior. At the first sensing step, the active and passive methods choose the same action vector (and therefore achieve the same performance) because the active methods are making their choice based on a uniform belief distribution. The two active methods (ABS-WFS and ABS-MI) consistently outperformed their passive counterparts (SFS-FS and SFS-MI),

except for at $N = 3$ and $T = 2$. The choice of the objective function did not affect the performance significantly. The performance of all four methods (for a given sensing step) improved with the number of sensors in the array. For example, ABS-WFS obtained 86.8% (at $N = 2$ and $T = 10$) but with increasing number of sensors, the classification rates improved to 92.8% (at $N = 3$) and 93.8% (at $N = 4$). The main reason for this improvement is that the methods acquire more features per time step due to the increase in the number of sensors.

6.2.3 Performance vs. SNR

In the second experiment, we tested the performance of ABS and SFS with varying levels of SNR in test data. We generated ten datasets (as described in section 6.2.1) with additive Gaussian noises of standard deviations varying from $\sigma = 0.12$ to $\sigma = 0.3$ in steps of $\Delta\sigma = 0.02$. We then tested the performance of ABS-WFS, ABS-DMI, SFS-FS, and SFS-MI on these ten datasets with $N = 2$ and $T = 10$.

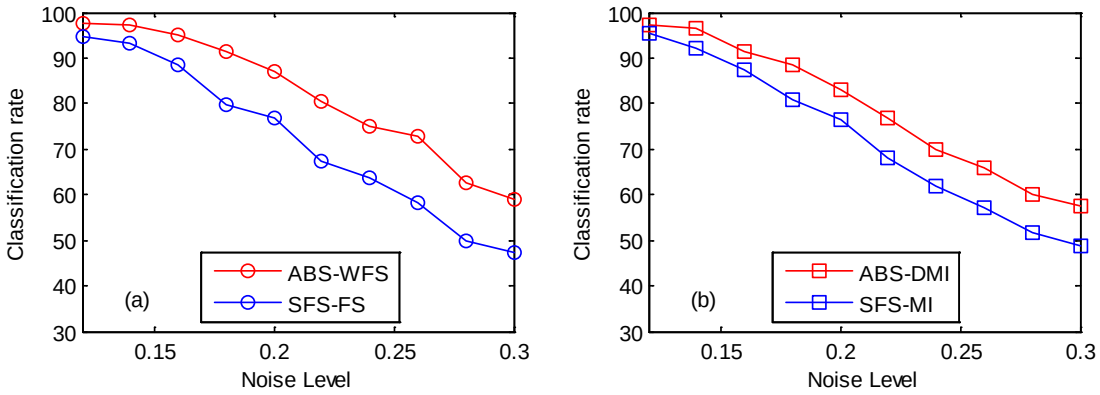

Figure 40 Classification performance of ABS and SFS vs. noise in test data when using (a) Fisher scores, and (b) mutual infromation as the objective functions.

The results of the experiment are summarized in Figure 40. As expected the classification performance of all four methods deteriorated with decreasing SNR. However, the active sensing methods (ABS-WFS and ABS-DMI) outperformed their passive counterparts (SFS-FS and SFS-MI) at all SNRs. As in the previous experiment, the choice of the objective function did not influence the results significantly. However, one interesting observation in this experiment is that, for Fisher scores, the difference between active (ABS-WFS) and passive (SFS-FS) methods increased with the decreasing SNR, whereas, for mutual information, the differences between active (ABS-DMI) and passive (SFS-MI) methods stayed similar with increasing noise. This suggests that the WFS objective function adapts better to noise in test data.

6.2.4 Heterogeneous arrays

In the previous two experiments, we validated ABS on homogenous sensor arrays: they had the same sensing capabilities. In this experiment, we consider an experimental scenario where the sensors are not identical, i.e., they have different sensing ranges. To simulate an array of N FPIs with different sensing ranges, the 116 configurations were sequentially distributed among the sensors. For example, if the array had $N = 4$ sensors, then sensor s_1 had access to the first 29 configurations, s_2 had access to the next 29 configurations, and so on.

For this experiment, we modified the forward search of ABS as described in section 6.1.7. We also modified SFS such that, when selecting features sequentially for an N sensor array, the k^{th} feature would always be chosen for the $(1 + (k - 1) \mathrm{mod} N)^{th}$ sensor, where mod is the modulo operation. Thus, if $N = 2$, the first feature would be selected for s_1, the second feature would be selected for s_2, the third feature would again be selected for s_1, and so on.

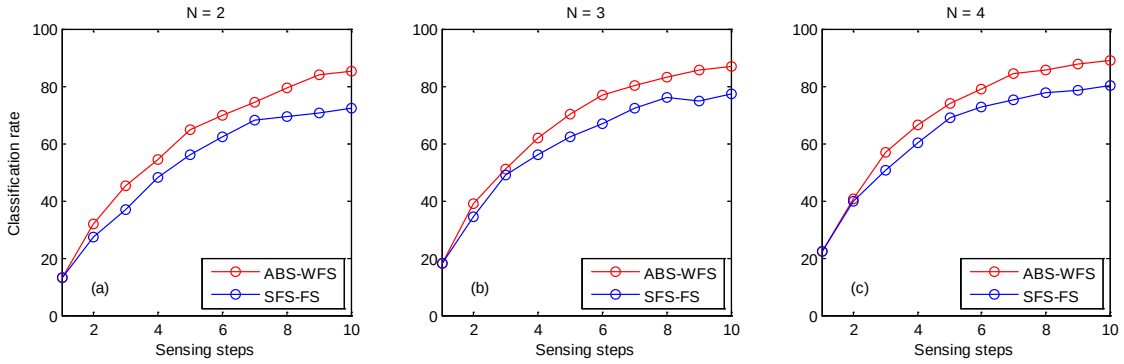

Figure 41 Classification peformance of ABS and SFS vs. number of sensing steps using Fisher scores as the objective function for (a) $N = 2$, (b) $N = 3$, and (c) $N = 4$.

The results of the experiment are summarized in Figure 41 and Figure 42. The classification performance of ABS-WFS and SFS-WFS increase with the number of sensing steps. As in the previous two experiments, the active methods outperform the passive ones. Also, the difference between the active and passive methods increased with the number of sensing steps. These results are consistent with the ones obtained on homogenous sensor arrays (section 6.2.2).

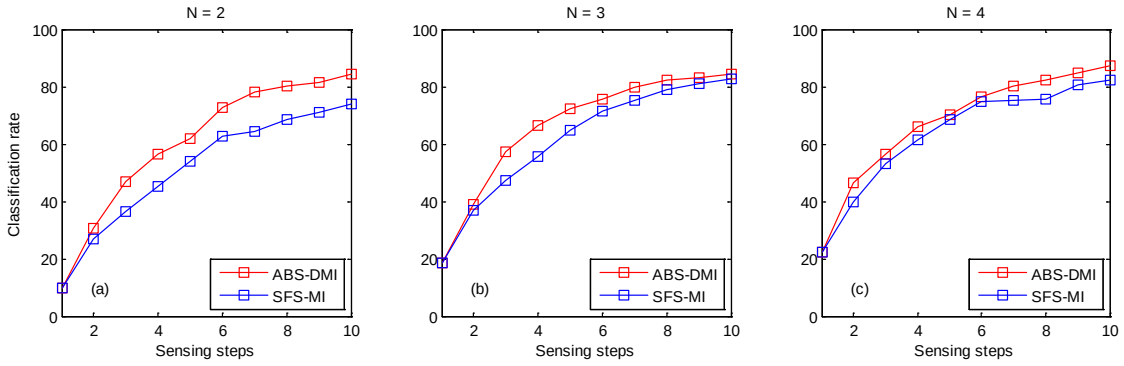

Figure 42 Classification peformance of ABS and SFS vs. number of sensing steps using mutual information as the objective function for (a) $N = 2$, (b) $N = 3$, and (c) $N = 4$.

The results in Figure 42 (with mutual information as the objective function) follow similar trends as in Figure 41. However, the difference between the active and passive methods reduces with increasing number of sensors: for $N = 2$, the average difference (across all sensing steps) between ABS-DMI and SFS-MI is 8.9%; this difference reduced to 4.5% for $N = 3$ and 3.9% for $N = 4$. We believe this is because DMI uses the current belief distribution to bias the feature relevance measures (i.e. MI between features and class labels) but not the redundancy measures (i.e. MI between features themselves). Therefore, with increasing number of sensors, the advantages of DMI seem to be reducing because the increasing redundancy is not biased. If DMI were also to incorporate the belief distribution into the redundancy measures, then it needs to estimate 3-dimensional joint probability distributions of every pair of features and the class labels. This adds to the computational complexity and also creates a need for more training data to ensure that the 3-d distributions are accurately estimated.

6.2.5 Effects of sensor ordering

As discussed in section 6.1.7, one of the drawbacks of ABS is that the ordering of the sensors will influence the action vectors selected. To better understand the effects of sensor ordering on classification performance, we conducted the following experiment with a heterogeneous sensor array of $N = 3$ sensors. We first generated the six possible sensor orderings: $\langle s_1, s_2, s_3 \rangle, \langle s_1, s_3, s_2 \rangle, \langle s_2, s_1, s_3 \rangle, \langle s_2, s_3, s_1 \rangle, \langle s_3, s_1, s_2 \rangle, \langle s_3, s_2, s_1 \rangle$. Then, for each one of the six orderings, we estimated the classification performance of the four methods (ABS-WFS, SFS-FS, ABS-DMI, SFS-MI). To do this, the sequential search (for both ABS and SFS) was modified such that the features would be selected according to the sensor ordering. For example, while constructing an action vector for the 6th ordering $\langle s_3, s_2, s_1 \rangle$, ABS and SFS would always select the first configuration for s_3, then the second configuration for s_2, and finally the third configuration for s_1.

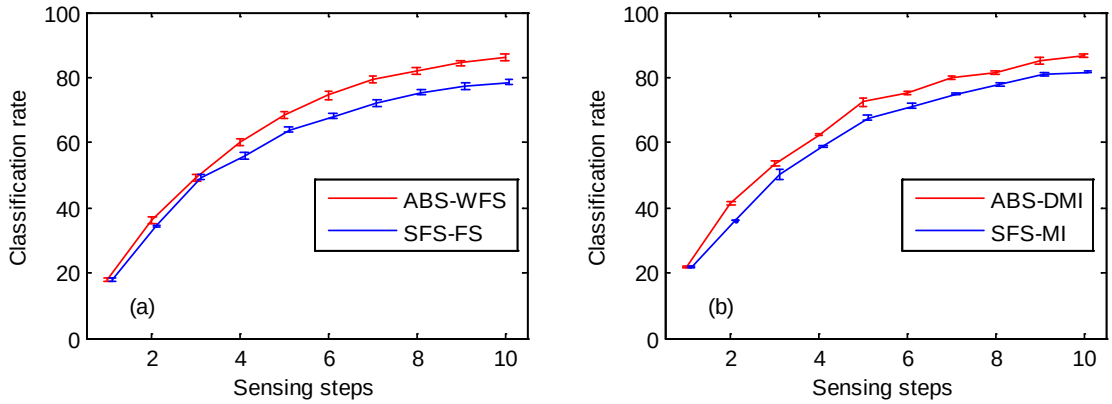

Figure 43 Classification rates of (a) ABS-WFS and SFS-FS, and (b) ABS-DMI and SFS-MI vs. number of sensing steps for $N = 3$ sensors, averaged over all six sensor orderings. To avoid overlap, the curves corresponding to SFS-FS and SFS-MI were shifted slightly along the x-axis.

Figure 43 presents the results in terms of classification rates, averaged over the six sensor orderings. These results are very similar to the ones presented in Figure 41 (b) and Figure 42(b): active sensing methods outperform the passive ones. However, the more important observation is that the standard deviations are very low (in order of 1-2% for all settings) for both active and passive sensing methods. This suggests that the sensor orderings had rather minimal effects on classification performance of all methods.

6.2.6 Stopping criterion

The ABS algorithm brings the sensing process to a halt when the allocated T sensing steps are completed. Based on the belief distribution b_t, the algorithm can make a decision about whether to continue acquiring new observations or to stop the sensing process. We propose one such stopping criterion that is based on the changes in the belief distribution as compared to the previous sensing steps. Given the current belief b_t and the previous belief b_{t-1}, we calculate the Manhattan distance between the two distributions and stop the sensing process when the distance is smaller than a predetermined threshold ψ. This criterion encourages ABS to halt when the

belief distribution starts converging, and thus prevents the future acquisition and computational costs. We tested this criterion on both ABS-WFS and ABS-DMI for $N = 2$ and $T = 10$: ABS would stop when either: (a) the threshold is reached, or (b) 10 sensing steps have been completed. The threshold was varied uniformly from $\psi = 2$ to $\psi = 0$ in ten steps.

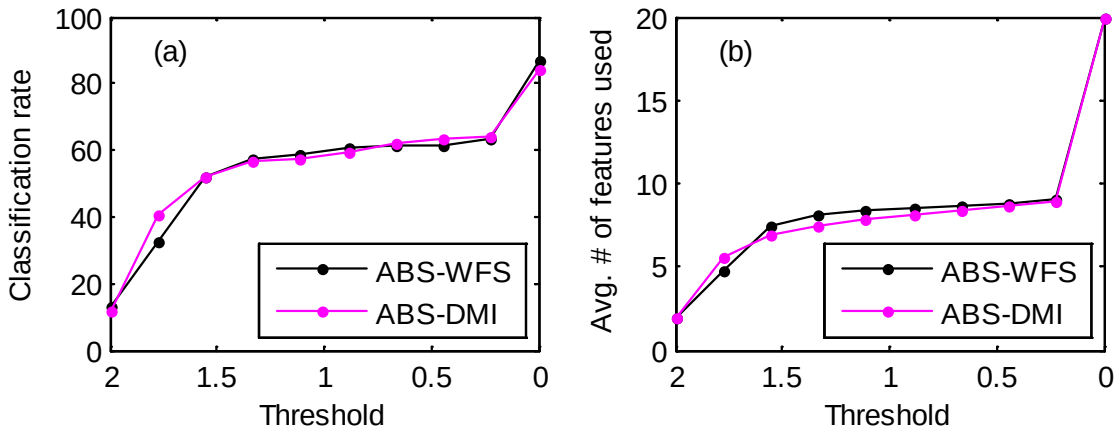

Figure 44 (a) Classification rates of ABS-WFS and ABS-DMI with varying stopping criterion, and (b) the corresponding average number of features acquired.

The results are summarized in Figure 44 in terms of classification rates and average number of features acquired. When $\psi = 2$, which is the maximum possible Manhattan distance between two belief distributions, ABS stops after the first sensing step. Therefore, the average number of features acquired is 2, and the classification rates are 12% (for ABS-DMI) and 13.2% (for ABS-WFS). As the threshold reduces, ABS needs to acquire more features to achieve more stable belief distributions. This is reflected in the corresponding classification rates. When $\psi = 0$, ABS goes through the entire ten sensing steps for every test case. As in the previous experiments, the choice of the objective function did not influence the classification rates significantly. However, for $\psi \in (0.2,\ 1.5)$, ABS-DMI used fewer features than ABS-WFS

(though the difference is not very large) suggesting that the belief distributions might converge faster with DMI than with WFS.

6.3 Experimental validation

We experimentally validated ABS on an array of commercially available MOX gas sensors. The array consisted of four TGS sensors (TGS 2620, TGS 2602, TGS 2610, and TGS 2611) manufactured by Figaro. The four sensors have similar working principles but different sensitivities: TGS 2620 is sensitive to organic vapors, TGS 2602 is sensitive to ammonia and hydrogen-sulphide, TGS 2610 is sensitive to hydrocarbons (propane and butane), and TGS 2611 is sensitive to methane. However, as mentioned in section 2.4, the four sensors are cross-sensitive.

We conducted two experiments on this sensor array. The first experiment compared the performance ABS and SFS in an offline fashion. The second experiment validated the methods in an online manner to establish a proof-of-concept. Over the course of these experiments, the TGS 2611 sensor was damaged; therefore the corresponding data was dropped.

6.3.1 Data

We collected a dataset of temperature-modulated responses of the MOX sensor to five household chemicals: mineral spirits, acetone, ammonia, denatured alcohol, and isopropyl alcohol. The experimental procedures used in this section are similar to the ones described in section 4.3.1; except, here we recorded the responses from all sensors. The sensors were operated at 10 different heater voltages ranging from $3 \leq V_H \leq 7.5$ in steps of 0.5 V. For each chemical, the sensor array was driven with 20 random sequences, each sensor with a different sequence and each sequence contained the 10 operating voltages; this resulted in a dataset with 100 samples (5 chemicals × 20 sequences). Figure 45 shows the response of the three MOX

sensors when exposed to mineral spirits and driven with a sequence of five voltage pulses. In this example, the all sensors in the array were driven with the same sequence.

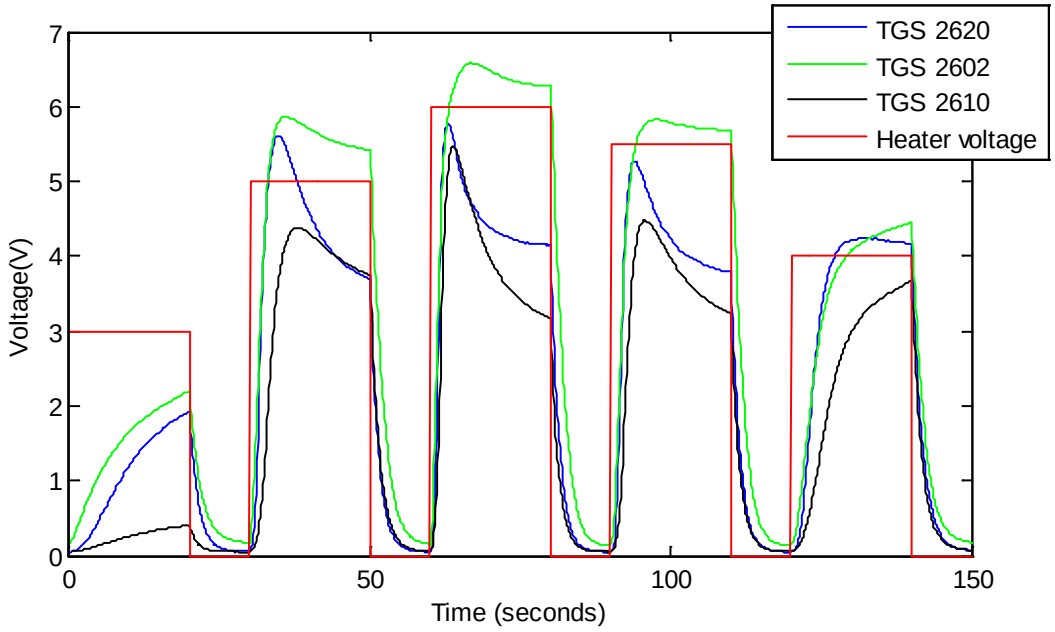

Figure 45 Transient sensor responses to mineral spirits. The sensors were driven with a sequence of 5 voltage pulses, each 20 seconds long.

Following the procedures described in section 5.3.2, we used Principal Component Analysis (PCA) to extract features from the sensor transients. Figure 46 (a) shows 25 transient responses of the TGS 2620 sensor to the five chemicals (5 transients per chemical) when driven with a 20-second 3.5V pulse; Figure 46 (b) shows the first three loadings. Then, we used one-dimensional GMMs (one per principal component) to create the probabilistic sensor models; details are available in section 4.1.1.

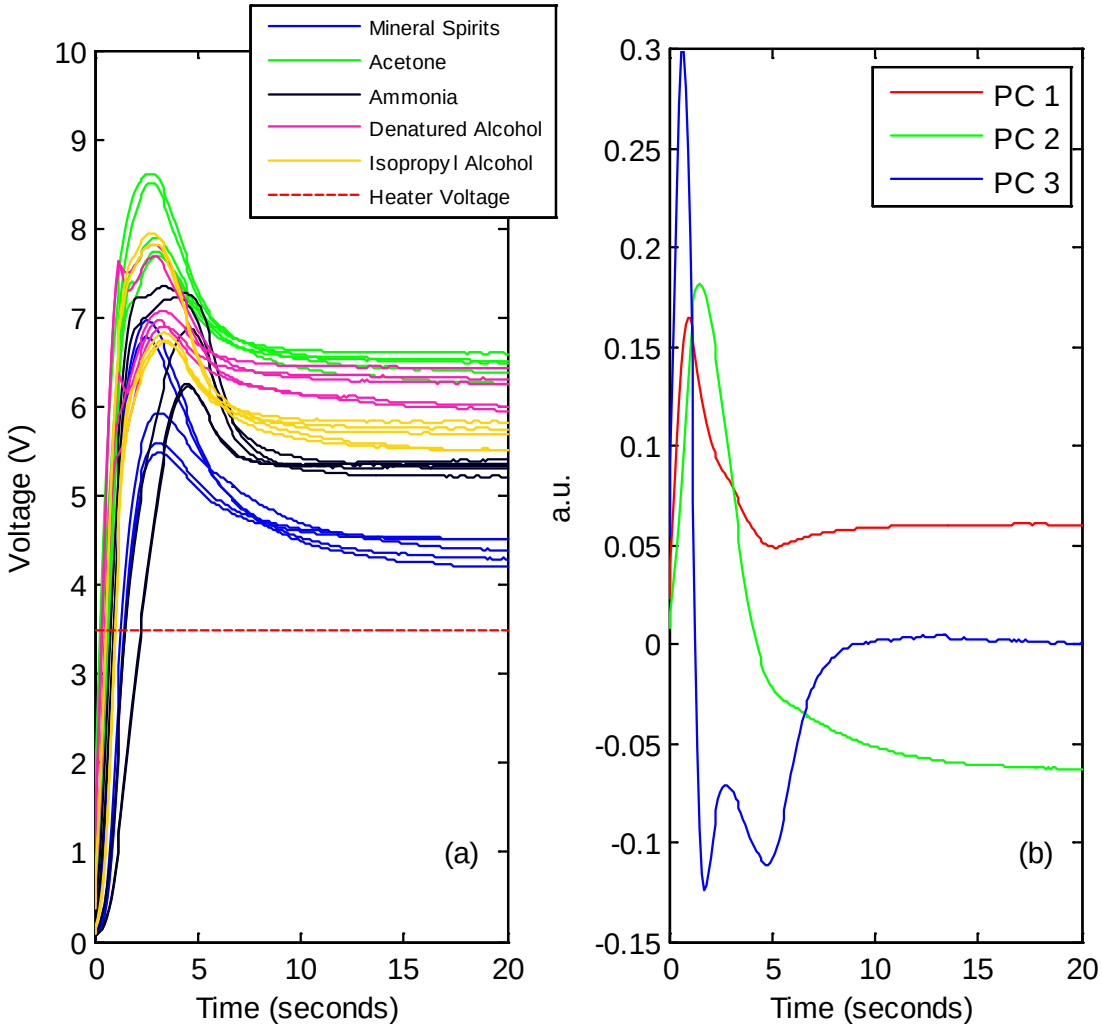

Figure 46 (a) Sensor transients to the five chemicals (5 repetitions per chemical) in response to a 3.5V pulse in heater voltage. This pulse was preceded and succeeded by a 0V pulse of 10-second duration. (b) The first three principal components extracted from the transients in (a).

6.3.2 Offline testing

The first experiment compares the performance of ABS with SFS for $N = 2$ and $N = 3$ sensors with varying number of sensing steps. For $N = 2$, we used data from TGS 2620 and TGS 2602 sensors, and for $N = 3$, we used data from TGS 2620, TGS 2602, and TGS 2610 sensors.

The experiment was conducted using a 20-fold cross-validation loop. For each fold, the experimental data (100 samples) was randomly divided into two subsets: a training dataset containing 40 samples (8 per chemical), and a test dataset containing 60 samples (12 per chemical). We then ran SFS (using both Fisher score and mutual information criteria) to select the best $T = 5$ action vectors. Since the sensors in this array are not identical, SFS was modified as described in section 6.2.4. For each value of N and T, a naïve Bayes classifier was trained on the corresponding feature set and the classification performance estimated on the corresponding test dataset. Using the same cross-validation loop, we also evaluated the classification performance of ABS for both objective functions. Figure 47 summarizes the results in terms of average classification rates and standard deviations.

These results are largely consistent with those on simulated data: the two active sensing methods outperform their passive counterparts, and with increasing number of sensors and sensing steps the classification performance improved. In contrast with results on simulated data, however, the choice of the objective function significantly influenced the performance. For example, at $N = 2$, ABS-WFS outperformed SFS-FS by 3.2% (averaged over all the sensing steps). On the other hand, ABS-DMI outperformed SFS-MI obtained by only 0.5%. We believe this subdued improvement of ABS-DMI is likely because of the limited number of training samples. For DMI to be effective, the joint probability distributions need to be estimated accurately. Considering that there were only 40 training samples for each round of cross-validation, the joint probability distributions are likely not very accurate and neither are the DMI measures.

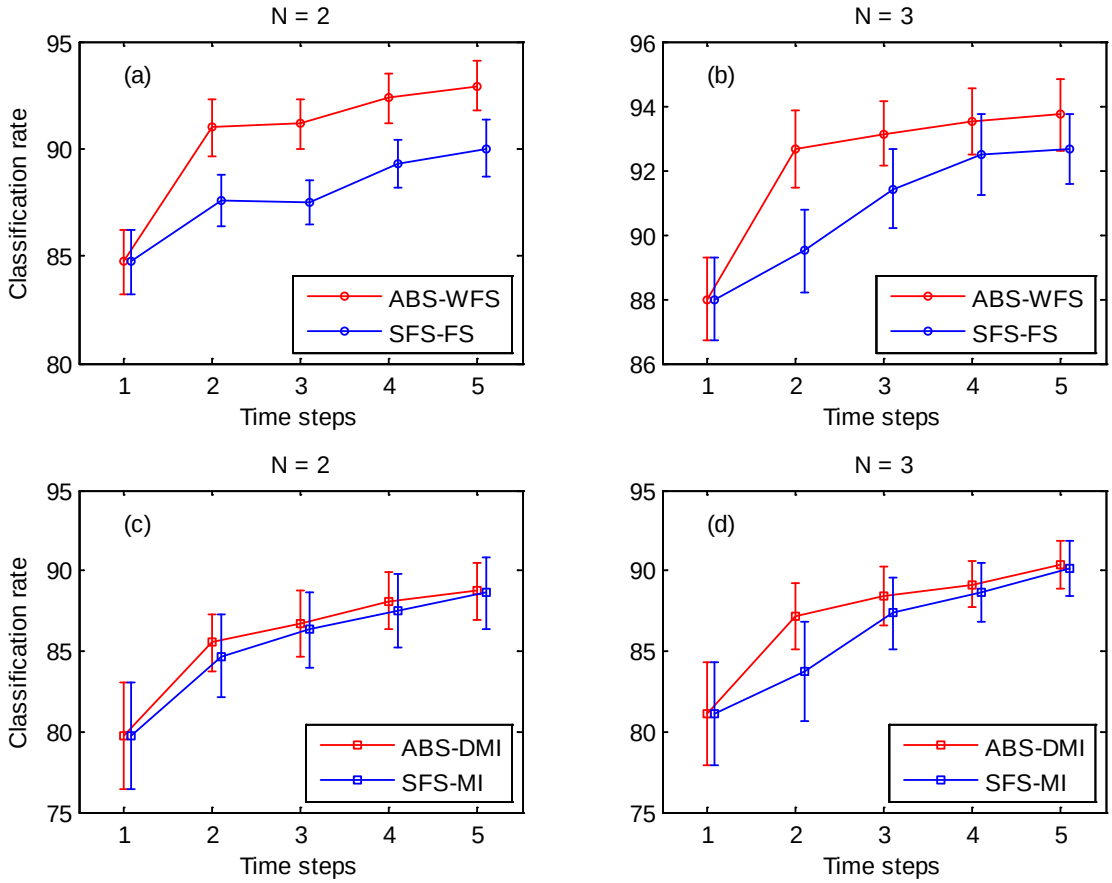

Figure 47 Classification performance of ABS-WFS and SFS-FS for (a) $N = 2$, and (b) $N = 3$ sensors.

A second important observation is that the difference between active and passive methods is lower than on simulated data. This could be attributed to the reduced complexity in the classification problem: 5-class problem vs. 50-class problem. In this classification problem, a single sensing step is sufficient to obtain close to 80% performance, which leaves a limited scope of improvement for active sensing. Lastly, the standard deviations associated with the performance of all four methods are large (in the order of 3%), and comparable to the average difference between active and passive methods. This suggests that there is significant amount of

variation in the data brought about due to the experimental setup. This is consistent with all the experimental results presented in this dissertation.

6.3.3 Online testing

In the final experiment, we validated ABS in an online fashion to establish proof-of-concept. Based on the results of section 6.3.2, we decided to test on a 2-sensor problem with 3 chemicals using WFS as the objective function. Using the experimental protocols described in section 6.3.1, we collected a dataset of temperature-modulated sensor responses for three chemicals: denatured alcohol, mineral spirits, and ammonia. The dataset contained a total of 30 training samples (10 per chemical).

We then tested ABS-WFS 60 times (20 times per chemical) in an online fashion with $f = 10$ sensing steps. As in section 4.3.4, for each test case, the sensor heater voltages were first set to 0V, then the chemical was introduced into the sensor chamber and allowed to equilibrate for 30 seconds, and then finally ABS was commenced to drive the sensors. For each test case, the ABS algorithm was executed for 10 sensing steps. Once again, we tracked all the intermediary belief distributions generated by ABS to estimate its classification performance for $f = 1$ to $f = 9$. We also evaluated SFS-FS with varying number of sensing steps ($f = 1$ to $f = 10$) using the same data that was obtained from the test cases of ABS. Since the two sensors are not are not identical, SFS was modified as described in section 6.2.4. Lastly, we also evaluated the performance of ABS in an offline fashion using only 80% of the training data and 10 rounds of cross-validation.

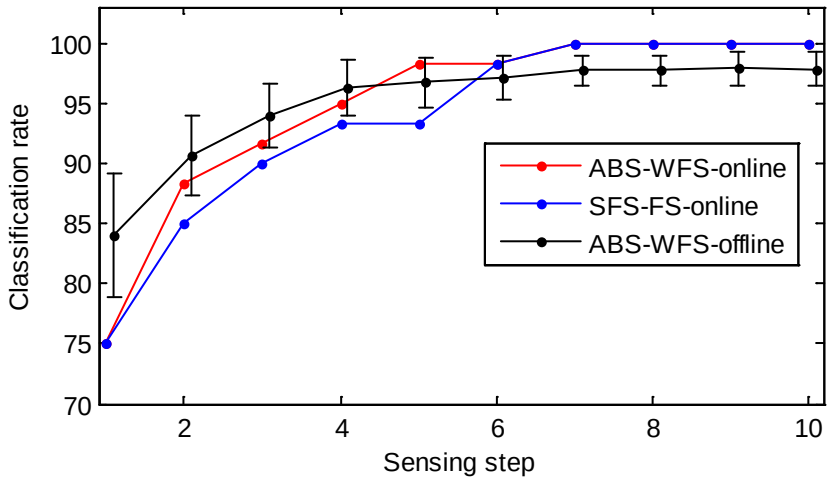

Figure 48 Classification performance of ABS-WFS and SFS-FS for online testing.

Figure 48 summarizes the results in terms of classification rates. As expected, at $f = 1$ both methods obtain the same classification performance. For $f = 2$ to $f = 5$, the active method outperforms the passive method. At $f = 5$, ABS obtains 5% improvement over SFS, which corresponds to 3 test cases. From $f = 6$ to $f = 10$, the two methods obtain the same performance; beyond $f = 7$ both methods obtain perfect classification. In the early sensing steps, the offline version of ABS outperforms online versions of both ABS and SFS. This result is pretty consistent with our observations on the other online experiments (sections 4.3.4 and 5.3.5), and also with the overall hypothesis of this dissertation. Therefore, we conclude that this experiment serves as a preliminary proof-of-concept for active sensing with MOX sensor arrays.

6.4 Discussion

Our results show that ABS is most effective when applied to arrays with moderate number of sensors (in the range of $N = 2$ to $N = 6$). When applied to larger arrays ($N > 10$), the benefits may no longer be very evident because each sensing step provides a large number of different features. If those features have low correlation, most test samples may be classified in a

single step, thus rendering active sensing irrelevant. However, with most commercially available chemical sensor arrays, the features usually have strong correlations. In such scenarios, the ABS method could be used to select configurations, not just for a single sensor but for groups of sensors. As an example, consider a sensor array with $N = 10$ identical sensors. In this case, the sensors may be organized into two groups, each group containing five sensors. At each sensing step, ABS would select two operating configurations, one for each group of sensors, and operate the sensors accordingly. The resulting observations would be averaged by the groups, and this process would reduce noise in the observations. Finally, the averaged values would be used to update the belief distributions.

We also observed that the mutual information measures are sensitive to the discretization step, both to the discretization algorithm itself as well as to the resolution (number of bins and bin sizes). As briefly mentioned in the previous chapter, the choice of the discretization algorithm choice is very important. Uniform discretization, though easy to implement, will fail to capture the nuances of a continuous distribution and is also very sensitive to outliers. Therefore, a clustering-based discretization approach is the better option. Likewise, the number of bins and bin sizes also influence the mutual information measures. If the distribution is approximated with too few bins, the continuous nature of the distribution is not captured. On the other hand, too many bins would result in a very sparse (almost uniform) distribution, and would also increase the run-time computational complexity. For the experimental work in this chapter, we chose the bin counts empirically. This choice could be optimized using a more systematic approach.

WFS assumes that the discriminatory information is in the class means and therefore will be ineffective if the discriminatory information is the variance of the data. This drawback would be more evident if the class-distributions are significantly non-Gaussian. This issue could be addressed by using non-parametric versions of between-class and within-class scatter

matrices as described in Li et al. [223]. Non-parametric scatter matrices are calculated using the differences between samples and their nearest-neighbors (that belong to other classes), as opposed to differences from class-means. This use of local information preserves the structure of the data and relaxes the uni-modal Gaussian assumption of WFS.

The computational complexity of ABS-WFS could be reduced by using trace-ratio: $tr\left(W(\boldsymbol{a}_t)^{-1}B(\boldsymbol{a}_t)\right)$ or ratio-trace: $\frac{tr(B(\boldsymbol{a}_t))}{tr(W(\boldsymbol{a}_t))}$ in eq. (47), as opposed to the determinant. To use trace-ratio, we need to estimate the inverse of within-class variance matrix $W(\boldsymbol{a}_t)$, which can be computed in $O(N^{2.37})$ steps using the Coppersmith-Winograd algorithm [224]. Therefore, with trace-ratio, the worst-case computation complexity of ABS-WFS is $O(N^{3.37}D)$. On the other hand, ratio-trace can be estimated with a much smaller computational cost of $O(N)$, which will reduce ABS's worst-case computational complexity to $O(N^2D)$. However, when compared to trace-ratio or determinant, ratio-trace is a far less informative measure for classification.

ABS chooses sensor configurations based on localized projections of the belief distribution. For example, while selecting the configuration for sensor s_1, the algorithm only considers how this particular configuration would affect the belief distribution b_t. This selection process could be improved by considering centralized projections of belief distribution. For example, if ABS has chosen configuration ρ_1 for sensor s_1. Now, while making a selection for s_2, ABS could consider how the configurations for s_1 and s_2 in combination would affect the belief distribution. The challenge with this centralized approach is that the total number of predicted belief distributions will increase exponentially with each iteration of the forward search.

Over the course of the experimental work for this dissertation, we faced two challenges with online testing due to the nature of the experiment and the experimental apparatus. The first challenge is that, with online testing, the algorithmic parameters have to be set before the

experiment and cannot be adjusted later. If the parameters were not properly selected, then the experiment needs to be repeated, which is both costly and time consuming. The second challenge is that it is rather very tedious to generalize the results of online sensing. With offline experiments, cross-validation can be used to ensure the conclusions were independent of the data sets. On the other hand, with online testing, the only fair way to generalize the results would be to run the experiment multiple times and average the results.

Lastly, ABS is not restricted only to sensor arrays. It could also be applied to a distributed network of tunable chemical sensors, where the sensors are operated by a centralized controller. With sensor arrays, we can assume the concentrations of the target gases across all sensors remains constant. However, this assumption has to be relaxed for a distributed sensor network. Therefore, additional work is needed to address situations where chemical concentrations change both with time and sensor location.

7. CONCLUSIONS

This dissertation studies the benefits and computational challenges of active chemical sensing. Inspired from the rich history and success of active sensing in the fields of robotics, vision, and target tracking, we hypothesized that a similar approach would improve the selectivity and performance of chemical sensors. To test this hypothesis, we developed three active sensing algorithms that address three different computational problems in chemical sensing. These algorithms enable tunable chemical sensors to adapt their configurations in real-time based on the information acquired from the environment, though at an added computational cost. The algorithms are summarized in the following paragraphs.

The first algorithm (Chapter 4) addresses the problem classification with a single chemical sensor. In this algorithm, the dynamic responses of the sensor to different chemicals are model using IOHMMs. Using the IOHMM parameters, the classification problem is then formulated as a POMDP. The POMDP is solved using a myopic approach, which selects the next sensing action to minimize the expected Bayesian risk of misclassifying the sample.

The second algorithm (Chapter 5) tackles the problem of multicomponent analysis. Once again, the problem is cast as that of probabilistic state estimation, where each state represents a different concentration profile and a belief distribution maintains probabilities of all profiles. A myopic approach is employed to solve the problem: it selects the next operating configuration to minimize uncertainty in the future belief distribution.

The third algorithm (Chapter 6) extends active sensing to arrays of tunable sensors. This algorithm combines concepts from feature subset selection and active sensing to enable real-time adaptation of sensor arrays. The critical aspect of this algorithm is its objective function, which incorporates the current belief distribution in evaluating a given sensor-array configuration. We

presented two such objective functions: WFS, which is based on class variance measures, and DMI, which is based on mutual information.

We evaluated the three active sensing algorithms on simulated FPIs and experimentally validated them on platform of commercially available MOX sensors, and compared against similar passive sensing algorithms. Results from the experimental work lead to two main conclusions:

Improved classification performance: For a given sensing budget (for e.g. the number of sensing steps, or total sensing cost), the active sensing methods obtain better classification performance than passive methods. This conclusion is supported by results in sections 4.2.3, 4.3.2, 5.2.2, 5.3.3, 6.2.2, and 6.3.2. This superior performance of active sensing methods can be attributed to two factors: (1) adaptive selection of features: since active methods acquire features based on the current belief distribution, they can modify the sensing programs according to the test sample at hand, and (2) intelligent stopping criterion: active methods can stop acquiring features when either the current belief distribution has sufficient certainty, or further acquisition of features may no longer be beneficial towards classifying the sample.

Robustness to noise: Active sensing methods are more robust to additive Gaussian noise in sensor measurements than passive methods. This conclusion is supported by the results in sections 4.2.4, 5.2.3, and 6.2.3. This could again be attributed to the fact that active methods select features at measurement time, which allows it to adapt to noise, whereas passive methods uses a pre-specified set of features that was computed off-line using noise-free training data. We observed that active methods counter the increasing sensor noise by acquiring more relevant features, which drive the beliefs towards the certainty levels required to stop the sensing process.

7.1 Relation to existing frameworks

The three methods presented in this dissertation are equivalent to POMPDs. However, the definition of states and state transitions vary between the methods. In the first formulation (chapter 4), which is based on Ji et al. [142], each state in the POMDP corresponds to a component in the Gaussian mixture models. In retrospect, we believe that it would be more appropriate if the POMDP states were defined as class labels as opposed to Gaussian components. This is because, with Gaussian components, the true underlying state of the system would become dependent on the observations. In a POMDP formulation, though the observations are necessary to estimate the state of the system, the true underlying state is independent of the observations. This independency is lost with Gaussian components as the POMDP's states.

In the second (chapter 5) and the third (chapter 6) formulations, as in Guo [225], the POMDP states represent the class labels of the test sample. Therefore, the underlying state of the system (which is hidden) is not dependent on the observations. However, this formulation has one main difference from a traditional POMDP: the true state of the system does not change when an action is executed. This is because the sensing actions do not affect the identity or concentration of the sample. Since the true state remains unchanged over time, the formulations adhere to the Markovian assumption.

7.2 Future work

Non-myopic extensions: The three active sensing algorithms presented in this dissertation are myopic: they select sensor configurations with the best potential to reduce uncertainty in the future belief distribution, i.e., immediately after the sensing action. This is in contrast with exact solutions that are optimized over longer horizons though at additional computational costs. Thus, there is a trade-off between long-term and short-term gains [130], and

the required computational complexity. Considering the limited specificity of MOX sensors (even after temperature modulation) and the dynamic characteristics of chemical stimuli, it is unclear whether an optimal policy would provide significant improvements in performance when compared to the myopic approaches tested in this dissertation.

Beyond classification and internal modulation: This dissertation focused largely on the classification problem, and to some extent explores the mixture analysis problem. However, there are many other computational problems in the field of chemosensing, where active sensing would provide significant gains. Examples include: (a) mixture analysis, where the constituent chemicals are unknown, (b) concentration-independent classification, which involves determining the sample's identity irrespective of its concentration, and (c) background suppression, which involves estimating concentration of a chemical in the presence of complex unknown backgrounds. Another extension of this work would be to adapt external parameters of the sensors such as coatings, catalysts, delivery process, or even sensing materials. This would be possible by controlling dilutors, actuators, or microfluidic devices.

Theoretical bounds of active sensing: This dissertation uses an empirical approach to demonstrate the gains of active sensing; however it does not do any formal analysis of the bounds of active sensing. What are the achievable limits of active sensing? How do gains due to active sensing change with noise conditions, distributions of classes, and the regularity of decision boundaries? Answers to such questions would show various conditions under which one can expect significant gains through active sensing. This analysis could be based on minimax methods, which were previously used by Castro et al. [226] to catalogue many theoretical bounds of active learning.

Relaxing the naïve Bayes assumption: The naïve Bayes assumption states that features are independent if the class labels are known. Therefore, the feature vector can be modeled as a

set of independent one-dimensional probability distributions. All active sensing methods presented in this dissertation use the naïve Bayes assumption. Though the naïve Bayes assumption is an over-simplification of the problem (features are rarely independent), research in machine learning has shown that in practice it is very effective because the final classification decisions are often correct even if the probability estimates are inaccurate [227]. If this assumption was relaxed, we do not expect to see significant changes in the results presented in this dissertation. However, relaxing the assumption would increase the run-time computational of the methods, and also create a need for more training data to estimate the required high-dimensional probability distributions.

Experimental apparatus: The apparatus used for this dissertation could be improved to gain more control over the experimental conditions and thus increase the repeatability of the experiments. Specifically, the odor delivery system could be improved with the addition of mass-flow controllers to stabilize analyte flow, and pressurized cylinders to deliver analytes at a precise concentration. Finally, a closed-loop temperature control would give greater control over the sensors' operating temperatures.

Taking cues from chemistry: Organic chemicals are often organized based on their functional groups. A functional group is a group of atoms or bonds that are responsible for the characteristic chemical and physical properties of an organic compound. For example, ketones are a group of organic compounds that contain a carbonyl group bonded to two hydrocarbon groups. Ketones demonstrate a strong absorption peak in the infrared spectrum near 1700 cm^{-1}. We expect this type of chemical information would be useful in driving the active sensing methods. During the early sensing steps, the sensors would be operated at wavelengths that can eliminate certain functional groups completely, and thus effectively simplify the classification problem. Inspired from workings of biological olfactory systems, Raman et al. [228] proposed a

similar approach where odors were organized hierarchically based on their chemical composition. The authors demonstrated that the approach not only allowed them to classify samples from known classes but also identify the composition of samples that were not a part of the training dataset.

This discussion brings about two hypotheses that warrant further investigation. First, when tested on a large classification problem (which includes chemicals from various functional groups), we expect that the active sensing algorithms will automatically learn to sense at the characteristic wavelengths of different functional groups. Second, as in Raman's work, if the active sensing methods were tested on sample from an unknown class, we expect the methods would assign high belief to known classes that contain the same functional group as the test sample. This would help to partially identify the chemical composition of the sample.

REFERENCES

[1] R. Gutierrez-Osuna, and A. Hierlemann, "Adaptive microsensor systems," *Annual Review of Analytical Chemistry,* vol. 3, no. 1, pp. 255-276, 2010.

[2] A. O. Hero, D. A. Castanon, D. Cochran, and K. Kastella, *Foundations and applications of sensor management,* 1st ed., New York: Springer, 2007.

[3] N. Barsan, D. Koziej, and U. Weimar, "Metal oxide-based gas sensor research: How to?," *Sensors and Actuators B: Chemical,* vol. 121, no. 1, pp. 18-35, 2007.

[4] J. Janata, and M. Josowicz, "Conducting polymers in electronic chemical sensors," *Nature materials,* vol. 2, no. 1, pp. 19-24, 2003.

[5] C. Di Natale, R. Paolesse, A. Macagnano, A. Mantini, C. Goletti, and A. D'Amico, "Characterization and design of porphyrins-based broad selectivity chemical sensors for electronic nose applications," *Sensors and Actuators B: Chemical,* vol. 52, no. 1, pp. 162-168, 1998.

[6] W. Gopel, "Supramolecular and polymeric structures for gas sensors," *Sensors and Actuators B: Chemical,* vol. 24, no. 1, pp. 17-32, 1995.

[7] B. Stuart, *Infrared spectroscopy: fundamentals and applications,* 1st ed., Chichester: Wiley, 2004.

[8] J. W. Akitt, and B. E. Mann, *NMR and Chemistry: An introduction to modern NMR spectroscopy,* 4th ed., New York: Sanley Thornes, 2000.

[9] J. Janata, *Principles of chemical sensors,* 2nd ed., New York: Springer Verlag, 2009.

[10] A. Hierlemann, and R. Gutierrez-Osuna, "Higher-order chemical sensing," *Chemical Reviews,* vol. 108, no. 2, pp. 563-613, 2008.

[11] F. Rock, N. Barsan, and U. Weimar, "Electronic nose: Current status and future trends," *Chemical Reviews,* vol. 108, no. 2, pp. 705-725, 2008.

[12] D. Diamond, S. Coyle, S. Scarmagnani, and J. Hayes, "Wireless sensor networks and chemo-/biosensing," *Chemical Reviews,* vol. 108, no. 2, pp. 652-679, 2008.

[13] E. L. Kalman, F. Winquist, and I. Lundstrom, "A new pollen detection method based on an electronic nose," *Atmospheric Environment,* vol. 31, no. 11, pp. 1715-1719, 1997.

[14] W. Tsujita, A. Yoshino, H. Ishida, and T. Moriizumi, "Gas sensor network for air-pollution monitoring," *Sensors and Actuators B: Chemical*, vol. 110, no. 2, pp. 304-311, 2005.

[15] B. Pejcic, P. Eadington, and A. Ross, "Environmental monitoring of hydrocarbons: A chemical sensor perspective," *Environmental Science and Technology*, vol. 41, no. 18, pp. 6333-6342, 2007.

[16] K. S. Johnson, J. A. Needoba, S. C. Riser, and W. J. Showers, "Chemical sensor networks for the aquatic environment," *Chemical Reviews*, vol. 107, no. 2, pp. 623-640, 2007.

[17] D. Thouron, R. Vuillemin, X. Philippon, A. Lourenço, C. Provost, A. Cruzado, and V. Garçon, "An autonomous nutrient analyzer for oceanic long-term in situ biogeochemical monitoring," *Analytical Chemistry*, vol. 75, no. 11, pp. 2601-2609, 2003.

[18] N. N. Rabalais, "Nitrogen in aquatic ecosystems," *AMBIO: A Journal of the Human Environment*, vol. 31, no. 2, pp. 102-112, 2002.

[19] R. Cardell-Oliver, M. Kranz, K. Smettem, and K. Mayer, "A reactive soil moisture sensor network: Design and field evaluation," *International Journal of Distributed Sensor Networks*, vol. 1, no. 2, pp. 149-162, 2005.

[20] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pp. 214-226, 2004.

[21] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh, "Deploying a wireless sensor network on an active volcano," *IEEE Internet Computing*, vol. 10, no. 2, pp. 18-25, 2006.

[22] H. T. Nagle, S. S. Schiffman, and R. Gutierrez-Osuna, "The how and why of electronic noses," *IEEE Spectrum*, vol. 35, no. 9, pp. 22-31, 1998.

[23] D. A. Healy, C. J. Hayes, P. Leonard, L. McKenna, and R. O'Kennedy, "Biosensor developments: application to prostate-specific antigen detection," *Trends in Biotechnology*, vol. 25, no. 3, pp. 125-131, 2007.

[24] H. Vaisocherova, K. Mrkvova, M. Piliarik, P. Jinoch, M. Steinbachova, and J. Homola, "Surface plasmon resonance biosensor for direct detection of antibody against Epstein-Barr virus," *Biosensors and Bioelectronics*, vol. 22, no. 6, pp. 1020-1026, 2007.

[25] J. Wei, Y. Mu, D. Song, X. Fang, X. Liu, L. Bu, H. Zhang, G. Zhang, J. Ding, and W. Wang, "A novel sandwich immunosensing method for measuring cardiac troponin I in sera," *Analytical Biochemistry*, vol. 321, no. 2, pp. 209-216, 2003.

[26] M. Tierney, J. Tamada, R. Potts, L. Jovanovic, and S. Garg, "Clinical evaluation of the GlucoWatch biographer: A continual, non-invasive glucose monitor for patients with diabetes," *Biosensors and Bioelectronics*, vol. 16, no. 9, pp. 621-629, 2001.

[27] T. Vo-Dinh, and B. Cullum, "Biosensors and biochips: Advances in biological and medical diagnostics," *Fresenius' Journal of Analytical Chemistry*, vol. 366, no. 6, pp. 540-551, 2000.

[28] A. K. Deisingh, D. C. Stone, and M. Thompson, "Applications of electronic noses and tongues in food analysis," *International Journal of Food Science and Technology*, vol. 39, no. 6, pp. 587-604, 2004.

[29] M. Peris, and L. Escuder-Gilabert, "A 21st century technique for food control: Electronic noses," *Analytica Chimica Acta*, vol. 638, no. 1, pp. 1-15, 2009.

[30] C. Pinheiro, C. M. Rodrigues, T. Schafer, and J. G. Crespo, "Monitoring the aroma production during wine–must fermentation with an electronic nose," *Biotechnology and Bioengineering*, vol. 77, no. 6, pp. 632-640, 2002.

[31] J. Brezmes, E. Llobet, X. Vilanova, J. Orts, G. Saiz, and X. Correig, "Correlation between electronic nose signals and fruit quality indicators on shelf-life measurements with pinklady apples," *Sensors and Actuators B: Chemical*, vol. 80, no. 1, pp. 41-50, 2001.

[32] W. X. Du, C. M. Lin, T. Huang, J. Kim, M. Marshall, and C. I. Wei, "Potential application of the electronic nose for quality assessment of salmon fillets under various storage conditions," *Journal of Food Science*, vol. 67, no. 1, pp. 307-313, 2002.

[33] M. Penza, and G. Cassano, "Recognition of adulteration of Italian wines by thin-film multisensor array and artificial neural networks," *Analytica Chimica Acta*, vol. 509, no. 2, pp. 159-177, 2004.

[34] R. Stella, J. N. Barisci, G. Serra, G. G. Wallace, and D. De Rossi, "Characterisation of olive oil by an electronic nose based on conducting polymer sensors," *Sensors and Actuators B: Chemical*, vol. 63, no. 1, pp. 1-9, 2000.

[35] E. J. Houser, T. E. Mlsna, V. K. Nguyen, R. Chung, R. L. Mowery, and R. Andrew McGill, "Rational materials design of sorbent coatings for explosives: Applications with chemical sensors," *Talanta*, vol. 54, no. 3, pp. 469-485, 2001.

[36] S. J. Toal, and W. C. Trogler, "Polymer sensors for nitroaromatic explosives detection," *Journal of Materials Chemistry,* vol. 16, no. 28, pp. 2871-2883, 2006.

[37] J. I. Steinfeld, and J. Wormhoudt, "Explosives detection: A challenge for physical chemistry," *Annual Review of Physical Chemistry,* vol. 49, no. 1, pp. 203-232, 1998.

[38] A. Jeremic, and A. Nehorai, "Landmine detection and localization using chemical sensor array processing," *IEEE Transactions on Signal Processing,* vol. 48, no. 5, pp. 1295-1305, 2000.

[39] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, and M. Gouda, "A line in the sand: A wireless sensor network for target detection, classification, and tracking," *Computer Networks,* vol. 46, no. 5, pp. 605-634, 2004.

[40] M. A. El-Sherif, J. Yuan, and A. Macdiarmid, "Fiber optic sensors and smart fabrics," *Journal of intelligent material systems and structures*, vol. 11, no. 5, pp. 407-414, 2000.

[41] S. J. Martin, "Conventional analytical methods for chemical warfare agents," *Pure and Applied Chemistry*, vol. 74, no. 12, pp. 2281-2292, 2002.

[42] D. Zarzhitsky, D. F. Spears, and W. M. Spears, "Distributed robotics approach to chemical plume tracing," *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 4034-4039, 2005.

[43] A. Branca, P. Simonian, M. Ferrante, E. Novas, and R. M. Negri, "Electronic nose based discrimination of a perfumery compound in a fragrance," *Sensors and Actuators B: Chemical*, vol. 92, no. 1, pp. 222-227, 2003.

[44] L. Jing, G. Yu, L. Yijiang, H. Chang, A. Hannon, D. Kim, and S. Dennis, "Nanotechnology based cell-all phone-sensors for extended network chemical sensing," *Proceedings of the IEEE Sensors Conference*, pp. 1-4, 2012.

[45] G. Bazzu, G. G. Puggioni, S. Dedola, G. Calia, G. Rocchitta, R. Migheli, M. S. Desole, J. P. Lowry, R. D. O'Neill, and P. A. Serra, "Real-time monitoring of brain tissue oxygen using a miniaturized biotelemetric device implanted in freely moving rats," *Analytical Chemistry*, vol. 81, no. 6, pp. 2235-2241, 2009.

[46] A. Schwake, B. Ross, and K. Cammann, "Chrono amperometric determination of hydrogen peroxide in swimming pool water using an ultramicroelectrode array," *Sensors and Actuators B: Chemical*, vol. 46, no. 3, pp. 242-248, 1998.

[47] A. van den Berg, A. Grisel, E. Verney-Norberg, B. H. van der Schoot, M. Koudelka-Hep, and N. F. de Rooij, "On-wafer fabricated free-chlorine sensor with ppb detection limit for drinking-water monitoring," *Sensors and Actuators B: Chemical,* vol. 13, no. 1, pp. 396-399, 1993.

[48] C. G. Zoski, *Handbook of electrochemistry,* 1st ed., Amsterdam: Elsevier Science, 2007.

[49] A. J. Bard, and L. R. Faulkner, *Electrochemical methods: Fundamentals and applications,* 2nd ed., New York: Wiley

[50] J. Faist, F. Capasso, D. L. Sivco, C. Sirtori, A. L. Hutchinson, and A. Y. Cho, "Quantum cascade laser," *Science,* vol. 264, pp. 553-556, 1994.

[51] G. Wysocki, R. Lewicki, R. Curl, F. Tittel, L. Diehl, F. Capasso, M. Troccoli, G. Hofler, D. Bour, and S. Corzine, "Widely tunable mode-hop free external cavity quantum cascade lasers for high resolution spectroscopy and chemical sensing," *Applied Physics B: Lasers and Optics,* vol. 92, no. 3, pp. 305-311, 2008.

[52] A. P. Lee, and B. J. Reedy, "Temperature modulation in semiconductor gas sensing," *Sensors and Actuators B: Chemical,* vol. 60, no. 1, pp. 35-42, 1999.

[53] T. Otagawa, and J. R. Stetter, "A chemical concentration modulation sensor for selective detection of airborne chemicals," *Sensors and Actuators,* vol. 11, no. 3, pp. 251-264, 1987.

[54] C. Hagleitner, D. Lange, A. Hierlemann, O. Brand, and H. Baltes, "CMOS single-chip gas detection system comprising capacitive, calorimetric and mass-sensitive microsensors," *IEEE Journal of Solid-State Circuits,* vol. 37, no. 12, pp. 1867-1878, 2002.

[55] S. Vaihinger, W. Gopel, and J. R. Stetter, "Detection of halogenated and other hydrocarbons in air: Response functions of catalyst/electrochemical sensor systems," *Sensors and Actuators B: Chemical,* vol. 4, no. 3–4, pp. 337-343, 1991.

[56] P. M. Schweizer-Berberich, S. Vaihinger, and W. Gopel, "Characterisation of food freshness with sensor arrays," *Sensors and Actuators B: Chemical,* vol. 18, no. 1, pp. 282-290, 1994.

[57] K. Persaud, and G. Dodd, "Analysis of discrimination mechanisms in the mammalian olfactory system using a model nose," *Nature,* vol. 299, no. 5881, pp. 352-355, 1982.

[58] L. Buck, and R. Axel, "A novel multigene family may encode odorant receptors: A molecular basis for odor recognition," *Cell,* vol. 65, no. 1, pp. 175-187, 1991.

[59] D. M. Wilson, S. Garrod, S. Hoyt, S. McKennoch, and K. S. Booksh, "Array optimization and preprocessing techniques for chemical sensing microsystems," *Sensors Update,* vol. 10, no. 1, pp. 77-106, 2002.

[60] C. Hagleitner, D. Lange, A. Hierlemann, O. Brand, and H. Baltes, "CMOS single-chip gas detection system comprising capacitive, calorimetric and mass-sensitive microsensors," *Solid-State Circuits, IEEE Journal of,* vol. 37, no. 12, pp. 1867-1878, 2002.

[61] P. Kurzawski, C. Hagleitner, and A. Hierlemann, "Detection and discrimination capabilities of a multitransducer single-chip gas sensor system," *Analytical Chemistry,* vol. 78, no. 19, pp. 6910-6920, 2006.

[62] B. J. Polk, J. A. Smith, S. P. DeWeerth, Z. Zhou, J. Janata, and K. Domansky, "Design of solid state array for simultaneous potentiometric and impedance sensing in gas phase," *Electroanalysis,* vol. 11, no. 10-11, pp. 707-711, 1999.

[63] A. A. Tomchenko, G. P. Harmer, B. T. Marquis, and J. W. Allen, "Semiconducting metal oxide sensor array for the selective detection of combustion gases," *Sensors and Actuators B: Chemical,* vol. 93, no. 1, pp. 126-134, 2003.

[64] N. A. Rakow, and K. S. Suslick, "A colorimetric sensor array for odour visualization," *Nature,* vol. 406, no. 6797, pp. 710-713, 2000.

[65] Y. Li, C. Vancura, D. Barrettino, M. Graf, C. Hagleitner, A. Kummer, M. Zimmermann, K. U. Kirstein, and A. Hierlemann, "Monolithic CMOS multi-transducer gas sensor microsystem for organic and inorganic analytes," *Sensors and Actuators B: Chemical,* vol. 126, no. 2, pp. 431-440, 2007.

[66] D. Sauter, U. Weimar, G. Noetzel, J. Mitrovics, and W. Gopel, "Development of modular ozone sensor system for application in practical use," *Sensors and Actuators B: Chemical,* vol. 69, no. 1, pp. 1-9, 2000.

[67] K. J. Albert, N. S. Lewis, C. L. Schauer, G. A. Sotzing, S. E. Stitzel, T. P. Vaid, and D. R. Walt, "Cross-reactive chemical sensor arrays," *Chemical Reviews,* vol. 100, no. 7, pp. 2595-2626, 2000.

[68] K. Ihokura, and J. Watson, *The stannic oxide gas sensor: Principles and applications*, 1st ed., Boca Raton: CRC Press, 1994.

[69] A. Gutierrez Galvez, "Coding and learning of chemosensor array patterns in a neurodynamic model of the olfactory system," Ph.D. Dissertation, Texas A&M University, College Station, 2006.

[70] H. Abe, T. Yoshimura, S. Kanaya, Y. Takahashi, Y. Miyashita, and S. I. Sasaki, "Automated odor-sensing system based on plural semiconductor gas sensors and computerized pattern recognition techniques," *Analytica Chimica Acta,* vol. 194, pp. 1-9, 1987.

[71] M. J. Madou, and S. R. Morrison, *Chemical sensing with solid state devices*, 1st ed., Boston: Academic Press, 1989.

[72] S. Semancik, R. E. Cavicchi, M. C. Wheeler, J. E. Tiffany, G. E. Poirier, R. M. Walton, J. S. Suehle, B. Panchapakesan, and D. L. DeVoe, "Microhotplate platforms for chemical sensor research," *Sensors and Actuators B: Chemical,* vol. 77, no. 1, pp. 579-591, 2001.

[73] R. E. Cavicchi, J. S. Suehle, K. G. Kreider, M. Gaitan, and P. Chaparala, "Fast temperature programmed sensing for micro-hotplate gas sensors," *IEEE Electron Device Letters,* vol. 16, no. 6, pp. 286-288, 1995.

[74] W. M. Sears, K. Colbow, and F. Consadori, "General characteristics of thermally cycled tin oxide gas sensors," *Semiconductor Science and Technology,* vol. 4, no. 5, pp. 351-359, 1989.

[75] S. Nakata, S. Akakabe, M. Nakasuji, and K. Yoshikawa, "Gas sensing based on a nonlinear response: Discrimination between hydrocarbons and quantification of individual components in a gas mixture," *Analytical Chemistry,* vol. 68, no. 13, pp. 2067-2072, 1996.

[76] S. Nakata, H. Okunishi, and Y. Nakashima, "Distinction of gases with a semiconductor sensor under a cyclic temperature modulation with second-harmonic heating," *Sensors and Actuators B: Chemical,* vol. 119, no. 2, pp. 556–561, 2006.

[77] S. Nakata, H. Okunishi, and S. Inooka, "Gas-sensing system based on the cyclic temperature: Further characterization by the second harmonic perturbation," *Analytica Chimica Acta,* vol. 517, no. 1, pp. 153-159, 2004.

[78] A. Heilig, N. Barsan, U. Weimar, M. Schweizer-Berberich, J. W. Gardner, and W. Gopel, "Gas identification by modulating temperatures of SnO2-based thick film sensors," *Sensors and Actuators B: Chemical,* vol. 43, no. 1, pp. 45-51, 1997.

[79] A. Fort, M. Gregorkiewitz, N. Machetti, S. Rocchi, B. Serrano, L. Tondi, N. Ulivieri, V. Vignoli, G. Faglia, and E. Comini, "Selectivity enhancement of SnO2 sensors by means of operating temperature modulation," *Thin Solid Films,* vol. 418, no. 1, pp. 2-8, 2002.

[80] R. Ionescu, and E. Llobet, "Wavelet transform-based fast feature extraction from temperature modulated semiconductor gas sensors," *Sensors and Actuators B: Chemical,* vol. 81, no. 2, pp. 289-295, 2002.

[81] A. Vergara, J. L. Ramírez, and E. Llobet, "Reducing power consumption via a discontinuous operation of temperature-modulated micro-hotplate gas sensors: Application to the logistics chain of fruit," *Sensors and Actuators B: Chemical,* vol. 129, no. 1, pp. 311-318, 2008.

[82] A. Vergara, E. Llobet, J. Brezmes, X. Vilanova, P. Ivanov, I. Gracia, C. Cane, and X. Correig, "Optimized temperature modulation of micro-hotplate gas sensors through pseudorandom binary sequences," *IEEE Sensors Journal,* vol. 5, no. 6, pp. 1369-1378, 2005.

[83] A. Vergara, E. Llobet, J. Brezmes, P. Ivanov, X. Vilanova, I. Gracia, C. Cane, and X. Correig, "Optimised temperature modulation of metal oxide micro-hotplate gas sensors through multilevel pseudo random sequences," *Sensors and Actuators B: Chemical,* vol. 111, pp. 271-280, 2005.

[84] A. Vergara, E. Llobet, J. Brezmes, P. Ivanov, C. Cane, I. Gracia, X. Vilanova, and X. Correig, "Quantitative gas mixture analysis using temperature-modulated micro-hotplate gas sensors: Selection and validation of the optimal modulating frequencies," *Sensors and Actuators B: Chemical,* vol. 123, no. 2, pp. 1002-1016, 2007.

[85] Y. Hiranaka, T. Abe, and H. Murata, "Gas-dependent response in the temperature transient of SnO2 gas sensors," *Sensors and Actuators B: Chemical,* vol. 9, no. 3, pp. 177-182, 1992.

[86] B. Yea, T. Osaki, K. Sugahara, and R. Konishi, "The concentration-estimation of inflammable gases with a semiconductor gas sensor utilizing neural networks and fuzzy inference," *Sensors and Actuators B: Chemical,* vol. 41, no. 1, pp. 121-129, 1997.

[87] Y. Kato, K. Yoshikawa, and M. Kitora, "Temperature-dependent dynamic response enables the qualification and quantification of gases by a single sensor," *Sensors and Actuators B: Chemical,* vol. 40, no. 1, pp. 33-37, 1997.

[88] X. Huang, F. Meng, Z. Pi, W. Xu, and J. Liu, "Gas sensing behavior of a single tin dioxide sensor under dynamic temperature modulation," *Sensors and Actuators B: Chemical,* vol. 99, no. 2, pp. 444-450, 2004.

[89] K. A. Ngo, P. Lauque, and K. Aguir, "High performance of a gas identification system using sensor array and temperature modulation," *Sensors and Actuators B: Chemical,* vol. 124, no. 1, pp. 209-216, 2007.

[90] A. Vergara, E. Llobet, E. Martinelli, C. Di Natale, A. D'Amico, and X. Correig, "Feature extraction of metal oxide gas sensors using dynamic moments," *Sensors and Actuators B: Chemical,* vol. 122, no. 1, pp. 219-226, 2007.

[91] R. Gutierrez-Osuna, A. Gutierrez-Galvez, and N. Powar, "Transient response analysis for temperature-modulated chemoresistors," *Sensors and Actuators B: Chemical,* vol. 93, no. 1, pp. 57-66, 2003.

[92] S. Marco, J. Samitier, and J. Morante, "A novel time-domain method to analyse multicomponent exponential transients," *Measurement Science and Technology,* vol. 6, no. 2, pp. 135-142, 1999.

[93] M. Nakamura, I. Sugimoto, H. Kuwano, and R. Lemos, "Chemical sensing by analysing dynamics of plasma polymer film-coated sensors," *Sensors and Actuators B: Chemical,* vol. 20, no. 2, pp. 231-237, 1994.

[94] R. Gosangi, and R. Gutierrez-Osuna, "Data-driven modeling of metal-oxide sensors with dynamic Bayesian networks," *AIP Conference Proceedings,* vol. 1362, no. 1, pp. 135-136, 2011.

[95] E. Martinelli, D. Polese, A. Catini, A. D'Amico, and C. Di Natale, "Self-adapted temperature modulation in metal-oxide semiconductor gas sensors," *Sensors and Actuators B: Chemical,* vol. 161, no. 1, pp. 534-541, 2012.

[96] T. A. Kunt, T. J. McAvoy, R. E. Cavicchi, and S. Semancik, "Optimization of temperature programmed sensing for gas identification using micro-hotplate sensors," *Sensors and Actuators B: Chemical,* vol. 53, no. 1, pp. 24-43, 1998.

[97] J. F. Mulligan, "Who were Fabry and Perot?," *American Journal of Physics,* vol. 66, no. 9, pp. 797-801, 1998.

[98] J. Vaughan, *The Fabry-Perot interferometer: History, theory, practice, and applications*, 1st ed., New York: Taylor & Francis, 1989.

[99] J. Jerman, and D. Clift, "Miniature Fabry-Perot interferometers micromachined in silicon for use in optical fiber WDM systems," *Proceedings of the International Conference on Solid-State Sensors and Actuators,* pp. 372-375, 1991.

[100] H. Alause, F. Grasdepot, J. Malzac, W. Knap, and J. Hermann, "Micromachined optical tunable filter for domestic gas sensors," *Sensors and Actuators B: Chemical,* vol. 43, no. 1, pp. 18-23, 1997.

[101] M. E. Nelson, and M. A. MacIver, "Sensory acquisition in active sensing systems," *Journal Of Comparative Physiology A,* vol. 192, no. 6, pp. 573-586, 2006.

[102] J. A. Thomas, C. F. Moss, and M. Vater, *Echolocation in bats and dolphins*, 1st ed., Chicago: University of Chicago Press, 2002.

[103] R. Bajcsy, "Active perception," *Proceedings of the IEEE,* vol. 76, no. 8, pp. 966-1005, 1988.

[104] J. J. Gibson, "Observations on active touch," *Psychological Review,* vol. 69, pp. 477-491, 1962.

[105] J. J. Gibson, *The ecological approach to visual perception,* 1st ed., New Jersey: Lawrence Erlbaum Associates, 1986.

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

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

[108] D. Murray, and A. Basu, "Motion tracking with an active camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 16, no. 5, pp. 449-459, 1994.

[109] E. Marchand, and F. Chaumette, "Active vision for complete scene reconstruction and exploration," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 21, no. 1, pp. 65-72, 1999.

[110] M. Tistarelli, and E. Grosso, "Active vision-based face authentication," *Image and Vision Computing,* vol. 18, no. 4, pp. 299-314, 2000.

[111] A. J. Davison, and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, no. 7, pp. 865-880, 2002.

[112] A. Mishra, Y. Aloimonos, and C. L. Fah, "Active segmentation with fixation," *Proceedings of the IEEE International Conference on Computer Vision,* pp. 468-475, 2009.

[113] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *The International Journal of Robotics Research,* vol. 30, no. 11, pp. 1343-1377, 2011.

[114] R. Simmons, and S. Koenig, "Probabilistic robot navigation in partially observable environments," *Proceedings of the International Joint Conference on Artificial Intelligence,* pp. 1080-1087, 1995.

[115] H. Zhou, and S. Sakane, "Mobile robot localization using active sensing based on Bayesian network inference," *Robotics and Autonomous Systems,* vol. 55, no. 4, pp. 292-305, 2007.

[116] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of Artificial Intelligence Research,* vol. 11, 1999.

[117] A. J. Davison, and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, no. 7, pp. 865-880, 2002.

[118] K. Patel, W. Macklem, S. Thrun, and M. Montemerlo, "Active sensing for high-speed offroad driving," *Proceedings of the IEEE International Conference on Robotics and Automation,* pp. 3162-3168, 2005.

[119] L. Pedersen, M. Wagner, D. Apostolopoulos, and W. R. Whittaker, "Autonomous robotic meteorite identification in Antarctica," *Proceedings of the IEEE International Conference on Robotics and Automation,* pp. 4158-4165, 2001.

[120] M. J. Pearson, A. G. Pipe, C. Melhuish, B. Mitchinson, and T. J. Prescott, "Whiskerbot: a robotic active touch system modeled on the rat whisker sensory system," *Adaptive Behavior,* vol. 15, no. 3, pp. 223-240, 2007.

[121] M. A. Fenelon, and T. Furukawa, "Design of an active flapping wing mechanism and a micro aerial vehicle using a rotary actuator," *Mechanism and machine theory,* vol. 45, no. 2, pp. 137-146, 2010.

[122] J. R. Solberg, K. M. Lynch, and M. Maclver, "Robotic electrolocation: Active underwater target localization with electric fields," *Proceedings of the IEEE International Conference on Robotics and Automation,* pp. 4879-4886, 2007.

[123] M. Kaneko, N. Kanayama, and T. Tsuji, "Active antenna for contact sensing," *IEEE Transactions on Robotics and Automation,* vol. 14, no. 2, pp. 278-291, 1998.

[124] L. Mihaylova, T. Lefebvre, H. Bruyninckx, K. Gadeyne, and J. D. Schutter, "Active sensing for robotics-a survey," *Proceedings of the International Conference on Numerical Methods and Applications,* pp. 316–324, 2002.

[125] C. Kreucher, K. Kastella, and A. O. Hero, "Sensor management using an active sensing approach," *Signal Processing*, vol. 85, no. 3, pp. 607-624, 2005.

[126] T. H. Chung, V. Gupta, J. W. Burdick, and R. M. Murray, "On a decentralized active sensing strategy using mobile sensor platforms in a network," *Proceedings of the IEEE Conference on Decision and Control*, pp. 1914-1919, 2004.

[127] P. Yang, R. A. Freeman, and K. M. Lynch, "Distributed cooperative active sensing using consensus filters," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 405-410, 2007.

[128] P. Yang, R. A. Freeman, and K. M. Lynch, "Multi-agent coordination by decentralized estimation and control," *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2480-2496, 2008.

[129] T. Mukai, and M. Ishikawa, "An active sensing method using estimated errors for multisensor fusion systems," *IEEE Transactions on Industrial Electronics*, vol. 43, no. 3, pp. 380-386, 1996.

[130] R. S. Sutton, and A. G. Barto, *Reinforcement Learning: An Introduction*, 1st ed., Cambridge: MIT Press, 1998.

[131] R. Bellman, "Dynamic programming and the smoothing problem," *Management Science*, vol. 3, no. 1, pp. 111-113, 1956.

[132] M. Ghallab, D. Lau, and P. Traverso, *Automated Planning - theory and practice*, 1st ed., Burlington: Morgan Kaufmann, 2004.

[133] G. E. Monahan, "A survey of partially observable Markov decision processes: Theory, models, and algorithms," *Management Science*, vol. 28, no. 1, pp. 1-16, January 1, 1982, 1982.

[134] R. A. Howard, *Dynamic programming and Markov process*, 1st ed., Cambridge: MIT Press, 1960.

[135] C. H. Papadimitriou, and J. N. Tsitsiklis, "The complexity of Markov decision processes," *Mathematics of Operations Research*, vol. 12, no. 3, pp. 441-450, 1987.

[136] O. Madani, S. Hanks, and A. Condon, "On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems," *Proceedings of the National Conference on Artificial Intelligence*, pp. 541-548, 1999.

[137] M. Hauskrecht, "Value-function approximations for partially observable Markov decision processes," *Journal of Artificial Intelligence Research,* vol. 13, pp. 33-94, 2000.

[138] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration: An anytime algorithm for POMDPs," *Proceedings of the International Joint Conference on Artificial Intelligence,* pp. 1025-1032, 2003.

[139] M. T. J. Spaan, and N. Vlassis, "Perseus: Randomized point-based value iteration for POMDPs," *Journal of Artificial Intelligence Research,* vol. 24, no. 1, pp. 195-220, 2005.

[140] D. A. Castanon, "Approximate dynamic programming for sensor management," *Proceedings of the IEEE Conference on Decision and Control,* pp. 1202-1207 vol.2, 1997.

[141] V. Krishnamurthy, "Algorithms for optimal scheduling and management of hidden Markov model sensors," *IEEE Transactions on Signal Processing,* vol. 50, no. 6, pp. 1382-1397, 2002.

[142] S. Ji, and L. Carin, "Cost-sensitive feature acquisition and classification," *Pattern Recognition,* vol. 40, no. 5, pp. 1474-1485, 2007.

[143] S. Ji, R. Parr, and L. Carin, "Nonmyopic multiaspect sensing with partially observable Markov decision processes," *IEEE Transactions on Signal Processing,* vol. 55, no. 6, pp. 2720-2730, 2007.

[144] S. A. Miller, Z. A. Harris, and E. K. P. Chong, "A POMDP framework for coordinated guidance of autonomous UAVs for multitarget tracking," *EURASIP Journal on Advances in Signal Processing,* vol. 2009, pp. 1-17, 2009.

[145] J. C. Gittins, and D. M. Jones, "A dynamic allocation index for the discounted multiarmed bandit problem," *Biometrika,* vol. 66, no. 3, pp. 561-565, 1979.

[146] V. Krishnamurthy, and R. J. Evans, "Hidden Markov model multiarm bandits: A methodology for beam scheduling in multitarget tracking," *IEEE Transactions on Signal Processing,* vol. 49, no. 12, pp. 2893-2908, 2001.

[147] N. O. Song, and D. Teneketzis, "Discrete search with multiple sensors," *Mathematical Methods of Operations Research,* vol. 60, no. 1, pp. 1-13, 2004.

[148] L. Keqin, and Z. Qing, "A restless bandit formulation of opportunistic access: Indexablity and index policy," *Proceedings of the IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks,* pp. 1-5, 2008.

[149] N. G. Pavlidis, N. M. Adams, D. Nicholson, and D. J. Hand, "Prospects for bandit solutions in sensor management," *The Computer Journal*, vol. 53, no. 9, pp. 1370-1383, 2010.

[150] Z. Feng, S. Jaewon, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61-72, 2002.

[151] A. S. Chhetri, D. Morrell, and A. Papandreou-Suppappola, "Scheduling multiple sensors using particle filters in target tracking," *Proceedings of the IEEE Workshop on Statistical Signal Processing*, pp. 549-552, 2003.

[152] A. Ryan, and J. K. Hedrick, "Particle filter based information-theoretic active sensing," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 574-584, 2010.

[153] P. Melville, M. Saar-Tsechansky, F. Provost, and R. Mooney, "An expected utility approach to active feature-value acquisition," *Proceedings of the IEEE International Conference on Data Mining*, pp. 745-748, 2005.

[154] P. Melville, F. Provost, M. Saar-Tsechansky, and R. Mooney, "Economical active feature-value acquisition through expected utility estimation," *Proceedings of the ACM International Workshop on Utility-based Data Mining*, pp. 10-16, 2005.

[155] A. Kapoor, and E. Horvitz, "Breaking boundaries: Active information acquisition across learning and diagnosis," *Advances in Neural Information Processing Systems*, pp. 898-906, 2009.

[156] V. S. Sheng, and C. X. Ling, "Feature value acquisition in testing: A sequential batch test algorithm," *Proceedings of the International Conference on Machine learning*, pp. 809-816, 2006.

[157] S. K. Nayar, and V. Branzoi, "Adaptive dynamic range imaging: Optical control of pixel exposures over space and time," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1168-1175, 2003.

[158] H. Mannami, R. Sagawa, Y. Mukaigawa, T. Echigo, and Y. Yagi, "Adaptive dynamic range camera with reflective liquid crystal," *Journal of Visual Communication and Image Representation*, vol. 18, no. 5, pp. 359-365, 2007.

[159] M. P. Christensen, G. W. Euliss, M. J. McFadden, K. M. Coyle, P. Milojkovic, M. W. Haney, J. van der Gracht, and R. A. Athale, "Active-eyes: An adaptive pixel-by-pixel image-segmentation sensor architecture for high-dynamic-range hyperspectral imaging," *Applied Optics*, vol. 41, no. 29, pp. 6093-6103, 2002.

[160] J. Kinast, and M. Gehm, "Adaptive dynamic range matching for spectroscopic measurements," *Applied Optics*, vol. 48, no. 10, pp. 1891-1897, 2009.

[161] B. Gedik, L. Liu, and P. S. Yu, "ASAP: An adaptive sampling approach to data collection in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 12, pp. 1766-1783, 2007.

[162] A. Jain, and E. Y. Chang, "Adaptive sampling for sensor networks," *Proceeedings of the International Workshop on Data Management for Sensor Networks*, pp. 10-16, 2004.

[163] R. Willett, A. Martin, and R. Nowak, "Backcasting: Adaptive sampling for sensor networks," *Proceedings of the International Symposium on Information Processing in Sensor Networks*, pp. 124-133, 2004.

[164] R. Castro, C. Kalish, R. Nowak, R. Qian, T. Rogers, and X. Zhu, "Human active learning," *Advances in Neural Information Processing Systems*, pp. 241-248, 2008.

[165] C. E. Priebe, D. J. Marchette, and D. M. Healy, "Integrated sensing and processing decision trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 699-708, 2004.

[166] 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.

[167] D. Waagen, H. A. Schmitt, and N. Shah, "Activities in integrated sensing and processing," *Proceedings of the IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 295-300, 2004.

[168] L. Rokach, *Data mining with decision trees: Theory and applications*, 1st ed., Singapore: World scientific, 2007.

[169] T. Nakamoto, S. Ustumi, N. Yamashita, T. Moriizumi, and Y. Sonoda, "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.

[170] T. Nakamoto, N. Okazaki, and H. Matsushita, "Improvement of optimization algorithm in active gas/odor sensing system," *Sensors and Actuators A: Physical*, vol. 50, no. 3, pp. 191-196, 1995.

[171] M. J. Wenzel, A. Mensah-Brown, F. Josse, and E. E. Yaz, "Online drift compensation for chemical sensors using estimation theory," *IEEE Sensors Journal*, vol. 11, no. 1, pp. 225-232, 2011.

[172] R. Lomasky, C. Brodley, M. Aernecke, D. Walt, and M. Friedl, "Active Class Selection," *Proceedings of the 18th European conference on Machine Learning*, pp. 640-647, 2007.

[173] M. E. Gehm, and J. Kinast, "Adaptive spectroscopy: towards adaptive spectral imaging," *Proceedings of SPIE*, pp. 1-11, 2008.

[174] R. Gutierrez-Osuna, "Pattern analysis for machine olfaction: A review," *IEEE Sensors Journal*, vol. 2, no. 3, pp. 189-202, 2002.

[175] S. Marco, and A. Gutierrez-Galvez, "Signal and data processing for machine olfaction and chemical sensing: A review," *IEEE Sensors Journal*, vol. 12, no. 11, pp. 3189-3214, 2012.

[176] 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.

[177] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1-38, 1977.

[178] D. M. Titterington, A. F. M. Smith, and U. E. Makov, *Statistical analysis of finite mixture distributions*, 1st ed., New York: Wiley, 1985.

[179] Y. Bengio, and P. Frasconi, "An input output HMM architecture," *Advances in Neural Information Processing Systems*, pp. 427-434, 1995.

[180] L. Rabiner, and B. Juang, "An introduction to hidden Markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4-16, 1986.

[181] K. P. Murphy, "Dynamic Bayesian networks: Representation, inference and learning," Ph.D. Dissertation, University of California, Berkeley, 2002.

[182] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," *Proceedings of the International Conference on Machine Learning*, pp. 194-202, 1995.

[183] J. Catlett, "On changing continuous attributes into ordered discrete attributes," *Proceedings of the European Working Session on Machine Learning*, pp. 164-178, 1991.

[184] K. Murphy, "The Bayes net toolbox for matlab," *Computing Science and Statistics*, vol. 33, no. 2, pp. 1024-1034, 2001.

[185] K. D. Schierbaum, U. Weimar, and W. Gopel, "Multicomponent gas analysis: An analytical chemistry approach applied to modified SnO2 sensors," *Sensors and Actuators B: Chemical*, vol. 2, no. 1, pp. 71-78, 1990.

[186] H. Sundgren, F. Winquist, I. Lukkari, and I. Lundstrom, "Artificial neural networks and gas sensor arrays: Quantification of individual components in a gas mixture," *Measurement Science and Technology*, vol. 2, no. 5, pp. 464-469, 1991.

[187] X. Wang, W. P. Carey, and S. S. Yee, "Monolithic thin-film metal-oxide gas-sensor arrays with application to monitoring of organic vapors," *Sensors and Actuators B: Chemical*, vol. 28, no. 1, pp. 63-70, 1995.

[188] E. Llobet, R. Ionescu, S. Al-Khalifa, J. Brezmes, X. Vilanova, X. Correig, N. Barsan, and J. W. Gardner, "Multicomponent gas mixture analysis using a single tin oxide sensor and dynamic pattern recognition," *IEEE Sensors Journal*, vol. 1, no. 3, pp. 207-213, 2001.

[189] M. C. Burl, B. J. Doleman, A. Schaffer, and N. S. Lewis, "Assessing the ability to predict human percepts of odor quality from the detector responses of a conducting polymer composite-based electronic nose," *Sensors and Actuators B: Chemical*, vol. 72, no. 2, pp. 149-159, 2001.

[190] A. Hierlemann, U. Weimar, G. Kraus, M. Schweizer-Berberich, and W. Gopel, "Polymer-based sensor arrays and multicomponent analysis for the detection of hazardous oragnic vapours in the environment," *Sensors and Actuators B: Chemical*, vol. 26, no. 1, pp. 126-134, 1995.

[191] K. Domansky, D. L. Baldwin, J. W. Grate, T. B. Hall, J. Li, M. Josowicz, and J. Janata, "Development and calibration of field-effect transistor-based sensor array for measurement of hydrogen and ammonia gas mixtures in humid air," *Analytical Chemistry*, vol. 70, no. 3, pp. 473-481, 1998.

[192] W. P. Carey, and S. S. Yee, "Calibration of nonlinear solid-state sensor arrays using multivariate regression techniques," *Sensors and Actuators B: Chemical*, vol. 9, no. 2, pp. 113-122, 1992.

[193] G. Huyberechts, P. Szecowka, J. Roggen, and B. W. Licznerski, "Simultaneous quantification of carbon monoxide and methane in humid air using a sensor array and an artificial neural network," *Sensors and Actuators B: Chemical*, vol. 45, no. 2, pp. 123-130, 1997.

[194] C. Di Natale, F. A. M. Davide, A. D'Amico, A. Hierlemann, J. Mitrovics, M. Schweizer, U. Weimar, and W. Gopel, "A composed neural network for the recognition of gas mixtures," *Sensors and Actuators B: Chemical*, vol. 25, no. 1, pp. 808-812, 1995.

[195] M. Pardo, G. Faglia, G. Sberveglieri, M. Corte, F. Masulli, and M. Riani, "A time delay neural network for estimation of gas concentrations in a mixture," *Sensors and Actuators B: Chemical,* vol. 65, no. 1-3, pp. 267-269, 2000.

[196] S. De Vito, A. Castaldo, F. Loffredo, E. Massera, T. Polichetti, I. Nasti, P. Vacca, L. Quercia, and G. Di Francia, "Gas concentration estimation in ternary mixtures with room temperature operating sensor array using tapped delay architectures," *Sensors and Actuators B: Chemical,* vol. 124, no. 2, pp. 309-316, 2007.

[197] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*, 1st ed., Cambridge: The MIT Press, 2005.

[198] C. Walters, A. Keeney, C. T. Wigal, C. R. Johnston, and R. D. Cornelius, "The spectrophotometric analysis and modeling of sunscreens," *Journal of Chemical Education,* vol. 74, no. 1, pp. 99-101, 1997.

[199] P. K. Clifford, and D. T. Tuma, "Characteristics of semiconductor gas sensors II. Transient response to temperature change," *Sensors and Actuators,* vol. 3, pp. 255-281, 1982.

[200] S. Wlodek, K. Colbow, and F. Consadori, "Signal-shape analysis of a thermally cycled tin-oxide gas sensor," *Sensors and Actuators B: Chemical,* vol. 3, no. 1, pp. 63-68, 1991.

[201] A. Doucet, and V. Tadic, "Parameter estimation in general state-space models using particle methods," *Annals of the Institute of Statistical Mathematics,* vol. 55, no. 2, pp. 409-422, 2003.

[202] F. Hossein-Babaei, S. M. Hosseini-Golgoo, and A. Amini, "Extracting discriminative information from the Padé-Z-transformed responses of a temperature-modulated chemoresistive sensor for gas recognition," *Sensors and Actuators B: Chemical,* vol. 142, no. 1, pp. 19-27, 2009.

[203] I. Montoliu, R. Tauler, M. Padilla, A. Pardo, and S. Marco, "Multivariate curve resolution applied to temperature-modulated metal oxide gas sensors," *Sensors and Actuators B: Chemical,* vol. 145, no. 1, pp. 464-473, 2010.

[204] F. Hossein-Babaei, and A. Amini, "A breakthrough in gas diagnosis with a temperature-modulated generic metal oxide gas sensor," *Sensors and Actuators B: Chemical,* vol. 166, pp. 419-425, 2012.

[205] M. F. Huber, T. Bailey, H. Durrant-Whyte, and U. D. Hanebeck, "On entropy approximation for Gaussian mixture random vectors," *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems,* pp. 181-188, 2008.

[206] J. Denzler, and C. M. Brown, "Information theoretic sensor data selection for active object recognition and state estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 145-157, 2002.

[207] T. C. Pearce, and M. Sanchez-Montanes, "Chemical sensor array optimization: Geometric and information theoretic approaches," *Handbook of Artificial Olfaction Machines*, pp. 347-376, 2003.

[208] B. Raman, D. C. Meier, J. K. Evju, and S. Semancik, "Designing and optimizing microsensor arrays for recognizing chemical hazards in complex environments," *Sensors and Actuators B: Chemical*, vol. 137, no. 2, pp. 617-629, 2009.

[209] J. Fonollosa, L. Fernández, R. Huerta, A. Gutiérrez-Gálvez, and S. Marco, "Temperature optimization of metal oxide sensor arrays using mutual information," *Sensors and Actuators B: Chemical*, no. 0, 2012.

[210] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, 2nd ed., New York: Wiley, 2001.

[211] L. Jianhua, "Divergence measures based on the Shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145-151, 1991.

[212] D. D. Lewis, "Feature selection and feature extraction for text categorization," *Proceedings of the Workshop on Speech and Natural Language*, pp. 212-217, 1992.

[213] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 537-550, 1994.

[214] N. Kwak, and C. H. Choi, "Input feature selection for classification problems," *IEEE Transactions on Neural Networks*, vol. 13, no. 1, pp. 143-159, 2002.

[215] P. Hanchuan, L. Fuhui, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226-1238, 2005.

[216] F. Fleuret, "Fast binary feature selection with conditional mutual information," *The Journal of Machine Learning Research*, vol. 5, pp. 1531-1555, 2004.

[217] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. Dissertation, The University of Waikato, Hamilton, 1999.

[218] I. H. Witten, and E. Frank, *Data mining: Practical machine learning tools and techniques*, 2nd ed., San Francisco: Morgan Kaufmann, 2005.

[219] C. M. Bishop, *Pattern recognition and machine learning*, New York: Springer, 2006.

[220] K. Fukunaga, *Introduction to statistical pattern recognition*, 2nd ed., London: Academic Press, 1990.

[221] P. M. Narendra, and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Transactions on Computers*, vol. 100, no. 9, pp. 917-922, 1977.

[222] W. Siedlecki, and J. Sklansky, "On automatic feature selection," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 2, no. 2, pp. 197-220, 1988.

[223] Z. Li, D. Lin, and X. Tang, "Nonparametric discriminant analysis for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 755-761, 2009.

[224] D. Coppersmith, and S. Winograd, "Matrix multiplication via arithmetic progressions," *Journal of symbolic computation*, vol. 9, no. 3, pp. 251-280, 1990.

[225] A. Guo, "Decision-theoretic active sensing for autonomous agents," *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 1002-1003, 2003.

[226] R. M. Castro, and R. D. Nowak, "Minimax bounds for active learning," *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 2339-2353, 2008.

[227] I. Rish, "An empirical study of the naive Bayes classifier," *Proceedings of the International Joint Conference on Artifical Intelligence*, pp. 41-46, 2001.

[228] B. Raman, J. L. Hertz, K. D. Benkstein, and S. Semancik, "Bioinspired methodology for artificial olfaction," *Analytical Chemistry*, vol. 80, no. 22, pp. 8364-8371, 2008.

APPENDIX A: EXPERIMENTAL APPARATUS

This appendix describes the experimental apparatus (PRISM e-nose) used for collecting experimental data for this dissertation.

PRISM e-nose

The first prototype of PRISM e-nose was developed by Agustin Gutierrez [69] for his dissertational work. In spring 2010, a group of three undergraduate students (Brock Spratlen, Brandon Anderson, and Kyle Minton) from the department of Computer Science and Engineering deconstructed this prototype to develop a more compact and automated experimental apparatus. However, most of the components from the original prototype were still utilized. The three students worked on this project to fulfill the requirements for the course CPSC-483: Computer System Design and I worked as a supervisor for this project.

The experimental apparatus consists of three main components:

- Sensor chamber
- Delivery system
- Control and instrumentation

We describe these three components in a greater detail in the following sections.

Sensor chamber

The sensor chamber, as shown in Figure 49, was custom-built to accommodate eight sensors of TO-5 type packaging. We installed seven MOX sensors in this chamber that were purchased from Figaro. These include one TGS-2602, two TGS-2610, two TGS-2611, and one TGS-2620 sensors. The eighth slot was used for a temperature/humidity sensor (Honeywell HIH-4602-A).

Figure 49 Close-up of the sensor chamber. This view displays four MOX sensors mounted on one side of the chamber. The other side contains three other MOX sensors and a temperature/humidity sensor.

Delivery system

The delivery system consists of the following components: an air pump, glass vials, a valve manifold, gas diluters, air filter, and a flow buffer. The gases were delivered to sensor chamber from 30ml glass vials using an air pump connected downstream. The valve manifold was installed with 8 solenoid-valves, which allowed us to choose between 8 different vials and thus determine which chemical was delivered to the chamber. The output from the valve manifold is connected to a series of Model 1010 Precision Gas diluters. These diluters, which are manufactured by Custom Sensor Solutions, contain an electronically controlled valve that draws alternately from two sources: (1) chemical and (2) diluent (air), and mixes them in a small chamber. The diluent (air) source is connected through air filters containing desiccants to reduce humidity. The output of the diluters is connected to a flow buffer to reduce oscillations in the gas flow. The flow buffer consists of two 30ml glass vials connected to each other with a short piece of vinyl tubing. The output from the diluters is connected as an input to one of the two vials in the flow buffer. The output of the other vial is connected to the sensor chamber. A schematic of the entire experimental apparatus is shown in Figure 50.

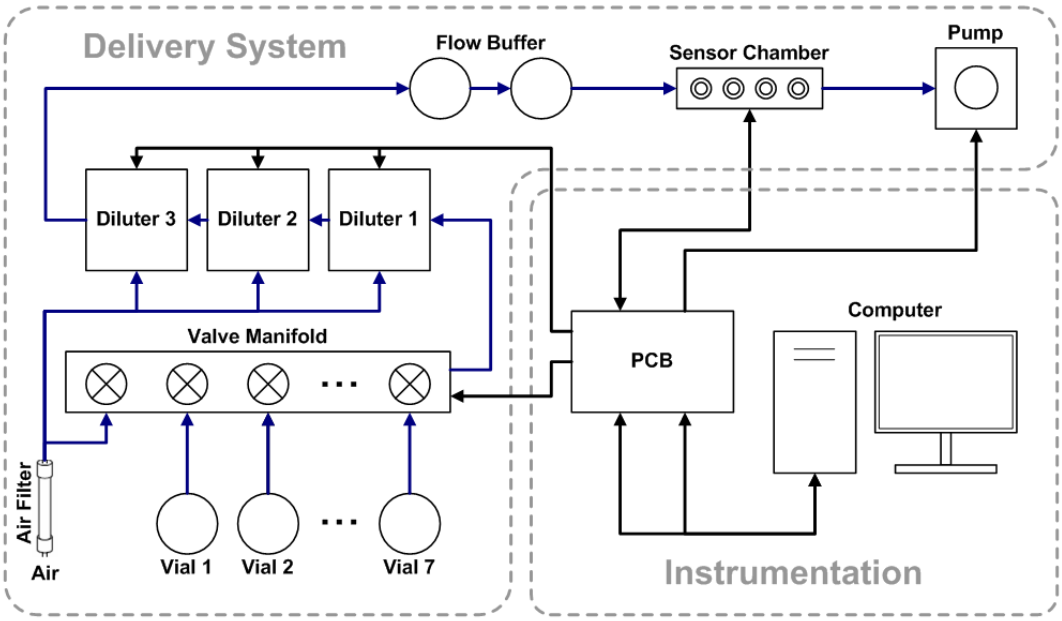

Figure 50 A schematic of the PRISM e-nose. The blue arrows denote the transfer of chemicals across the system. The black arrows represent transfer of data between the experimental apparatus, and the computer.

Control and instrumentation

The entire apparatus (sensors, diluters, pump, and valves) is controlled through an interface circuit (PCB). The PCB contains the circuitry required to operate the 7 MOX sensors, 8 solenoid valves, air pump, and the temperature/humidity sensor. The PCB also contains terminal connections required to connect the system components (sensors, valves, pump, and diluters) with the two data acquisition cards: NI DAQ PCI-6713 and PCI-6024e. The PCB's dimensions were 6"x3.5" and was manufactured by ExpressPCB. The entire apparatus was mounted on a flat base made of general purpose plywood of dimensions 25"x19"x¾". Figure 51 shows a picture of the experimental apparatus from the top view.

Figure 51 A top-view picture of the assembled experimental apparatus.

Circuitry

The Figaro MOX sensors have a very simple design consisting of two resistors (heater and sensor) packed into a standard TO-5 metal can. The measuring circuit for these sensors (as shown in Figure 52) is a voltage divider, where the sensor resistor is connected in series with a load resistor R_L, which in turn is connected to the ground. We apply a fixed circuit voltage V_c across the sensor and the load resistors, and a variable heater voltage V_H across the heater resistor. The sensor resistance R_s can be measured as:

$$R_s = \frac{R_L V_S}{V_c - V_s} = \frac{R_L (V_c - V_L)}{V_L} \tag{57}$$

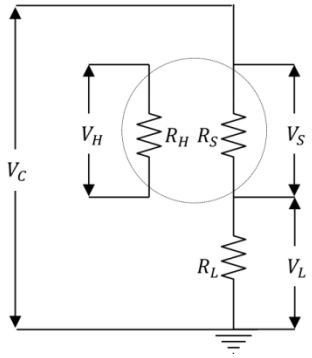

Figure 52 The measurement circuit for MOX sensors. In the experimental apparatus, we set $V_C = 10$V and used $R_L = 5$kΩ resistors. R_H is dependent on the sensor but typically in the range of 50 to 100 Ω.

The control circuitry for valves converts the digital output from data acquisition card into the necessary voltages (Figure 53). Namely, when the digital output is set to 0 or 1, the circuit produces a constant 0 V or 12 V output respectively. The air pump was also controlled with a similar circuit.

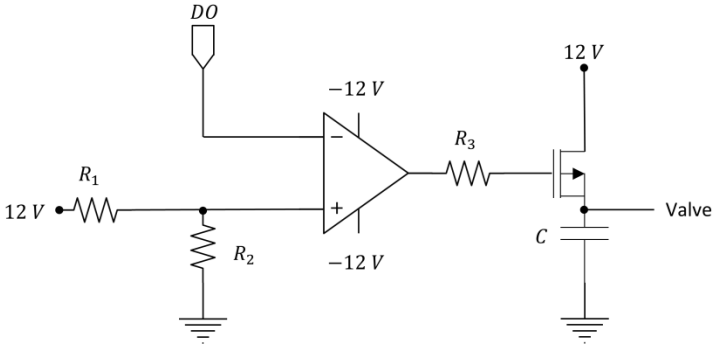

Figure 53 Control circuit for the valves. In the experimental apparatus, the three resistor values were $R_1 =$ 10kΩ , $R_2 = 3$kΩ, $R_3 = 10$kΩ, the capacitor value was $C = 100$pF, and DO represents the digital output form the data acquisition card.

The PCB board also included heater driving circuitry which was necessary to supply enough current to the sensors (Figure 54). This circuit is a voltage follower; it ensures the analog output from the data acquisition card is applied across the sensor's heating resistor.

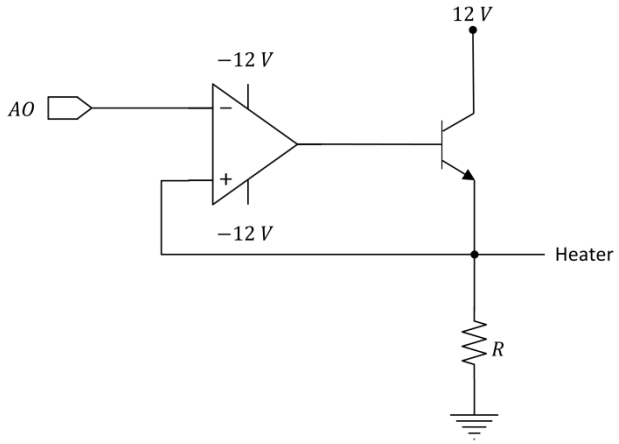

Figure 54 Heater driving circuit ($R = 10\text{k}\Omega$).

APPENDIX B: LISTS OF CHEMICALS

Table 9 lists the chemicals from the experiments in section 4.2.

Table 9 List of chemicals from Figure 12.

Number	Chemical name
1.	Trans-2-(9h-purin-9-yl)-cyclohexanol
2.	Esterethyl2-tert-butyl-acidcarbanilic
3.	2-chloro-4-tert-butyl-acetanilide
4.	Hydrochloride2-propen-1-amine
5.	2-acetamido-2-deoxy-alphad-glucopyranoside
6.	9-methyl-9h-purine-6(1h)-thione
7.	Estertri(2-methylpropyl)acid123-propenetricarboxylic
8.	2-chloro-6-dimethylaminopurine
9.	n-(37-dinitro-9-oxofluoren-2-yl)-acetamide
10.	Bis-44-(2-chloro-methyleneaniline)
11.	Tri(acetylricinoleate)glyceryl
12.	Bis[4-(1133-tetramethylbutyl)]-11-oxy-benzene
13.	Bis(44'-dimethylaminophenyl)-methane
14.	(Ethylamino)-5-amino-4-pyrimidine
15.	p-di-tert-butylbenzene
16.	Hydrochloride3-(2-benzylamino)ethyl-3-phenyl-indol-2(3h)-one
17.	26-dimethyl-p-dioxane
18.	113-trimethoxy-butane
19.	2-(diethylamino)-fluoren-9-ol
20.	8-(methylsulfonyl)-purine
21.	8-ethyl-6-mercapto-9h-purine
22.	(34-dihydroxyphenyl)-acidacetic
23.	Triethanolamine
24.	n-tert-butyl-o-toluidine
25.	4-methyl-n-tert-butyl-aniline
26.	Hydrochloride3-methyl-1-phenyl-138-triazaspiro[45]decan-4-one
27.	3-methyl-n-tert-butyl-aniline
28.	Formanilide

Table 9 Continued

Number	Chemical name
29.	nn-dimethylbenzylamine
30.	4'-bromoacetanilide
31.	910-diformamido-anthracene
32.	hydrochloride26-dimethyl-1-phenyl-4(1h)-pyridone
33.	4-phenylamino-triazolyl-5-carboxamide
34.	2-phenyl-1-hydrazide5-nitro-acid2-furoic
35.	1-phenylhydrazide5-nitro-acid2-furoic
36.	3-iodo-5-nitroaniline
37.	1-[2-(dimethylamino)ethyl]-4-methyl-piperazine
38.	n-beta-cyanoethyl-acetanilide
39.	2-amino3-[35-diiodo-4-[4-hydroxy-3-(1-methylethyl)phenoxy]phenyl]-acidpropanoic'
40.	Phenyl)-1-propanol3-(o-methoxy)
41.	Esterdi-(2-methylpropyl)acidnonanedioic
42.	esterdi-isooctylacidadipic
43.	Esterdibutylacidhexanedioic
44.	nn-diallyl-isopropylamine
45.	Vinyl6-methylheptylether
46.	17alpha-methyl-beta-nortestosterone
47.	14-bis-(23-epoxypropyl)-25-dimethyl-piperazine
48.	Dioctanoateglycoltriethylene
49.	12-hydroxy-acidstearic
50.	25-dimethyl-morpholine

Table 10 lists the chemicals from the experiments in section 6.2.

Table 10 List of chemicals from Figure 38.

Number	Chemical name
1.	9-cyclohex-2-enyl-6-hydrazino-9h-purine
2.	Esterethyl2-tert-butyl-acidcarbanilic
3.	Hydrochloride2-propen-1-amine

Table 10 Continued

Number	Chemical name
4.	2-acetamido-2-deoxy-alphad-glucopyranoside
5.	9-methyl-9h-purine-6(1h)-thione
6.	Estertri(2-methylpropyl)acid123-propenetricarboxylic
7.	2-chloro-6-dimethylaminopurine
8.	n-(37-dinitro-9-oxofluoren-2-yl)-acetamide
9.	3-chloro-diphenylamine
10.	tri(acetylricinoleate)glyceryl
11.	Bis[4-(1133-tetramethylbutyl)]-11'-oxy-benzene
12.	Bis(44'-dimethylaminophenyl)-methane
13.	(ethylamino)-5-amino-4-pyrimidine
14.	Hydrochloride3-(2-benzylamino)ethyl-3-phenyl-indol-2(3h)-one
15.	6-dimethylamino-2-methylamino-purine
16.	26-dimethyl-p-dioxane
17.	26-methano-14(2h)-benzoxazocin-5(6h)-one4(3h)-cyclohexyl-6-phenyl-
18.	2-(diethylamino)-fluoren-9-ol
19.	8-(methylsulfonyl)-purine
20.	7-chloro-1245-tetrahydro-14-dimethyl-3h-14-benzodiazepine-25-dione
21.	8-chloro-1245-tetrahydro-14-dimethyl-3h-14-benzodiazepine-25-dione
22.	8-ethyl-6-mercapto-9h-purine
23.	Tris(mercaptoacetae)2-(hydroxymethyl)-2-methyl-13-propanediol
24.	Tetrakis(mercaptoacetate)pentaerythritol
25.	(34-dihydroxyphenyl)-acidacetic
26.	4-methyl-n-tert-butyl-aniline
27.	Hydrochloride3-methyl-1-phenyl-138-triazaspiro[45]decan-4-one
28.	Triallyl-s-triazine-246-(1h3h5h)-trione
29.	3-methyl-n-tert-butyl-aniline
30.	44'-malonyldi-morpholine
31.	Phenylhydrazoneacetone
32.	Formanilide
33.	nn-dimethylbenzylamine
34.	4'-bromoacetanilide
35.	910-diformamido-anthracene
36.	1245-tetrahydro-1-methyl-4-phenyl-3h-14-benzodiazepine-25-dione
37.	Hydrochloride26-dimethyl-1-phenyl-4(1h)-pyridone
38.	3-chloro-2-methyl-2-propenenitrile
39.	4-(p-methoxyphenyl)-1-methyl-1245-tetrahydro-3h-14-benzodiazepine

Table 10 Continued

Number	Chemical name
40.	5h-dibenzo[ad]-1-cyclohepten-5-ol
41.	4-phenylamino-triazolyl-5-carboxamide
42.	2-phenyl-1-hydrazide5-nitro-acid2-furoic
43.	1-phenylhydrazide5-nitro-acid2-furoic
44.	3-iodo-5-nitroaniline
45.	1-[2-(dimethylamino)ethyl]-4-methyl-piperazine
46.	13-bis(anilino)propane
47.	n-beta-cyanoethyl-acetanilide
48.	5-oxo-hexanenitrile
49.	Benzanilide3'4'5'-trimethoxy-2-methylamino-
50.	2-amino3-[35-diiodo-4-[4-hydroxy-3-(1-methylethyl)phenoxy]phenyl]-acidpropanoic

APPENDIX C: LIST OF PUBLICATIONS

Following is a list of publications pertaining to this dissertation.

Journal articles:

1. ‘Active temperature programming for metal-oxide chemoresistors.’ R. Gosangi and R. Gutierrez-Osuna, IEEE Sensors Journal, 10(6), 1075-1082, 2010.
2. ‘Active concentration-independent chemical identification with a tunable infrared sensor.’ J. Huang, R. Gosangi, and R. Gutierrez-Osuna, IEEE Sensors Journal, 12(11), 3135-3142, 2012.
3. ‘Active temperature modulation of metal-oxide sensors for quantitative analysis of gas mixtures.’ R. Gosangi, and R. Gutierrez-Osuna, Sensors and Actuators B: Chemical, 185, 201-210, 2013.

Conference proceedings:

1. ‘Active chemical sensing with Partially Observable Markov Decision Processes.’ R. Gosangi and R. Gutierrez-Osuna, International Symposium on Olfaction and Electronic Noses 2009.
2. ‘Energy-aware active chemical sensing.’ R. Gosangi and R. Gutierrez-Osuna, IEEE Sensors Conference 2010.
3. ‘Data-driven modeling of metal-oxide sensors with dynamic Bayesian networks.’ R. Gosangi and R. Gutierrez-Osuna, International Symposium on Olfaction and Electronic Noses 2011.
4. ‘Quantification of gas mixtures with active recursive estimation.’ R. Gosangi and R. Gutierrez-Osuna, International Symposium on Olfaction and Electronic Noses 2011.
5. ‘Active Sensing with Fabry-Perot infrared interferometers.’ J. Huang, R. Gosangi, and R. Gutierrez-Osuna, International Symposium on Olfaction and Electronic Noses 2011.

6. ‘Advances in Active and Adaptive Chemical Sensing.’ R. Gutierrez-Osuna, R. Gosangi, and A. Hierlemann, Invited paper, International Symposium on Olfaction and Electronic Noses 2011.