STATISTICAL MODEL-BASED METHODS FOR OBSERVATION SELECTION IN WIRELESS SENSOR NETWORKS AND FOR FEATURE SELECTION IN CLASSIFICATION

A Dissertation

Presented to

the Faculty of the Graduate School

University of Missouri

In Partial Fulfillment

Of the Requirements for the Degree

Doctor of Philosophy

---

by

QI QI

Dr. Yi Shang, Dissertation Supervisor

MAY 2012

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled

STATISTICAL MODEL-BASED METHODS FOR OBSERVATION SELECTION IN WIRELESS SENSOR NETWORKS AND FOR FEATURE SELECTION IN CLASSIFICATION

presented by Qi Qi

A candidate for the degree of Doctor of Philosophy,

and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Yi Shang _____

Dr. Michael Jurczyk _____

Dr. Jianlin Cheng _____

Dr. Tony X. Han _____

I dedicate my dissertation to
my mother, Yulan Wu,
my father, Songren Qi.

ACKNOWLEDGEMENTS

I feel very grateful to my esteemed advisor Dr. Yi Shang for his extensive help, wonderful guidance and generous support, by all of which it enables me to complete this work.

I extend my gratitude to my advisory committee members: Dr. Michael Jurczyk, Dr. Jianlin Cheng, and Dr. Tony X. Han, for their insightful comments and suggestions on my work.

I also owe a debt of gratitude to all faculties who taught me classes and assisted me in various ways during my course studies.

I want to thank our Computer Science Department and Graduate School for offering me such a wonderful environment of studying here. Thank our great staff Trish, Jodie, Sandy and Jeff for their hospitality and plenty of help. I would like to take this opportunity to extend many thanks to my colleagues in my lab and department. I will always cherish the friendship and appreciate the help of friends here at Mizzou. It is all of you that have made the past several years of enjoyable and memorable.

Finally, I am very grateful to my wife, Ni Li and my parents. Your accompany, supports and understanding are always the greatest treasure in my life.

TABLE OF CONTENTS

LIST OF ALGORITHMS

LIST OF FIGURES

LIST OF TABLES

STATISTICAL MODEL-BASED METHODS FOR OBSERVATION SELECTION IN WIRELESS SENSOR NETWORKS AND FOR FEATURE SELECTION IN CLASSIFICATION

Qi Qi

Dr. Yi Shang, Dissertation Supervisor

ABSTRACT

Wireless sensor networks have been deployed in real world applications ranging from environmental monitoring to ambient intelligence. This technology allow us to have a better understanding of the natural environment, human activities and even their interactions. Nowadays most wireless sensors are powered by batteries. However changing batteries for thousands of sensors with human intervention is infeasible for a large scale of deployment. It has been challenging to make wireless embedded sensor networks scalable and sustainable due to its restricted power source. The optimization problem is to make fewer number of

observations for reducing relevant power consumption, in the meantime obtaining sufficient information out of the observations. When or where to make the observations directly affect on predictive accuracy for unobserved points of interest.

In the dissertation, we apply statistical model-based approaches to address the temporal and spatial sensor observation selection challenges. For sensor observation selection in time domain, we first present an improved version of VoIDP algorithm that is the first optimal algorithms for efficiently selecting the subset of observations on chain graphical models. Then we apply the time series model-based approach to a wireless sensor scheduling problem. For location-based sensor observation selection, we introduce two greedy heuristic methods by optimizing entropy and mutual information criteria based on Gaussian process models. The mutual information-based heuristic is powered by submodularity optimization that provides both efficiency and theoretical guarantee to its solution. We also demonstrate those heuristic methods in an application of placing road traffic monitoring sensors. Experimental results in a simulated environment showed that the entropy-based heuristic tends to place sensors around intersections, whereas the mutual information-based heuristic places sensors more widely and avoids repeatedly placing sensors at correlated locations on same road segments. We also compare the graphical model-based approach with the Gaussian process model-based approach for sensor observation selection, and our experimental results show that the graphical model-based approach is more robust and error-tolerant than the Gaussian process model-based approach.

Finally We also apply the mutual information-based selection method based on submodularity optimization to feature selection for classification problems. One type of the feature selection methods is to select important features only based on the characteristics of a data set, which is essentially similar to the observation selection problems. We compare the proposed method with existing state-of-the-art attribute selection methods through extensive experiments, and show that the proposed mutual information-based feature selection method perform comparably with, or even better than, other feature selection methods.

CHAPTER 1

INTRODUCTION

Wireless sensor networks have been deployed in real world applications ranging from environmental monitoring to ambient intelligence [7, 51, 52, 59]. This technology allow us to have a better understanding of the natural environment, human activities and even their interactions.

Nowadays most wireless sensors are powered by batteries. However changing batteries for thousands of sensors with human intervention is infeasible for large scale deployment of this technology. It has been challenging to make wireless embedded sensor networks scalable and sustainable. Even for tomorrow's sensors that can harvest energy from their surrounding environment, the small energy they collect and store will never be taken for granted [9]. Conversely, every sensor observation paid by precious harvested energy should be as rewarding as possible.

It is challenging to make wireless sensor networks in real world applications scalable and sustainable due to the constraint power source. The optimization problem is to make fewer number of observations for saving energy, while obtaining sufficient information out

of the observations. When or where to make the observations directly impacts the predictive accuracy for unobserved points of interest.

In Chapter 2, we introduce an observation selection method based on time series models and its application to wireless sensor scheduling. The VoIDP algorithm is the first optimal algorithm for efficiently selecting the subset of observations in chain graphical models [30]. The original VoIDP algorithm has a mistake in the process of recovering the optimal selections, and fails to produce correct outputs. In this paper, we present an improved version of the algorithm; which fixes the mistakes and verifies the solutions in experiments. Further more, we discuss some recent works in the area of subset selection problems, and present a simplified solution for computing the maximum expected total reward for a sub chain under certain circumstances.

In Chapter 3, we introduce two greedy heuristic methods based on entropy and mutual information criteria under multivariate Gaussian models for optimizing location-based observation selection. Submodularity optimization plays an important role in developing an efficient and near-optimal approximate algorithm for maximizing the mutual information criterion. Mutual information functions can be considered as submodular functions [32], and its relevant greedy algorithm guarantees that its solution is as good as at least $(1-1/e)OPT$, where OPT is an optimal solution value. We apply these methods to place road traffic monitoring sensors in a simulated road map and compare their performance. Experimental results

show that the entropy criterion- based heuristic tends to place sensors around intersections, whereas the mutual information criterion-based heuristic places sensors more widely, and it avoids repeatedly placing sensors at the correlated locations on the same road segments.

In Chapter 4, we pull the two model based approaches introduced in previous chapters into one scenario, the sensor scheduling problem, and compare their performance. The first approach is to apply the corrected VoIDP algorithm on a chain graphical model for selecting a subset of observations that minimizes the overall uncertainty. The second approach is to find a selection of observations based on Gaussian Process model that maximizes the entropy and the mutual information criteria, respectively. We compare their performances in terms of predictive accuracy for the unobserved time points based on their selections of observations. Experimental results show that the Gaussian Process model based method achieves higher predictive accuracy if sensor data are accurate. However when observations have errors, its performance degrades quickly. In contrast, the graphical model based approach is more robust and error-tolerant.

In Chapter 5, we propose to apply the observation selection methods to select features on classification problems. Selecting features for classification is similar to selecting observations in wireless sensor networks. They share purpose in common that removing redundant and noise information and selecting out the valuable information that most contribute to classification or prediction models. Observation selection based on mutual information with

submodularity optimization is efficient and effective as shown in chapter 3. Its computational efficiency and near-optimal theoretical guarantee make it promising for feature selection on classifications. We conducted extensive experiments to compare its performance with other popular feature selection methods on multiple data sets with a variety of different classifiers. The results show that the submodularity optimization inspired mutual information-based selection method is a strong competitor among other feature selection methods.

CHAPTER 2

TIME-SERIES OBSERVATION SELECTION WITH ITS APPLICATION TO WIRELESS SENSOR SCHEDULING

2.1 INTRODUCTION

A typical problem in real world applications is the optimization of information gathering. Wireless sensor networks, for example, is a powerful tool for monitoring spatio-temporal phenomena. However, its limited power source makes sensing expensive. It is a trade off between obtaining more and useful information, versus making less observations. Scheduling a sensor to turn on to observe, then to turn off to save energy is a very big optimization problem.

A graphical model-based method for selecting sensor observations in the time domain is introduced in this chapter. It selects time points with the most rewarding observational

information for scheduling a sensor's on/off state. When time series data from a sensor is modeled by a chain graphical model, e.g. a Hidden Markov Model (HMM), the method can use observations at some time to infer sensing values at another time. The optimal selection of observations is to minimize the predictive uncertainty.

Chain graphical models such as Hidden Markov Models (HMM) can be trained using data time series from sensors. The observation variables of 24 time points roll over onto the chain, if each hour in a day is treated as a time point for observation. When a selection is made at a time point for observation, the distributions of observation variables after this point will become certain to some extent. The sensor scheduling problem then turns into optimizing a subset of observations. The selection in the chain graphical model is to minimize the uncertainty overall. The VoIDP algorithm is the first optimal algorithm for efficiently selecting a subset of observations in chain graphical models [30]. It is a dynamic programming approach to optimize the value of information. However, during our evaluation of this algorithm for the subset selection problem in chain graphical models, we discovered that following the exact algorithm could not give desirable solutions. We were hence motivated to improve on it.

We identified a critical overlook in the original VoIDP algorithm; which causes the failure. We will present the improved version of VoIDP algorithm in Section 2.3. In Section 2.4, we evaluated and verified the improved VoIDP algorithm and its solutions empirically. In

Section 2.5, we discuss a situation where the computation in the algorithm can be simplified. We will give a brief review in Section 2.6 regarding some interesting works recently published in the area of optimizing the information gathering. We will start in the following Section to give a brief description of the optimization problem. For convenience, same notation will be used as does in [30].

2.2 PROBLEM STATEMENT

Battery-equipped wireless sensors are power constrained. The challenge of changing batteries for thousands of sensors hinders wireless sensor networks to become scalable and sustainable. Hence, wireless sensors need to selectively observe in order to save their energies.

The observations from a sensor can be scheduled at some time moments and it goes to sleep mode at all other time. One criterion of observation selection is to maximize its informative values. Hidden Markov Models (HMM) have been used to model sensor observations along the time dimension. Each observation variable at a time point has a distribution over some hidden states. When an observation is made at one time, its value can contribute to infer observational values at another time. But the inference accuracy depends on the selection of observations. Formally the problem of optimizing the selection of observations across the time chain can be cast in the following subset selection problem...

Given a collection of random variables $\mathcal{X}_\mathcal{V} = (\mathcal{X}_1, \ldots, \mathcal{X}_n)$. A subset of the variables, $\mathcal{X}_\mathcal{A} = (\mathcal{X}_{i_1}, \ldots, \mathcal{X}_{i_k})$ are observed as $\mathrm{x}_\mathcal{A}$. The posterior distribution $P(\mathcal{X}_\mathcal{V} \mid \mathcal{X}_\mathcal{A} = \mathrm{x}_\mathcal{A})$ can be computed and used in a total reward $R(P(\mathcal{X}_\mathcal{V} \mid \mathcal{X}_\mathcal{A} = \mathrm{x}_\mathcal{A}))$. Since observational values of $\mathcal{X}_\mathcal{A}$ are unknown, an expected total reward is used to measure the quality of the subset selection.

Hence, the observation selection problem is to select a subset $\mathcal{A}^* \subseteq \mathcal{V}$ that maximizes [30],

$$\mathcal{A}^* = \mathrm{argmax}_{\mathcal{A} \subset \mathcal{V}} \sum_{\mathrm{x}_\mathcal{A}} P(\mathcal{X}_\mathcal{A} = \mathrm{x}_\mathcal{A}) R(P(\mathcal{X}_\mathcal{V} \mid \mathcal{X}_\mathcal{A} = \mathrm{x}_\mathcal{A}))$$

The expected total reward to maximize above is the sum of all the expected local rewards $R_j(\mathcal{X}_j \mid \mathcal{X}_\mathcal{A})$, because of the conditional independency held on chain graphical models. An expected local reward equals to $\sum_{\mathrm{x}_\mathcal{A}} P(\mathcal{X}_\mathcal{A} = \mathrm{x}_\mathcal{A}) R_j(\mathcal{X}_j \mid \mathrm{x}_\mathcal{A})$. A local reward $R_j(\mathcal{X}_j \mid \mathrm{x}_\mathcal{A})$ depends on $P(\mathcal{X}_j \mid \mathcal{X}_\mathcal{A} = \mathrm{x}_\mathcal{A})$, which is the marginal distribution of variable $\mathcal{X}_j$ conditioned on the observations $\mathcal{X}_\mathcal{A} = \mathrm{x}_\mathcal{A}$. It can be further deduced based on the conditional entropy as $R_j(\mathcal{X}_j \mid \mathrm{x}_\mathcal{A}) = -H(\mathcal{X}_j \mid \mathrm{x}_\mathcal{A}) = \int P(x_j, \mathrm{x}_\mathcal{A}) \log_2 P(x_j \mid \mathrm{x}_\mathcal{A}) \mathrm{d}x_j$.

Probabilistic inference techniques on chain graphical models simplify the evaluation of local rewards. For example, a HMM based on a sensor's temperature time series data has n time points. The observation at each time point is determined by a certain number of hidden states that are used to construct underlying inference chains. Evaluation of $P(\mathcal{X}_j \mid \mathcal{X}_\mathrm{A})$ only depends on $P(\mathcal{X} \mid \mathcal{X}_{j_{close}})$, where $j_{close} \in \mathrm{A}$ is the closest observation time point before j. The conditional independence property of graphical models implies that an expected total reward

along the entire time chain can be divided into expected rewards on small sub-chains [30]. It inspired the application of a divide-and-conquer strategy, essentially a dynamic programming approach, in the original VoIDP algorithm.

2.3 Improved Efficient Algorithm for Optimal Subset Selection in Chain Graphical Models

In this section, we will first give a brief description of the original VoIDP algorithm as appearing in papers [30, 29], then, discuss the reason for and present the improved version of this algorithm.

2.3.1 Original VoIDP Algorithm

In the subset selection problem, the target is to decide a subset of the variables to observe before any observation is made in order to predict the overall observation most accurately based on the observed values of the selected variables. In the running example, before a sensor is deployed, we want to find a number of time points out of 24 to pre-schedule its sensing for a day.

The original VoIDP algorithm in [30] was claimed to be the first optimal algorithm for efficient subset selection in chain graphical models. For convenience, we have attached its pseudo code shown in algorithm 1. The algorithm implements a dynamic programming ap-

proach that is inspired by the reward decomposition property briefly discussed in Section 2.2. It also considers some other factors in the subset selection process, such as operating within a limit budget B, the cost β_j of making observations, and associated penalties C_j applied to the expected total reward. It was proved that the time complexity of this algorithm given budget B in terms of evaluations of expected local rewards is $(\frac{1}{6}n^3 + O(n^2))B$, where $n = |\mathcal{V}|$.

Algorithm 1: VoIDP algorithm for optimal subset selection (Krause and Guestrin, [30])

Input: Budget B, rewards R_j, costs β_j and penalties C_j
Output: Optimal selection $\mathcal{A}$ of observation times

begin
  for $0 \leq a < b \leq n+1$ **do** compute $L_{a:b}(0)$;
  for $k = 1$ **to** B **do**
    for $0 \leq a < b \leq n+1$ **do**
      $sel(-1) := L_{a:b}(0)$;
      for $j = a+1$ **to** $b-1$ **do**
        $sel(j) := R_j(X_j|X_j) - C_j + L_{a:j}(0) + L_{j:b}(k-\beta_j)$;
      end
      $L_{a:b}(k) = \max_{j\in\{a+1,\ldots,b-1,-1\}} sel(j)$;
      $\Lambda_{a:b}(k) = \operatorname{argmax}_{j\in\{a+1,\ldots,b-1,-1\}} sel(j)$;
    end
  end
  $a := 0; b := n+1; k := B; \mathcal{A} := \emptyset$;
  while $j \neq -1$ **do**
    $j := \Lambda_{a:b}(k)$;
    if $j \geq 0$ **then**
      $\mathcal{A} := \mathcal{A} \cup \{j\}; k := k - \beta_j$;
    end
  end
end

The original VoIDP algorithm as shown in 1 implements a dynamical programming approach to efficiently select an optimal subset of the variables to observe in chain graphical models. The algorithm is named as "VoIDP" because of its use of Dynamic Programming to optimize the information value.

The main body of the algorithm is to compute a number of tables of $L_{a:b}(k)$ which is denoted as the optimal expected total reward that can be achieved for the sub-chain $a:b$ with the budget k. And $L_{0:n+1}(B)$, therefore, denotes the optimal expected total reward for the entire chain with full budget B, while $L_{a:b}(0)$ is the total reward without any additional observations. The $\Lambda_{a:b}(k)$ stores the choice that realizes $L_{a:b}(k)$. The choices could be either the index of next variable to select or -1, which means no variable should be selected. In the innermost loop, $sel(j)$ is the expected total reward for the sub chain $a:b$ obtained by observing at j, and $sel(-1)$ is the reward if no observation is made. The optimal solution of subset selection is obtained by tracing out the quantities in $\Lambda_{a:b}(k)$.

When we evaluated the algorithm, however, it failed to give desirable outputs (see table 2.1) in the experiment. And we were thus motivated to improve on it. Following is an improved version of the VoIDP algorithm.

2.3.2 IMPROVED VOIDP ALGORITHM

The core part of VoIDP algorithm is to recursively compute the optimal expected total reward $L_{a:b}(k)$ for the sub chain $a:b$ using the budget k. The base case is simply $L_{a:b}(0)$, and the recursion for $L_{a:b}(k)$ is either $L_{a:b}(0)$ or $\max_{a<j<b,\beta_j\leq k}\{sel(j)\}$. It means that we can choose not to spend any more of the budget to reach the base case, or we can select the optimal observation at j, which depends on the obtained expected total rewards. In our experiments, we let reward penalty C_j be zero, and let selection cost β_j be one. In this situation, the computation of $L_{a:b}(k)$ can actually be further simplified. We will discuss this in more details in Section 2.5.

According to the reward decomposing property (see in [30]), selecting an observation will divide the computation of expected total reward of the chain into expected total reward computations along the two sub chains separated by the observation. This is reflected in the equation (2.1) of computing $sel(j)$ which is the expected total reward for chain $a:b$ when making an observation at j.

$$sel(j) := R_j(\mathcal{X}_j \mid \mathcal{X}_j) - C_j + L_{a:j}(0) + L_{j:b}(k-\beta_j) \tag{2.1}$$

The total reward of observing at j for the chain $a:b$ is the sum of the reward of observing j at itself, the optimal total reward achieved for the sub-chain $a:j$ without any spending and

the optimal total reward achieved for the sub-chain $j:b$ with the budget $k-\beta_j$ and minus the reward penalty C_j. In this way, all the selected observations will fall into one side $j:b$, and on the other side $a:j$ there is no selection yet. The variable selection candidate j separates these into two sides. After all the tables $L_{a:b}(k)$ and $\Lambda_{a:b}(k)$ are computed, we can trace back to find all the optimal selections in $\Lambda_{a:b}(k)$. One key point here is that after we find an optimal selection at j, the entry for locating the next optimal selection should be at $\Lambda_{j:b}(k-\beta_j)$. The pseudo code of the improved VoIDP algorithm is illustrated in algorithm 2. For convenience, we use the same notations as in algorithm 1.

Algorithm 2 shows the corrected VoIDP. The first part of the algorithm from line 2 to 12 dynamically computes a three-dimension (a, b and k) table, where a and b are denoting the two ends of a chain, and k is the budget. Each cell in the table is a tuple of two elements: $L_{a:b}(k)$ and $\Phi_{a:b}(k)$. $L_{a:b}(k)$ denotes the optimal expected total reward that can be achieved for the sub-chain $a:b$ with the budget k. $sel(j)$ is the expected total reward for the sub chain $a:b$ obtained after observing at the time point j, and $sel(0)$ is the reward when no observation is made. $\Phi_{a:b}(k)$ stores the j that maximizes the $sel(j)$ value. In the first part, it recursively computes optimal expected total rewards $L_{a:b}(k)$ for the sub chain $a:b$ given a budget k. The base case is simply $L_{a:b}(0)$, and the recursion for $L_{a:b}(k)$ is either $L_{a:b}(0)$ or $\max_{a<j<b, \beta_j \leq k} \{sel(j)\}$, which means that we can choose not to spend any more of the budget to reach the base case, or select the optimal observation at j with the cost β_j. C_j denotes

Algorithm 2: Improved VoIDP algorithm for optimizing observation selection on chain graphical models

Input: Budget B, rewards R_j, costs β_j and penalties C_j ($j \in \mathcal{V}, |\mathcal{V}| = n$)
Output: Optimal selection of observations at $\mathcal{A}$

begin
  for $0 \leq a < b \leq n+1$ **do** compute $L_{a:b}(0)$
  for $k = 1$ **to** B **do**
    for $0 \leq a < b \leq n+1$ **do**
      $sel(0) \leftarrow L_{a:b}(0)$
      for $j = a+1$ **to** $b-1$ **do**
        $sel(j) \leftarrow R_j(\mathcal{X}_j|\mathcal{X}_j) - C_j + L_{a:j}(0) + L_{j:b}(k-\beta_j)$
      end
      $L_{a:b}(k) \leftarrow \max_{j\in\{0,a+1,\ldots,b-1\}} sel(j)$
      $\Phi_{a:b}(k) \leftarrow \text{argmax}_{j\in\{0,a+1,\ldots,b-1\}} sel(j)$
    end
  end
  $a \leftarrow 0; b \leftarrow n+1$
  $k \leftarrow B$
  $\mathcal{A} \leftarrow \emptyset$
  repeat
    $y \leftarrow \Phi_{a:b}(k)$
    if $y > 0$ **then**
      $\mathcal{A} \leftarrow \mathcal{A} \cup \{y\}$
      $\mathbf{a} \leftarrow \mathbf{y}$
      $k \leftarrow k - \beta_y$
    else
      break
    end
  until $k \leq 0$
end

penalty on the reward by selecting at j. In our experiments, we let C_j be zero, and β_j be one.

The second part of the Algorithm 2 from line 13 to 25, computes optimal selections by tracking through the values of $\Phi_{a:b}(k)$. Initially a and b are set to represent the entire chain. The algorithm starts with the full budget B and the empty selection set $\mathcal{A}$. In the following loop, the optimal selection y from $\Phi_{a:b}(k)$ is returned and added into the selection set $\mathcal{A}$. Whenever a selection is made, a budget of β_y is spent, and the chain is cut into two parts, $a:y$ and $y:b$. The searching of optimal observation is continued on the second sub-chain. This process stops when the budget is used up. Line 20 in Algorithm 2 is important because it makes the tracking to fall into a correct domain in the whole table. But it was missed out in the original VoIDP algorithm.

The main change in the improved version is reflected in the second part. After $L_{a:b}(k)$ and $\Lambda_{a:b}(k)$ are all computed, we need to find out the optimal selection from the series of $\Lambda_{a:b}(k)$ tables. Initially, it will start from the entire chain with the full budget B and an empty selection set $\mathcal{A}$. The first selection will thus be $\Lambda_{0:n+1}(B)$. If it is set to j, then the next selection should be from $\Lambda_{j:b}(k-\beta_j)$ instead of $\Lambda_{0:b}(k-\beta_j)$ as in the original VoIDP (see figure 1). This slight change, however, leads to a dramatically improved outputs (see table 2.1). As discussed in the last paragraph, the way of tracing back the optimal selections is actually determined by how they were calculated. With this crucial change in the process of recovering optimal selections, the improved version of VoIDP produces desirable outputs.

We have verified the effectiveness of those optimal selections through experiments. The results will be presented in the next section.

2.4 EXPERIMENTS

In this section, we first compare the selection outputs of the original VoIDP algorithm with those of our improved version. Then, we evaluate the optimal selections by the improved VoIDP against the ones generated by a greedy heuristic method and an uniform spacing method. We will describe the two simple methods shortly. We use the temperature time series data set, which was also used in paper [30]. The data set was collected from a network of wireless sensors deployed in the Intel Berkeley Research Lab [8].

2.4.1 OPTIMAL OBSERVATION SELECTION IN WIRELESS SENSOR SCHEDULING

One of the research problems in wireless sensor networks is that how a sensor should be scheduled for sensing in order to both save its power and, in the meantime, obtain the most informative observation possible. In the example, wireless sensors were deployed to monitor indoor temperature and the sensing frequency was only once an hour. The goal was to select k out of 24 time points for scheduling a sensor to turn on so that its expected observations would be the most informative.

The temperature time series data were pre-processed to compensate for missing data, and

each temperature value was discretized into 10 bins of 2 Kelvins. We got 45 sample time series combined from the data collected by three adjacent sensors (#3, #4, and #6) lasting 19 days. We used them to train a HMM that also had four latent states representing from 12 am - 7 am, 7 am - 12 pm, 12 pm - 7 pm and 7 pm - 12 am. All the input rewards used both in the original VoIDP and the improved version were computed from this trained chain graphical model under the filtering case, with assumed unit cost and zero reward penalty for making any observations.

Table 2.1: Optimal observation selections by the original VoIDP [30] and the improved VoIDP 2 (in this example we let unit cost and zero penalty when selecting any observations).

Budget	Optimal Observation Outputs (time point ranges from 1 to 24)			
	Original VoIDP		Improved VoIDP	
	outputs	reward	reward	outputs
1	6	-32.2979	-32.2979	6
2	$5,6$	-31.2704	-29.1210	$5,14$
3	$4,5,6$	-30.3201	-26.8453	$4,10,17$
4	$3,4,5,6$	-29.4919	-25.1229	$3,7,12,18$
5	$1,3,4,5,6$	-28.5762	-23.7130	$1,5,9,14,19$
6	$1,1,3,4,5,6$	-28.5762	-22.5522	$1,5,8,12,16,20$
7	$1,1,1,3,4,5,6$	-28.5762	-21.6115	$1,5,8,11,14,17,21$
8	$1,1,1,1,3,4,5,6$	-28.5762	-20.8333	$1,4,6,9,12,15,18,21$

Table 2.1 shows the comparison of the outputs supposed to be the optimal selections of

observational time points from both algorithms and their relevant rewards. The higher reward and the better quality of selection of the improved algorithms are quite evident. As can be seen from the table, the original VoIDP algorithm repeatedly selects the first time point after budget 5, which is apparently a waste of the budget. In contrast, the improved version (see Algorithm 2) shows optimal selection results. We will further evaluate the solutions given by the improved VoIDP algorithm.

2.4.2 PERFORMANCE COMPARISON

Since the original VoIDP algorithm does not produce satisfying outputs, we will not evaluate it in the following experiments. To examine the performance of the improved VoIDP algorithm, we also use a greedy heuristic and an uniform spacing method for comparison in our experiments.

The selection of observations in the greedy method is accumulated recursively. Assuming unit selection cost, when $k = 1$, the only best observational time point is selected from the entire chain. When $k = 2$, the selection in $k = 1$ case is adopted as the first selection. It divides the whole chain into two sub-chains, and two optimal observations can be computed out of each of them. Then the second selection is the best one among the two observations that has a bigger expected total reward. This process can be deduced into $k = m$ case. When $k = m$, first $m-1$ observations are generated from $k = m-1$ step. The mth selection is picked

among the m optimal observations for the m sub-chains formed by dividing the whole chain using the first $m-1$ selections of observations. The uniform spacing selection is to simply distribute k selections of observations evenly across the entire chain with equal distance.

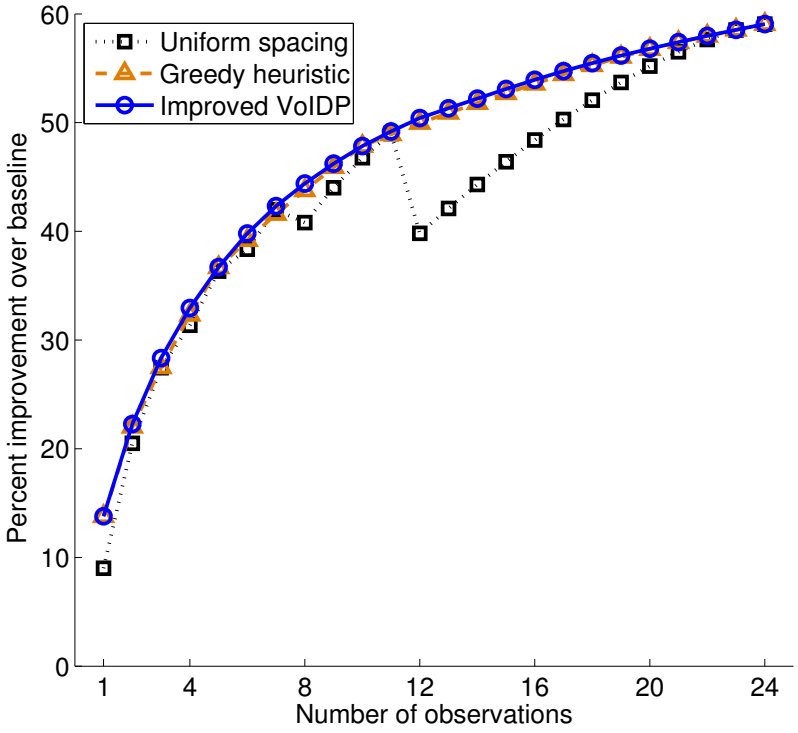


Figure 2.1: Baseline performance comparison: The relative improvement of the uniform spacing method, the greedy heuristic, and the improved VoIDP algorithm over the baseline reward which is the expected total reward for the entire chain without any observations.

In figure 2.1, all the performance results are compared against the baseline, which is the expected total reward for the entire chain without any observation. The performance

is measured as an increase of the expected total reward, which is equivalent to decrease of expected entropy for the entire chain. It shows that the optimal selections given by the improved VoIDP algorithm outperform those by both the heuristics. To give a better picture of how much the improved VoIDP algorithm outperforms the greedy heuristic, we then compare their relative improvements against the uniform spacing. The result is illustrated by figure 2.2. Here, performance is measured as an increase of expected total reward, with the uniform spacing as the baseline.

As shown in figure 2.2, the difference of performance improvement between the optimal selections given by the improved VoIDP algorithm and those by greedy heuristic is obvious, when fewer number of observations are selected. It can be seen that if k is less than one third of all possible observations, the optimal gain by the improved VoIDP algorithm is more than one percent over that by the greedy heuristic. And the gain remains even when k reaches about two thirds of all possible observations. After that, the optimal subset and the subset selected by the greedy heuristic are almost identical. These results empirically verify the effectiveness of the optimal selections produced by the improved VoIDP algorithm presented in Algorithm 2.

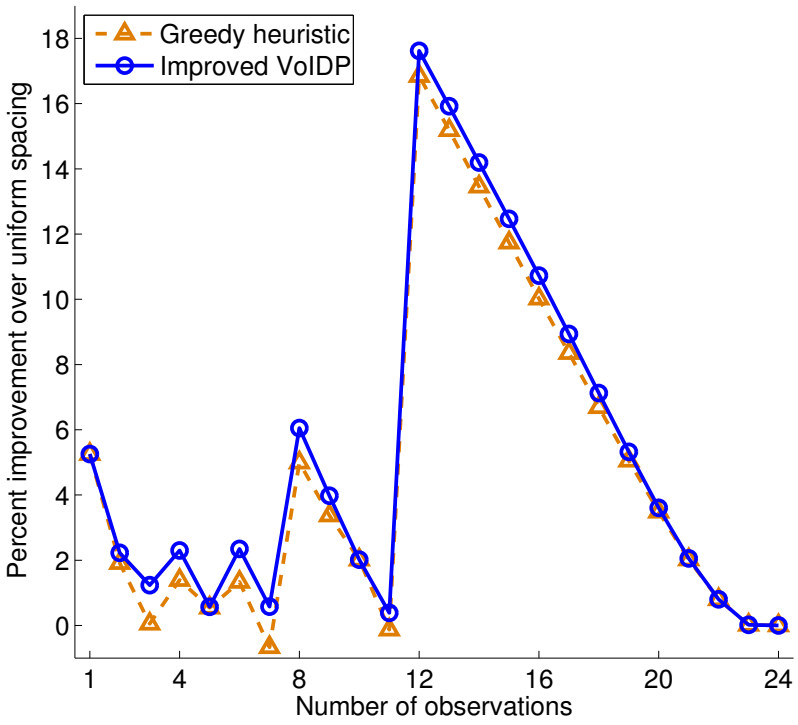


Figure 2.2: Relative performance comparison between the greedy heuristic and the Algorithm 2 over the performance of the uniform spacing method.

2.5 DISCUSSION

The VoIDP algorithm was claimed in [30] as the first optimal algorithm for nonmyopically computing and optimizing value of information in chain graphical models. This algorithm appears at least in [29, 30, 25] and remains in the same form. We evaluated the algorithm for subset selection problem as appearing in [30] and found the issue illustrated by table 2.1. We think it necessary to improve on the algorithm by giving its corrected version.

The computation of $L_{a:b}(k)$ (see notations in 2.3.1) in the improved VoIDP algorithm can be further simplified. If there was no penalty towards the total reward of making any observations, which means $C_j = 0$, then $sel(j) := R_j(\mathcal{X}_j \mid \mathcal{X}_j) + L_{a:j}(0) + L_{j:b}(k - \beta_j)$. The computation of $L_{a:b}(k)$ would become the following, because in this case any $sel(j)$ would be bigger than $L_{a:b}(0)$.

$$L_{a:b}(k) = \max_{j:a<j<b, \beta_j \leq k} \{R_j(\mathcal{X}_j \mid \mathcal{X}_j) + L_{a:j}(0) + L_{j:b}(k - \beta_j)\} \tag{2.2}$$

In other words, if there were no reward penalties, then the expected total reward for sub chain $a : b$ after making any additional observations would always be larger than that of making no additional observations. In this situation, the algorithm (see figure 2) does not need to compute $sel(0)$ in the inner loop, and hence $L_{a:b}(k)$ does not need to compare with $L_{a:b}(0)$.

2.6 RELATED WORK

The original VoIDP algorithm as an efficient tool for selecting observation to maximize the value of information in chain graphical models was first introduced in [29], and was further presented in [30]. Although the optimization problem can be effectively solved for chain graphical models, it is much harder for more general graphical models. The authors proved that the problem of subset selection for even discrete polytree is computational intractable. There are some other approaches suggested for selecting observations in graphical models, but the authors of [30] argued that either some of them, such as the greedy methods, do not have theoretical performance guarantees, or others are running in exponential worst-case time, although they could be applied to more general graphical models. Besides developing algorithms to schedule a single sensor, the authors in [29, 30] also studied scheduling multiple sensors whose measurements are correlated, in which case the graphical model becomes more general, consisting of multiple chains.

The optimization problem of selectively gathering information with a variety of objectives exists in many tasks of real world applications. When, for example, deploying a wireless sensor network to monitor a spatio-temporal phenomenon, we want to choose locations and time points to deploy and schedule the sensors in order to maximize the information gains, and in the meantime minimize its communication costs. A doctor may want to have a most effective diagnostic plan designed at a minimal cost for a patient. Nowadays, the In-

ternet provides a vast amount of information, but people would like to spend a small amount of time to read the most important news or useful information. Several efficient algorithms have been developed to address such problems.

In spatial monitoring such as in [33, 46], the sensor placement problem can be modeled using Gaussian Processes with a mutual information criterion, which is a submodular function. Submodularity, an intuitive diminishing returns property, can be exploited to develop faster, strongly polynomial time combinatorial algorithms with provable theoretical performance guarantees ([53, 22, 26, 35]). It turns out that many observation selection problems ([38, 27, 31, 43, 14]) can utilize this important structural property to develop efficient and near optimal algorithms incorporating greedy heuristic. However, in a more complex setting where another criterion besides the informativeness needs to be considered such as communication cost, greedy algorithms perform arbitrarily badly [34]. The authors of [34] presented a non-myopic algorithm *pSPIEL* which can near-optimally trade off between information and communication cost. Another non-myopic algorithm *Saturate* [36] was designed to minimize the uncertainty that could be exploited by adversaries. In [54, 42], it is shown that submodular functions are applicable to optimization of informative paths for multiple robots. Submodular functions have inspired researchers not only to develop efficient algorithms but also to study theoretical foundation of solving complex combinatorial problems. The authors in [12] introduced an algorithmic framework for studying combinatorial prob-

lems with multi-agent submodular cost functions and presented an approximate algorithm with theoretic lower bound.

There is another alternative approach to selection problems. Other than choosing according to a model (open-loop) before any observations are obtained, sequential planning (closed-loop) decides on the next selection based on previously observed values. In paper [28], the authors compared a sequential algorithm sequentially optimizing mutual information in Gaussian Processes with the model based selection approach, and quantified the advantage of the sequential strategy. A conditional planning based algorithm for selecting observations in chain graphical models was also presented in [29, 30].

2.7 CONCLUSIONS

We present an improved version of VoIDP algorithm for optimally selecting sensor observations on a chain graphical model for time series observations. A mistake in the original VoIDP algorithm is corrected and verified by experimental results. It is a slight-change but significant improvement to the original VoIDP algorithm, because it is critical for producing the desired optimal selection. We also discuss a case when there are no reward penalties, the expected total reward of making any observations will always be larger than that of making no observations. This is used to simplify the computation of the optimal expected total reward for a sub chain.

CHAPTER 3

SPATIAL OBSERVATION SELECTION WITH ITS APPLICATION TO PLACE ROAD TRAFFIC MONITORING SENSORS

3.1 INTRODUCTION

Wireless sensor networks have been deployed in real world applications ranging from environmental monitoring to ambient intelligence [7, 51, 52, 59]. This technology allow us to have a better understanding of the natural environment, human activities and even their interactions. Nowadays most wireless sensors are powered by batteries. However changing batteries for thousands of sensors with human intervention is infeasible for large scale deployment of this technology. It has been challenging to make wireless embedded sensor

networks scalable and sustainable. Even for tomorrow's sensors that can harvest energy from their surrounding environment, the small energy they collect and store will never be taken for granted [9]. Conversely, every sensor observation paid by precious harvested energy should be as rewarding as possible.

Making optimal selection of sensor observations with a limited power budget to maximize its information gains has become an important problem. On the one hand the number of observations needs to be minimized in order to save energy. On the other hand the values of information obtained from the observations need to be maximized. In this chapter, we present a model based approach to solve the optimization problem, specifically to determine at what locations the sensor observations should be made.

In section 3.3, a multivariate Gaussian model-based approach for selecting sensor observations in the spatial domain is presented for the problem of placing traffic monitoring sensors in a simulated road network. Wireless sensors have been deployed in the real world for road traffic surveillance [18] and improved the traffic light controllers [46, 57, 65]. The observations of traffic flow volumes at different locations in a road network of a city can be modeled as a joint multivariate Gaussian distribution. The observations collected from traffic sensors can be used to predict traffic volume information at locations where no sensors are deployed. The multivariate Gaussian model-based approach is also known as Gaussian Process (GP). The optimal selection of observations at different places can be approximately

solved by greedy heuristics based on entropy and mutual information [3] criteria. In particular, the performance of mutual information heuristic is guaranteed by a theoretical lower bound. The authors in [33] found out that entropy heuristic placed temperature and rain sensors near the border of their sensing field, whereas the mutual information heuristic placed more sensors in the central area. During our study of placing traffic monitoring sensors on a road map, we discover that the entropy heuristic places more sensors around intersections, whereas the mutual information heuristic spreads out traffic sensors across the road map, not only taking care of the intersections but also having more sensors deployed at locations near the sources or destinations of traffic flows. Mutual information is also better than entropy in avoiding repeatedly placing sensors at directly correlated locations on the same road segments.

3.2 Related Work

The problem of making optimal selection exists in other fields too. For example, variable or model selection is a typical problem in statistics. The goal is to select a subset of variables and to eliminate the rest from usually a linear regression model to maximize the predictive accuracy, or to get the "big picture" with the strongest effects of predictors [13, 19]. The selected subset of variables however predict a single variable of interest. In wireless sensor networks, selected observations will be used to predict all unobserved points of interest.

The Gaussian Process (GP) model is a generalization of linear regression based on multivariate Gaussian distributions [50]. It has been applied to select optimal locations for spatial monitoring using wireless sensor networks, e.g. prediction of road traffic volumes and controlling traffic signal lights [46, 57, 65]. The [46] shows that a Gaussian Process model has better predictive accuracy of road traffic volumes than a correlation-coefficient based method.

Mutual information functions were proved to be submodular functions [33]. Submodularity reflecting the intuitive property of diminishing returns can be exploited to develop strongly polynomial time combinatorial algorithms with provable theoretical performance guarantees ([22, 26, 35, 53]). Even though many observation selection problems ([14, 27, 31, 38, 43, 54]) utilized the important structural property to develop efficient and near optimal algorithms with greedy heuristics, when other criteria besides the informativeness needed to optimize such as communication costs, etc., the greedy heuristics then performed arbitrarily badly as shown in [34, 42]. Non-myopic algorithms have been developed in [34, 36, 28] that can leverage near-optimally among multiple criteria besides the value of information obtained from observations. Submodular functions have not only inspired researchers to develop efficient algorithms but also to study the theoretical foundation of solving complex combinatorial problems. The authors in [12] introduced an algorithmic framework of studying the combinatorial problems with multi-agent based submodular cost functions and

presented an approximate algorithm with a theoretic lower bound.

3.3 PROBLEM STATEMENT

The observational values of sensors are not only affected by its sensing time but also by its locations. One of the problems is to determine where to place those sensors, because a limited budget usually not allows deployment at everywhere especially for a large sensing area. For example, when wireless sensors are deployed to monitor vehicular traffic flows in a metropolitan area, it would not be feasible to deploy a wireless sensor on every road segment. Selecting optimal locations for placing sensors thus becomes such an important problem that these sensors can be best utilized to predict road traffic volumes and to guide traffic signal controllers to adapt in order to reduce traffic jams.

A road network is illustrated in Fig. 3.1. It is comprised of nine junctions indicated by the squares having four green dots inside. A sensor can be deployed on each end of a road segment. There are totally 72 potential locations for sensor placement. The problem is to select sensor locations that maximize the informative values of their observations for building an accurate traffic model as possible.

Conditional probability $P(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_{\mathcal{A}})$ is helpful to analyze traffic flow volumes at locations where no sensors are placed given the observations from deployed sensors at locations $\mathcal{A}$. If we use entropy to measure uncertainty then the problem becomes to select a subset of

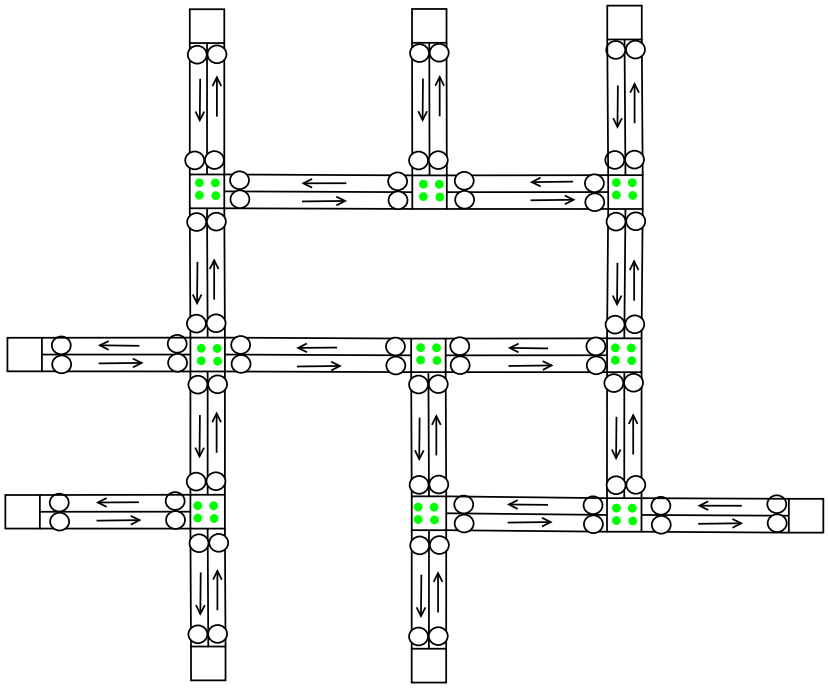


Figure 3.1: A traffic road map with potential sensor deploying locations

locations $\mathcal{A}$ out of the full set $\mathcal{V}$ that minimizes the conditional entropy:

$$H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_{\mathcal{A}}) = -\iint \mathrm{p}(x_{\mathcal{V}\setminus\mathcal{A}}, x_{\mathcal{A}}) \log_2 \mathrm{p}(x_{\mathcal{V}\setminus\mathcal{A}} \mid x_{\mathcal{A}}) \mathrm{d}x_{\mathcal{V}\setminus\mathcal{A}} \mathrm{d}x_{\mathcal{A}} \tag{3.1}$$

where $\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}}$ denotes the random variable of observational values at locations $\mathcal{V}$ excluding $\mathcal{A}$, and $H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_{\mathcal{A}})$ is the entropy of conditional joint probability distribution of the random variables at unobserved locations $\mathcal{V}\setminus\mathcal{A}$ given the observations at locations $\mathcal{A}$.

Because of the chain rule, $H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_{\mathcal{A}}) = H(\mathcal{X}_{\mathcal{V}}) - H(\mathcal{X}_{\mathcal{A}})$, minimizing $H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_{\mathcal{A}})$ is equivalent to maximizing $H(\mathcal{X}_{\mathcal{A}})$. Hence, the problem can also be formulated as selecting the observations at location set $\mathcal{A}$ such that:

$$\mathcal{A}^* = \mathrm{argmax}_{\mathcal{A}\subset\mathcal{V}} H(\mathcal{X}_{\mathcal{A}}) \tag{3.2}$$

which means finding a subset of locations where its sensing observations are the most uncertain.

3.4 Multivariate Gaussian Model for Sensor Placement

The task is to deploy wireless sensors on the road map in Fig. 3.1 to monitor traffic volumes. Each sensor will record the number of cars passing it within a time interval. Con-

sidering observations of each sensor as a random variable, if we assume that observations collected from all of the locations have a joint multivariate Gaussian distribution, then any finite number of the collection of these random variables also have a joint Gaussian distribution. This modeling is also known as a *Gaussian Process* (GP) [50]. The joint Gaussian probability distribution is:

$$P(\mathcal{X}_{\mathcal{V}} = \mathrm{x}_{\mathcal{V}}) = \frac{1}{(2\pi)^{n/2}|\Sigma_{\mathcal{V}\mathcal{V}}|^{1/2}} e^{-\frac{1}{2}(\mathrm{x}_{\mathcal{V}}-\mu_{\mathcal{V}})^T \Sigma^{-1}_{\mathcal{V}'\mathcal{V}'}(\mathrm{x}_{\mathcal{V}}-\mu_{\mathcal{V}})} \tag{3.3}$$

where $\mathcal{V}$ denotes the whole set of sensor locations with $|\mathcal{V}| = n$, $\mu_{\mathcal{V}}$ is the mean vector, and $\Sigma_{\mathcal{V}\mathcal{V}}$ is the covariance matrix. If we take a subset $\mathcal{A}$ from $\mathcal{V}$, then it also satisfies $\mathcal{X}_{\mathcal{A}} \sim \mathcal{N}(\mu_{\mathcal{A}}, \Sigma_{AA})$ where $\mu_{\mathcal{A}}$ is a sub vector of $\mu_{\mathcal{V}}$, and $\Sigma_{\mathcal{A}\mathcal{A}}$ is a corresponding submatrix of $\Sigma_{\mathcal{V}\mathcal{V}}$. This consistency property is also known as the marginalization property. The nice property also applies to the conditional probability $P(\mathcal{X}_{\mathcal{U}} \mid \mathcal{X}_{\mathcal{A}} = \mathrm{x}_{\mathcal{A}})$ that is a joint probability distribution of observations at location subset $\mathcal{U}$ conditional on the observations $\mathrm{x}_{\mathcal{A}}$ at location subset $\mathcal{A}$, assuming $\mathcal{U}, \mathcal{A} \subset \mathcal{V}$. Its conditional mean $\mu_{\mathcal{U}|\mathcal{A}}$ and variance $\sigma^2_{\mathcal{U}|\mathcal{A}}$ are given by:

$$\mu_{\mathcal{U}|\mathcal{A}} = \mu_{\mathcal{U}} + \Sigma_{\mathcal{U}\mathcal{A}}\Sigma^{-1}_{\mathcal{A}\mathcal{A}}(\mathrm{x}_{\mathcal{A}} - \mu_{\mathcal{A}}) \tag{3.4}$$

$$\sigma^2_{\mathcal{U}|\mathcal{A}} = \Sigma_{\mathcal{U}\mathcal{U}} - \Sigma_{\mathcal{U}\mathcal{A}}\Sigma^{-1}_{\mathcal{A}\mathcal{A}}\Sigma_{\mathcal{A}\mathcal{U}} \tag{3.5}$$

where $\mu_{\mathcal{A}}$ is the mean vector of random variable set $\mathcal{X}_{\mathcal{A}}$; $\Sigma_{\mathcal{U}\mathcal{U}}$, $\Sigma_{\mathcal{A}\mathcal{A}}$, $\Sigma_{\mathcal{U}\mathcal{A}}$ and $\Sigma_{\mathcal{A}\mathcal{U}}$ are the corresponding sub-matrices of $\Sigma_{\mathcal{V}\mathcal{V}}$. For example, $\Sigma_{\mathcal{U}\mathcal{A}}$ is formed by the $\mathcal{U}$ rows and the $\mathcal{A}$ columns in $\Sigma_{\mathcal{V}\mathcal{V}}$.

3.5 GREEDY HEURISTICS

3.5.1 MAXIMIZING ENTROPY CRITERION

For solving the problem formulated in (3.2), a greedy method is to start with the selection set as $\mathcal{A}_0 = \emptyset$, then iteratively adding the next location y^*_{i+1} into $\mathcal{A}_i$ that includes all the indices of selected locations in the ith iteration. $y^*_{i+1} \in \mathcal{V} \backslash \mathcal{A}_i$ has the highest conditional entropy:

$$y^*_{i+1} = \text{argmax}_{y_{i+1}} H(\mathcal{X}_{y_{i+1}} \mid \mathcal{X}_{\mathcal{A}_i}), \tag{3.6}$$

Based on the multivariate Gaussian model and the entropy definition in (3.1), we can calculate the entropy of a conditional probability distribution $P(\mathcal{X}_y \mid \mathcal{X}_{\mathcal{A}})$ as:

$$\begin{aligned} H(\mathcal{X}_y \mid \mathcal{X}_A) &= \log \sigma_{\mathcal{X}_y \mid \mathcal{X}_{\mathcal{A}}} + \frac{\log \pi}{2} + \frac{\log 2}{2} + \frac{1}{2} \\ &= \frac{1}{2} \log \sigma^2_{\mathcal{X}_y \mid \mathcal{X}_{\mathcal{A}}} + \frac{1}{2}(\log \pi + \log 2 + 1) \end{aligned} \tag{3.7}$$

where $\sigma^2_{X_y|X_{\mathcal{A}}}$ can be computed using (4.4). Because the log function is monotonic, the value of $\sigma^2_{X_y|X_{\mathcal{A}}}$ is proportional to that of $H(X_y \mid X_A)$.

Algorithm 3: Greedy algorithm of maximizing entropy $H(\mathcal{A})$

Input: covariance matrix $\Sigma_{\mathcal{V}\mathcal{V}}$, k
Output: selection set $\mathcal{A}(\mathcal{A} \subseteq \mathcal{V}, \text{and } |\mathcal{A}| = k)$

```
1 begin
2     A ← ∅
3     for i = 1 to k do
4         foreach y ∈ V\A do δ_y ← σ²_{X_y|X_A}
5         y* ← argmax_{y∈V\A} δ_y
6         A ← A ∪ {y*}
7     end
8 end
```

The algorithm of maximizing the entropy criterion is given in Algorithm 3. Notice that the value of $H(X_y \mid X_{\mathcal{A}_i})$ decreases as the size of $\mathcal{A}_i$ increases. In other words, if there are more observations collected at different locations, then it will make more certain of the predictive value for a unobserved location. Considering the sequence of decreasing $H(X_y \mid X_{\mathcal{A}_i})$ values (when fixed y) as $\mathcal{A}_i$ gets bigger, we actually do not have to calculate δ_y for all $y \in \mathcal{V}\backslash\mathcal{A}$ every time at the line 4 in Algorithm 3. Because that values of δ_y computed in current selection iteration will be no bigger than theirs in the previous iteration when $\mathcal{A}$ is in a smaller size.

Based on this observation and the idea of lazy evaluation in [33], an improved version of greedy algorithm for maximizing the entropy criterion is presented in Algorithm 4. δ_{y^*} is supposed to store the value of $H(X_{y^*} \mid X_{\mathcal{A}})$, but we actuallly only need to compute $\sigma^2_{X_{y^*}|X_{\mathcal{A}}}$

Algorithm 4: Greedy algorithm of maximizing entropy $H(\mathcal{A})$ using lazy evaluation

Input: covariance matrix $\Sigma_{\mathcal{V}\mathcal{V}}$, k
Output: selection set $\mathcal{A}(\mathcal{A} \subseteq \mathcal{V})$

begin
  $\mathcal{A} \leftarrow \emptyset$
  foreach $y \in \mathcal{V}$ **do** $\delta_y \leftarrow +\infty$; $\Phi_y \leftarrow 0$
  for $i = 1$ **to** k **do**
    repeat
      $y^* \leftarrow \text{argmax}_{y \in \mathcal{V} \setminus \mathcal{A}} \delta_y$
      if $\Phi_{y^*} == i$ **then**
        break
      else
        $\delta_{y^*} \leftarrow \sigma^2_{X_{y^*}|X_{\mathcal{A}}}$
        $\Phi_{y^*} \leftarrow i$
      end
    until 0
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{y^*\}$
  end
end

because $\sigma^2_{X_{y^*}|X_{\mathcal{A}}} \propto H(X_{y^*} \mid X_{\mathcal{A}})$. Φ_{y^*} records in which iteration of the *for* loop that δ_{y^*} was updated. If δ_{y^*} is the maximal and is updated in the current iteration, then y^* will be selected into the observation set $\mathcal{A}$. This saves computation of other δ_y's, for the rest of $y \in \mathcal{V} \backslash \mathcal{A}$. We will show how much computation can be saved in the experimental section.

3.5.2 Maximizing Mutual Information Criterion

Besides the entropy based greedy heuristic, another heuristic is to find the observation locations $\mathcal{A}^*$ that maximizes the *entropy reduction*:

$$\mathcal{A}^* = \text{argmax}_{\mathcal{A} \subset \mathcal{V}} (H(X_{\mathcal{V} \backslash \mathcal{A}}) - H(X_{\mathcal{V} \backslash \mathcal{A}} \mid X_{\mathcal{A}})) \tag{3.8}$$

This entropy reduction is also known as the Mutual Information (MI). The mutual information of selection set $\mathcal{A}$ is denoted as MI($\mathcal{A}$), and is given by the following formula:

$$\begin{aligned} MI(\mathcal{A}) &= \text{I}(X_{\mathcal{V} \backslash \mathcal{A}}; X_{\mathcal{A}}) \\ &= H(X_{\mathcal{V} \backslash \mathcal{A}}) - H(X_{\mathcal{V} \backslash \mathcal{A}} \mid X_{\mathcal{A}}) \\ &= H(X_{\mathcal{A}}) - H(X_{\mathcal{A}} \mid X_{\mathcal{V} \backslash \mathcal{A}}) \end{aligned} \tag{3.9}$$

The mutual information criterion was originally proposed by Caselton and Zidek in [3]. Krause et al.in [33] used this criterion to place temperature and rain fall sensors. They found

out that the mutual information led to a more intuitively central placement in a sensing space than the entropy criterion did. The latter instead placed sensors mostly at boundaries.

Solving the problem in (4.8) turns out to be NP-hard. However, Krause et al. [33] proved that the set function of the mutual information MI($\mathcal{A}$) is a submodular function. This discovery eventually led to a development of an efficient approximate algorithm. The concept of *submodularity* was originally introduced by Nemhauser et al. [44]. A set function F is *submodular*, if for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ and $i \in \mathcal{V} \backslash \mathcal{B}$ it holds that $F(\mathcal{A} \cup i) - F(\mathcal{A}) \geq F(\mathcal{B} \cup i) - F(\mathcal{B})$. This demonstrates an inherent property of a submodular function, *diminishing returns*. That is, adding another observation to a smaller set of observations helps more than adding it to a larger set.

The problem in (4.8) is to find the observation set $\mathcal{A}$ that maximizes the submodular function, MI($\mathcal{A}$). For a monotone submodular set function F such that $F(\mathcal{A} \cup i) \geq F(\mathcal{A})$ for all $i \in \mathcal{V}$, a greedy algorithm selecting k elements that maximizes $F(\mathcal{A}_G)$ (where $|\mathcal{A}_G| = k$) is guarantee to have an optimal lower bound. This fundamental result was discovered by Nemhauser et al. [44], and it was stated as below:

Theorem 1 (Nemhauser et al. [44]) *Let F be a* monotone submodular set function *over a finite ground set $\mathcal{V}$ with $F(\emptyset) = 0$. Let $\mathcal{A}_G$ be the set of the first k elements chosen by the* greedy algorithm*, and let $OPT = \max_{\mathcal{A} \subset \mathcal{V}, |\mathcal{A}| = k} F(\mathcal{A})$. Then, $F(\mathcal{A}_G) \geq (1 - (\frac{k-1}{k})^k) OPT \geq (1 - 1/e) OPT$.*

The *greedy algorithm* schema is to select the element i^* that maximizes $F(\mathcal{A} \cup i) - F(\mathcal{A})$. If the $F(\mathcal{A})$ is *monotonic*, it guarantees that the greedy algorithm will produce a solution at least $(1 - 1/e)OPT$, where OPT is the optimal solution value.

The mutual information in (3.9), however, is not always monotonically increasing since $MI(\emptyset) = 0$ and $MI(\mathcal{V}) = 0$. It will actually keep increasing, and after the selection set reaches to a certain size it will then become decreasing. However it holds the monotonicity for a partial selection although not for all the sensor locations. Krause et al. [33] proves that the mutual information is ε-approximately monotonic for selection sets of size up to $2k$, and shows that the quality of a selection given by the greedy algorithm has a optimal lower bound:

Theorem 2 (Krause et al. [33]) *The greedy algorithm,* $y^* = \text{argmax}_y MI(\mathcal{A} \cup y) - MI(\mathcal{A})$*, is guaranteed to select a set* $\mathcal{A}$ *of k sensors for which* $MI(\mathcal{A}) \geq (1 - 1/e)(OPT_{MI} - k\varepsilon)$*, where* OPT_{MI} *is the optimal solution value given by the mutual information set function.*

This theorem not only provides a lower bound for the performance of the greedy algorithm compared with the optimal solution, but also implies an upper bound for the optimal solution of maximizing the mutual information.

An efficient version of the greedy algorithm using lazy evaluation was presented by Krause et al. in [33]. It benefits from the fact that the sequence of mutual information gains

denoted as Δ_y, where $\Delta_y = MI(\mathcal{A} \cup y) - MI(\mathcal{A})$, is monotonically decreasing during the course of the greedy algorithm.

The mutual information gain can be deduced further as:

$$\begin{aligned} \Delta_y &= MI(\mathcal{A} \cup y) - MI(\mathcal{A}) \\ &= H(y \mid \mathcal{A}) - H(y \mid \overline{\mathcal{A}}) \qquad (3.10) \\ &= \frac{1}{2}\log_2\left(\frac{\sigma^2_{X_y|X_{\mathcal{A}}}}{\sigma^2_{X_y|X_{\overline{\mathcal{A}}}}}\right) \qquad (3.11) \end{aligned}$$

where,$\overline{\mathcal{A}}$ denotes $\mathcal{V}\backslash(\mathcal{A} \cup y)$. The deduction of $H(y \mid \mathcal{A}) - H(y \mid \overline{\mathcal{A}})$ is obtained after plugging (3.9), and can be further extended by using (3.7) if we assume a multivariate Gaussian model.

The efficient greedy algorithm of maximizing the mutual information is shown in Algorithm 5. We rewrite it as in the context of the multivariate Gaussian model given in section 3.4. This approximate algorithm is to select a sensor location that maximizes the mutual information gain.

The Φ_{y^*} is used to record in which iteration Δ_{y^*} is being updated. The lazy evaluation saves a lot of computation of Δ_y based on the insight that a sequence of mutual information gains given a fixed y will decreased as set $\mathcal{A}$ gets more observations being added. It will select the y^* if the maximal Δ_{y^*} is updated in the current iteration. Otherwise it will update

Algorithm 5: Greedy algorithm of maximizing mutual information gain $MI(\mathcal{A} \cup y) - MI(\mathcal{A})$ using lazy evaluation

Input: covariance matrix $\Sigma_{\mathcal{V}\mathcal{V}}$, k
Output: selection set $\mathcal{A}(\mathcal{A} \subseteq \mathcal{V})$, mutual information gains Δ

begin
  $\mathcal{A} \leftarrow \emptyset$
  foreach $y \in \mathcal{V}$ **do** $\Delta_y \leftarrow +\infty$; $\Phi_y \leftarrow 0$
  for $i = 1$ **to** k **do**
    repeat
      $y^* \leftarrow \text{argmax}_{y \in \mathcal{V} \setminus \mathcal{A}} \Delta_y$
      if $\Phi_{y^*} == i$ **then**
        break
      else
        $\overline{\mathcal{A}} \leftarrow \mathcal{V} - (\mathcal{A} \cup y^*)$
        $\Delta_{y^*} \leftarrow \frac{1}{2}\log_2(\frac{\sigma^2_{X_{y^*}|X_{\mathcal{A}}}}{\sigma^2_{X_{y^*}|X_{\overline{\mathcal{A}}}}})$
        $\Phi_{y^*} \leftarrow i$
      end
    until 0
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{y^*\}$
  end
end

Δ_{y^*} and Φ_{y^*}, and repeat the selection process.

The algorithm's complexity is $O(kn^3)$. It doesn't include the complexity of computing the covariance matrix, which is assumed as an input. When a data set is large in terms of its number of both instances and columns, the complexity of computing the covariance matrix will become very expensive.

3.6 Experiments of Sensor Placement In a Simulated Road Traffic Map

In this section, we run the two greedy heuristics that optimize entropy and mutual information criteria, respectively, for sensor placement on the traffic road map as shown in Fig. 3.1. We compare the resulted placement maps that shed lights on further understanding of these two optimization criteria in selecting sensor spatial observations.

The traffic road map was created in the Green Light District simulator. The GLD [61] is an open source software for simulating road traffic flows and traffic light controlling strategies. We implemented an sensor class in the GLD system for studying wireless sensor networks aided traffic light controls [46, 57]. The road map in Fig. 3.1 consists of 9 intersections with traffic light controllers, and 8 edge nodes that generate vehicles and serve as sources and destinations of corresponding traffic flows. There are totally 72 locations indicated as the unfilled round spots for potential traffic sensor deployment. Each location counts the number

of vehicles passing by within every 50 timestamps. We continually sensed a thousand rows of samples, for each row it's comprised of vehicle counts from all the 72 locations. We took first 700 rows of samples as the training data set for building a multivariate Gaussian model, and the rest 300 rows of samples as the test data set for testing performances of the selection algorithms.

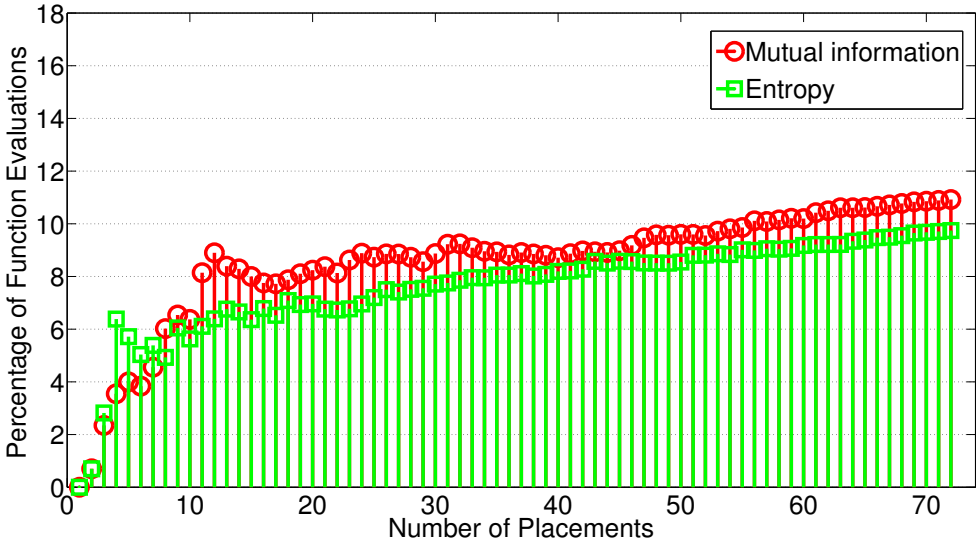


Figure 3.2: Efficiency comparison by the two greedy heuristics both using lazy evaluation

First we will look at how efficient the lazy evaluation strategy is. Algorithm 3 for maximizing entropy $H(\mathcal{A})$ computes $\sigma^2_{X_y|X_\mathcal{A}}$ for all $y \in \mathcal{V}\backslash\mathcal{A}$ in every iteration of selecting a sensor location. However the lazy evaluation strategy makes Algorithm 4 to avoid computing all $\sigma^2_{X_y|X_\mathcal{A}}$ in every iteration. Fig. 3.2 shows how much percentage of the function evaluations

of $\sigma^2_{X_y|X_{\mathcal{A}}}$ have been accounted for during each selection iteration in Algorithm 4. It also shows the efficiency of function evaluations given by the mutual information-based heuristic algorithm 5. The so called lazy evaluation is essentially an upper-bound-based pruning that works really effective, keeping both of heuristics under 12% of total functional evaluations for all kinds of placements. It also shows that the mutual information-based heuristic has a little higher cost than the entropy-based heuristic.

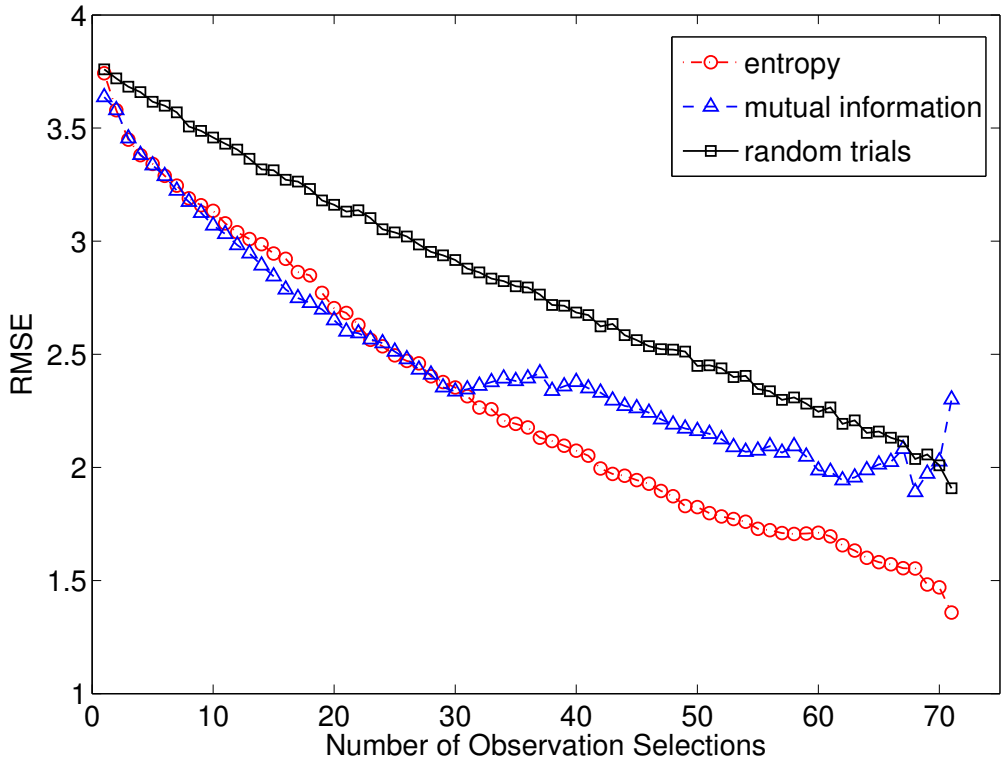


Figure 3.3: Performance comparison

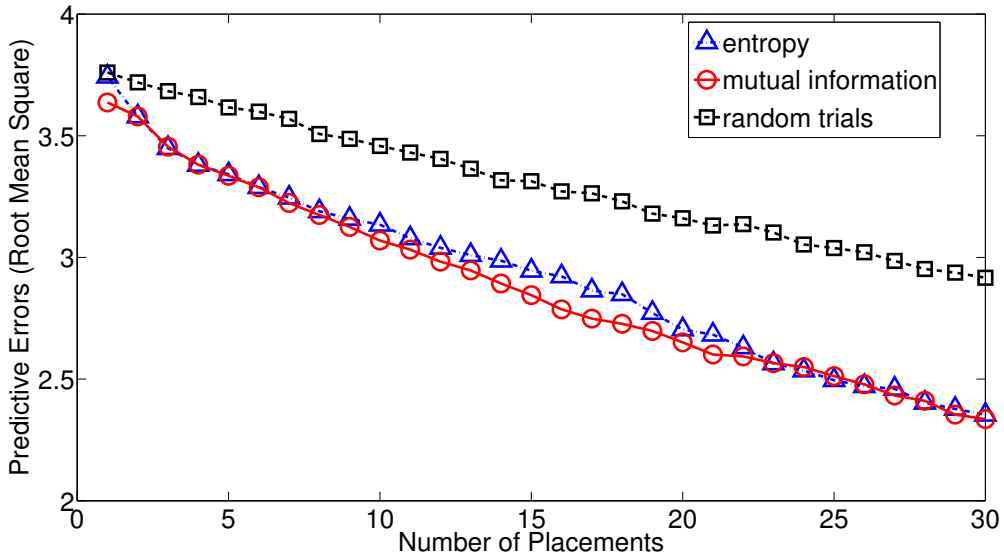
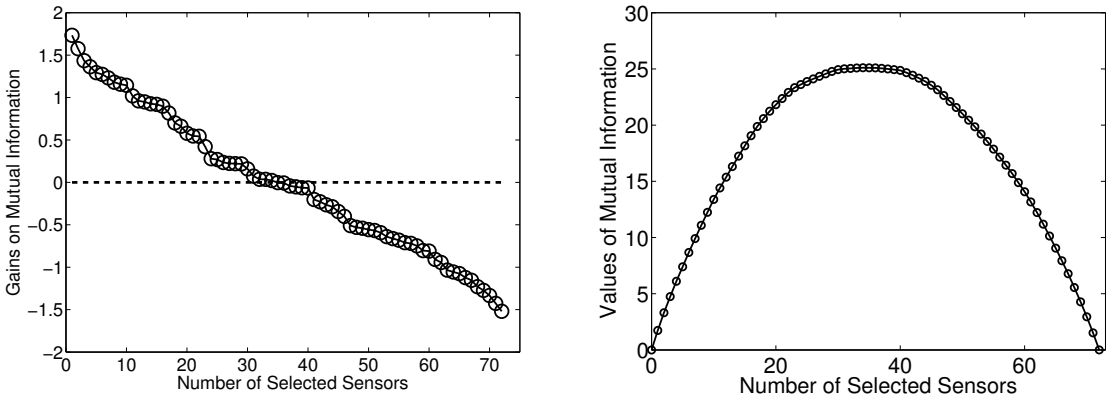


Figure 3.4: Performance comparison (with adjusted range on x-axis)



(a) Mutual information gains

(b) Mutual information values

Figure 3.5: Mutual information on traffic sensor selections

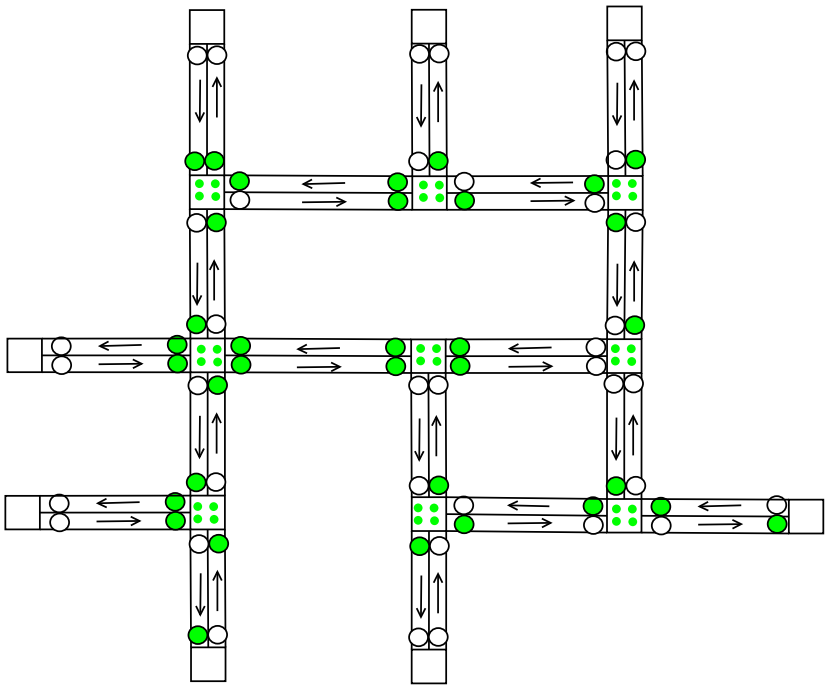
We applied the greedy algorithms of optimizing the entropy and the mutual information criteria and a random selection method for placing sensors on the traffic road map. Fig. 3.3 compares the results in predicting for traffic volumes of all unobserved locations on the map using a test data set. Prediction errors are measured in terms of the Root Mean Square Error. Every plotting point by the random selection method was an average of 100 random trials. The traffic prediction model based on the selected sensor observations given by the mutual information optimization criterion performs slightly better than that by the entropy criterion before approximately 30 observation selections (a range-adjusted version is also drawn in Fig. 3.4 to clearly show the improvement). After that, the mutual information criterion-based heuristic is outperformed by the entropy criterion-based heuristic. This is caused by the non-monotonicity of a mutual information function after a certain number of selections, which is demonstrated in Fig. 3.5. It makes sense to disregard the selection results given by the mutual information heuristic after the 30 sensor selections. Fig. 3.4 also shows that random selections perform much worse than both of the greedy heuristics.

Fig. 3.5 shows values of the mutual information ($\mathrm{MI}(\mathcal{A}_i)$) and the mutual information gains ($\mathrm{MI}(\mathcal{A}_i \cup y^*_{i+1}) - \mathrm{MI}(\mathcal{A}_i)$) during the course of sensor selection. The selection process is based on the multivariate Gaussian model that was built using a training data set. We can see in Fig. 3.5a that the gains on mutual information are positive though keeping decreased until the selection size reaches 34. It becomes negative afterwords that means the

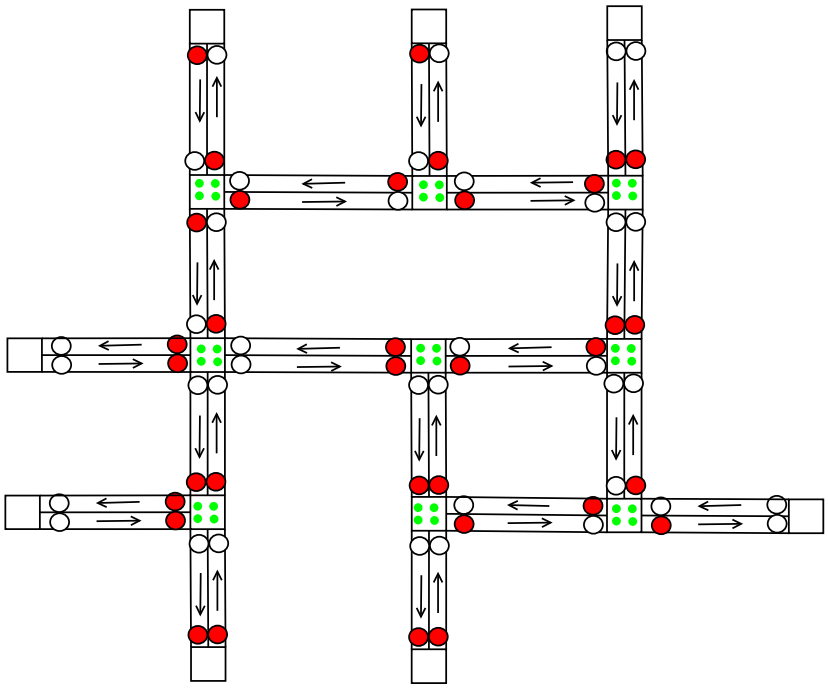
selection model based on the mutual information criterion obtains no gains after selecting 34 sensors. This change point also reflects in Fig. 3.5b, which shows that mutual information values monotonically increase and then decrease after passing the maximum value at selecting 34 sensors. Fig. 3.5 shows the qualities of observation selections based on the mutual information heuristic, which in turn explains why the selection method based on the mutual information criterion starts getting worse in predicting for traffic volumes compared to the entropy criterion-based method after deploying around 30 sensors as shown in Fig. 3.3.

Fig. 3.6 shows two maps of sensor deployment. The sensor locations were determined by the two greedy heuristics based on the mutual information and the entropy, respectively. Because the mutual information criterion works only within the 34 sensor selections in this example (see Fig. 3.5), we therefore set the selection size as 34. The filled round spots are denoted as the locations where sensors are placed. The number of common locations selected by both the greedy heuristics is 19, and the number of different placement locations is accounted for 15, a 44.12% of the whole set of sensor locations.

Fig. 3.6a shows that the sensor locations selected by optimizing the entropy criterion are mostly concentrated around the intersections and very few sensors are deployed near the edges. In contrast, the map in Fig. 3.6b shows that the sensor locations selected by optimizing the mutual information criterion are more dispersed, and there are more sensors deployed near the edge nodes.



(a) Sensor placement by optimizing entropy criterion



(b) Sensor placement by optimizing mutual information criterion

Figure 3.6: Sensor deployment maps

Every road segment can have two sensors respectively placed at each end of it. Intuitively, the sensor readings of the two sensors placed on the same road would be highly correlated. It can be seen from Fig. 3.6b that mutual information criterion-based heuristic never places two sensors on the same road segment in one direction. On the contrary, as shown in Fig. 3.6a, there are three road segments with two sensors placed by the entropy criterion-based heuristic method.

By comparing the two maps, the mutual information criterion-based heuristic picks locations more evenly than the entropy criterion-based heuristic does, and it also avoids repeatedly selecting the observational locations that have strong correlations.

3.7 CONCLUSIONS

In this chapter, for selection of sensor observations in the spatial domain, we apply two greedy heuristics based on the multivariate Gaussian model to select the optimal locations for deploying traffic monitoring sensors. Experimental results show that the entropy criterion-based heuristic places sensors mainly around intersections whereas the mutual information criterion-based heuristic disperses sensors more widely across the road network. Moreover, we discover that the mutual information criterion is better than the entropy criterion in avoiding repeatedly selecting strong correlated locations on the same road segments. However the mutual information criterion-based heuristic performs poorly in placing large number of sen-

sors.

CHAPTER 4

COMPARISON OF MODEL-BASED OPTIMAL OBSERVATION SELECTIONS

4.1 INTRODUCTION

The technology of wireless sensor networks has been popular for more than a decade in both academia and industry. Through the observations obtained by tiny embedded wireless sensors, we can have a better understanding of the natural environment, human activities and their interactions. Researchers have been trying to turn the current relatively small-scale wireless sensor networks to a future generation of large-scale, energy sustainable, and extensively long-standing deployments. One of the biggest hurdles is the constrained power source. Today's wireless sensors are mostly battery powered. It is not a viable way for human intervention to replace the batteries for thousands of wireless sensor nodes. The high

maintenance overhead prevents the wireless sensing technology from prevailing in real world large-scale applications.

A common technique for extending the life time of a wireless sensor network is to reduce its duty cycles, i.e. a sensor wakes up for a small amount of time in a fixed period of interval to sense and go to sleep the rest of the time. Besides reducing sensing times for saving energy, it would be desirable to have the informative values obtained from the sensing data as high as possible. The observations should be worth of their energy costs. This is even more important for the next generation of scalable wireless networked sensors.

Tomorrow's sensors will be much smaller and energy sustainable. They can harvest energy from the environment [9]. The ambient power sources in their surrounding such as heats, mechanical movements, electromagnetic induction, electrochemical reactions and etc., make sensing more sustainable and hassle-free. However, energy-harvesting on wireless sensors also brings a lot of challenges. First of all, the ambient power supply is often intermittent, and the smaller size sensor nodes are, the less power they can store. Moreover, the harvested energy usually needs to be accumulated to reach a certain level before being capable of performing some operations like taking a sensor reading or transmitting a packet. In such situations, the energy harvested sedulously should never be taken for granted. Instead, every sensing observation paid by the harvested energy should be as rewarding as possible.

Making optimal scheduling of observations with a limited power budget to maximize

their information gains has become an important problem in real world applications. It is a trade off between obtaining more and useful information, versus making less observations. The optimization of selecting observations can be considered as a subset selection problem. For example in a task of monitoring indoor temperature, if a wireless sensor is deployed to observe for only once per hour, and we want to turn on the sensor k times for a day, then it becomes such a *subset selection problem* that k out of the total 24 time points are chosen so that the k observations will be the best selection among the other options for having the most accurate predictions of temperature readings at the unobserved time points.

In the context of statistics, the problem of subset selection or variable selection determines a subset of variables and eliminates the rest from usually a linear regression model, in order to increase predictive accuracy or to get the "big picture" with the strongest effects of predictors [13, 19]. But the selection of variables are usually for predicting only a single variable of interest. However, in the sensor scheduling case, the subset selection needs to predict the temperature distribution that covers all the unobserved time points.

Probabilistic inference in graphical models [23] provides an effective tool to deal with the quantification of uncertainty existed in the subset selection problem. The optimal subset of observations is the one that minimizes the uncertainty of the posterior conditional probability distribution of the unobserved variables. Hidden Markov Models (HMMs), the chain graphical models, are often used for modelling time series data. The posterior con-

ditional probability of unobserved variables given observations can be efficiently computed on a HMM. The VoIDP algorithm [30] was claimed to be the first optimal algorithm for efficiently making the subset selection on the chain graphical models. The name "VoIDP" was after its usage of dynamic programming approach to optimize the information value. We corrected the original VoIDP algorithm by fixing a mistake in it [49]. In this chapter, we apply the corrected version of VoIDP algorithm on a HMM to solve the subset selection problem for sensor scheduling.

Gaussian Process (GP) is a generalization of linear regression based on multivariate normal distributions [50]. It is often applied to spatial monitoring problems [32, 46]. Greedy methods based on heuristics, such as entropy and mutual information, can efficiently make a selection of time points. The mutual information [3] criterion, which measures entropy reduction, was shown to have better solutions than the entropy criterion in a couple of sensor placement problems [32]. In this chapter, we use the GP based approach with both the entropy and mutual information heuristics to solve the sensor scheduling problem.

In summary, the objective of the sensor scheduling problem is to select a subset of time points to turn on a sensor and keep it off at the other time to minimize prediction errors at unobserved time points. Our contribution is employing the GP model based selection approach and our corrected version of VoIDP algorithm based on graphical models to solve the scheduling problem and comparing their performances. The GP based approach is more

data driven than the graphical model based approach. But the latter, such as a HMM, can capture the underlying structures of the latent variables that determines the observational values in time series. Through comparison experiments, we find out that the GP based selection approach achieves lower prediction error than the HMM based approach given with accurate observations. However the HMM based selection approach performs more stably and robustly than the GP based approach with erroneous observations.

We will briefly describe the graphical model based, particularly HMM based, selection approach in Section 4.2, and the GP based selection approach in Section 4.3. The experimental results will be presented and discussed in Section 4.4. Following that is the Conclusions.

4.2 Probabilistic Graphical Model Based Observation Selection

A sensor's time series of observations can be modeled using a probabilistic graphical model, such as a HMM. Each observation variable at a time point has a distribution over some hidden states such as different time periods or some events etc. Any observations made on the chain graphical model will contribute to the predictions of values for other unobserved variables. But the degree of the contributions depends on the selection of observations.

The quality of a selection of observations is measured based on how the observation subset changes the shape of the probability distribution of an unobserved variable. The prob-

ability distribution of any unobserved variable conditioned on a subset of observed variables can be efficiently computed using a trained HMM. A utility reward can thus be defined by the entropy on the posterior conditional probability distribution. When a subset of observation variables is selected, an expected total reward across the entire time chain is computed. Hence, the sensor scheduling problem can be formulated as a subset selection problem on a chain graphical model as follows...

Given random variables $\mathcal{X}_\mathcal{V} = (\mathcal{X}_1, \ldots, \mathcal{X}_n)$, a subset of the variables, $\mathcal{X}_\mathcal{A} = (\mathcal{X}_{i_1}, \ldots, \mathcal{X}_{i_k})$ is observed as $\mathrm{x}_\mathcal{A}$. $P(\mathcal{X}_\mathcal{V} \mid \mathcal{X}_\mathcal{A} = \mathrm{x}_\mathcal{A})$ is the conditional observation distribution over all variables given the observation $\mathrm{x}_\mathcal{A}$. A total reward $R(P(\mathcal{X}_\mathcal{V} \mid \mathcal{X}_\mathcal{A} = \mathrm{x}_\mathcal{A}))$ is also given. The subset selection problem is:

$$\mathcal{A}^* = \mathrm{argmax}_\mathcal{A} \sum_{\mathrm{x}_\mathcal{A}} P(\mathcal{X}_\mathcal{A} = \mathrm{x}_\mathcal{A}) R(P(\mathcal{X}_\mathcal{V} \mid \mathcal{X}_\mathcal{A} = \mathrm{x}_\mathcal{A})), \tag{4.1}$$

where the expected total reward is used because the future observations of $\mathcal{X}_\mathcal{A}$ are unknown. The selections are made based on the model before knowing any observational values. Because of conditional independency on chain graphical models, the expected total reward to maximize in (4.1) is also the summation of all the expected local rewards, where an expected local reward $R_j(\mathcal{X}_j \mid \mathcal{X}_\mathcal{A})$ equals to $\sum_{\mathrm{x}_\mathcal{A}} P(\mathcal{X}_\mathcal{A} = \mathrm{x}_\mathcal{A}) R_j(\mathcal{X}_j \mid \mathrm{x}_\mathcal{A})$. A local reward $R_j(\mathcal{X}_j \mid \mathrm{x}_\mathcal{A})$ depends on $P(\mathcal{X}_j \mid \mathcal{X}_\mathcal{A} = \mathrm{x}_\mathcal{A})$, which is the marginal distribution of variable $\mathcal{X}_j$ conditioned on observations $\mathcal{X}_\mathcal{A} = \mathrm{x}_\mathcal{A}$. We can define the reward by using conditional entropy, that is,

$R_j(\mathcal{X}_j \mid \mathrm{x}_{\mathcal{A}}) = -H(\mathcal{X}_j \mid \mathrm{x}_{\mathcal{A}}) = \int P(x_j, \mathrm{x}_{\mathcal{A}}) \log_2 P(x_j \mid \mathrm{x}_{\mathcal{A}}) \mathrm{d}x_j$.

The conditional independence property of chain graphical models simplifies the evaluation of $P(\mathcal{X}_j \mid \mathcal{X}_{\mathcal{A}})$. If only the observations made before a time point are used, then the evaluation of $P(\mathcal{X}_j \mid \mathcal{X}_{\mathcal{A}})$ only depends on $P(\mathcal{X}_j \mid \mathcal{X}_i)$, where $\mathcal{X}_i$ is the closest observation made before j. Moreover, the conditional independence property also decomposes the expected total reward for the entire chain into the local rewards for sub-chains that are separated by the observations. This important property inspired an efficient approximate algorithm based on the divide-and-conquer strategy [30].

The subset selection problem in (4.1) is a combinatorial optimization problem like the *Knapsack* problem although its utility function is more computationally complicated. It is a NP-hard problem. The *Knapsack* problem admits a pseudo-polynomial time algorithm, and all known such algorithms for NP-hard problems are based on dynamic programming [58].

The authors in [29, 30] developed an algorithm called VoIDP to solve the subset selection problem, and it was claimed to be the first optimal algorithm for selecting the optimal observations on chain graphical models. It implements a dynamic programming approach that exploits the chain structured models to efficiently evaluate the expected total reward. The time complexity of the algorithm was proved to be $(\frac{1}{6}n^3 + O(n^2))B$ given budget B in terms of the number of evaluations of the rewards, where $n = |\mathcal{V}|$. Although it works efficently and optimally with a chain graphical model, it is still much harder to be applied on more

general forms of graphical models. It was proved that the problem of selecting a subset of observations for even discrete poly-tree is computational intractable.

We found out a mistake in the originally published VoIDP algorithm that fails to give correct solution. We corrected the mistake and presented an improved version of the algorithm in [49]. In the experiments, we will use the corrected VoIDP algorithm that will be briefly presented in the following. Once the optimal observations are selected, the predictive value will be calculated by taking the expected mean of a posterior conditional observation distribution.

Algorithm 6 shows the corrected version of VoIDP. The first part of the algorithm dynamically computes a three-dimension (a, b and k) table, where a and b are the two ends of a chain, and k is the budget. Each cell in the table is a tuple of two elements: $L_{a:b}(k)$ and $\Phi_{a:b}(k)$. $L_{a:b}(k)$ represents the optimal expected total reward of the sub-chain $a:b$ with the budget k. $sel(j)$ is the expected total reward of the sub chain $a:b$ obtained by observing at time point j, and $sel(0)$ is the reward when no observation is made. $\Phi_{a:b}(k)$ stores the j that maximizes the value of $sel(j)$. $L_{a:b}(0)$ is the base case, and the recursion for $L_{a:b}(k)$ is either $L_{a:b}(0)$ or $\max_{a<j<b, \beta_j \leq k} \{sel(j)\}$. It means that we can choose not to spend any more of the budget to reach the base case, or select the optimal observation at j with the cost β_j. C_j denotes penalty on the reward by selecting at j. In our experiments, we let C_j be zero and β_j be one.

Algorithm 6: Improved VoIDP algorithm for optimizing observation selection on chain graphical models (restated in Chapter 4)

Input: Budget B, rewards R_j, costs β_j and penalties C_j ($j \in \mathcal{V}, |\mathcal{V}| = n$)
Output: Optimal selection of observations at $\mathcal{A}$

begin
  for $0 \leq a < b \leq n+1$ **do** compute $L_{a:b}(0)$
  for $k = 1$ **to** B **do**
    for $0 \leq a < b \leq n+1$ **do**
      $sel(0) \leftarrow L_{a:b}(0)$
      for $j = a+1$ **to** $b-1$ **do**
        $sel(j) \leftarrow R_j(\mathcal{X}_j|\mathcal{X}_j) - C_j + L_{a:j}(0) + L_{j:b}(k-\beta_j)$
      end
      $L_{a:b}(k) \leftarrow \max_{j\in\{0,a+1,\ldots,b-1\}} sel(j)$
      $\Phi_{a:b}(k) \leftarrow \operatorname{argmax}_{j\in\{0,a+1,\ldots,b-1\}} sel(j)$
    end
  end
  $a \leftarrow 0; b \leftarrow n+1$
  $k \leftarrow B$
  $\mathcal{A} \leftarrow \emptyset$
  repeat
    $y \leftarrow \Phi_{a:b}(k)$
    if $y > 0$ **then**
      $\mathcal{A} \leftarrow \mathcal{A} \cup \{y\}$
      $\mathbf{a} \leftarrow \mathbf{y}$
      $k \leftarrow k - \beta_y$
    else
      break
    end
  until $k \leq 0$
end

The second part of Algorithm 6 computes the optimal selections by tracking through the values of $\Phi_{a:b}(k)$. Initially a and b are set to represent the entire chain. It starts with the full budget B and the empty selection set $\mathcal{A}$. In the following loop, it finds the optimal selection y from $\Phi_{a:b}(k)$, and adds it into the selection set $\mathcal{A}$. Whenever a selection is made, it spent a budget of β_y and cuts the chain into two parts, $a:y$ and $y:b$. The searching of optimal observation will be continued on the second sub-chain $y:b$. This process will stop when the budget is used up. The bold faced $a \leftarrow y$ is important because it makes the tracking to fall into the correct domain in the whole table, but it was not in the original VoIDP algorithm.

4.3 GAUSSIAN PROCESS BASED OBSERVATION SELECTION

A Gaussian process (GP), by definition [50], is a collection of random variables, and any finite number of variables in the collection also have a joint Gaussian distribution. For scheduling a sensor, we assume that the joint probability of its observations at all the time points is a multivariate Gaussian distribution:

$$P(\mathcal{X}_{\mathcal{V}} = \mathrm{x}_{\mathcal{V}}) = \frac{1}{(2\pi)^{n/2}|\Sigma_{\mathcal{V}\mathcal{V}}|^{1/2}} e^{-\frac{1}{2}(\mathrm{x}_{\mathcal{V}}-\mu_{\mathcal{V}})^T \Sigma^{-1}_{\mathcal{V}\mathcal{V}}(\mathrm{x}_{\mathcal{V}}-\mu_{\mathcal{V}})} \tag{4.2}$$

where $\mathcal{V}$ denotes the whole set of the variable indices with $|\mathcal{V}| = n$. $\mu_{\mathcal{V}}$ is the mean vector and $\Sigma_{\mathcal{V}\mathcal{V}}$ is the covariance matrix. A subset $\mathcal{A}$ from $\mathcal{V}$ also satisfies that $\mathcal{X}_{\mathcal{A}} \sim \mathcal{N}(\mu_{\mathcal{A}}, \Sigma_{\mathcal{A}\mathcal{A}})$,

where $\mu_{\mathcal{A}}$ is a sub vector of $\mu_{\mathcal{V}}$ and $\Sigma_{\mathcal{A}\mathcal{A}}$ is the relevant sub-matrix of $\Sigma_{\mathcal{V}\mathcal{V}}$. This consistency property is also known as the marginalization property.

The nice property also holds in $P(\mathcal{X}_i \mid \mathcal{X}_{\mathcal{A}} = \mathrm{x}_{\mathcal{A}})$, the probability distribution of $\mathcal{X}_i$ conditioned on the observational values $\mathrm{x}_{\mathcal{A}}$ of a selected variable set $\mathcal{X}_{\mathcal{A}}$. This posterior conditional probability is also a Gaussian distribution with conditional mean $\mu_{i|\mathcal{A}}$ and variance $\sigma^2_{i|\mathcal{A}}$ given by:

$$\mu_{i|\mathcal{A}} = \mu_i + \Sigma_{i\mathcal{A}}\Sigma^{-1}_{\mathcal{A}\mathcal{A}}(\mathrm{x}_{\mathcal{A}} - \mu_{\mathcal{A}}), \tag{4.3}$$

$$\sigma^2_{i|\mathcal{A}} = \Sigma_{ii} - \Sigma_{i\mathcal{A}}\Sigma^{-1}_{\mathcal{A}\mathcal{A}}\Sigma_{\mathcal{A}i}, \tag{4.4}$$

where $\mu_{\mathcal{A}}$ is the mean vector of variable set $\mathcal{X}_{\mathcal{A}}$, $\Sigma_{i\mathcal{A}}$ and $\Sigma_{\mathcal{A}i}$ are the corresponding sub-matrices of $\Sigma_{\mathcal{V}\mathcal{V}}$. For instance, the $\Sigma_{i\mathcal{A}}$ is calculated by taking the ith row and the $\mathcal{A}$ columns of $\Sigma_{\mathcal{V}\mathcal{V}}$. μ_i and Σ_{ii} can be considered as the prior mean and the prior variance of the observations at i.

4.3.1 THE ENTROPY HEURISTIC

The conditional probability $P(\mathcal{X}_i \mid \mathcal{X}_\mathcal{A})$ carries very important information for evaluating the quality of the selection set $\mathcal{A}$, which can be measured by the conditional entropy:

$$\begin{aligned} H(\mathcal{X}_i \mid \mathcal{X}_\mathcal{A}) &= -\iint P(x_i, x_\mathcal{A}) \log P(x_i \mid x_\mathcal{A}) \mathrm{d}x_i \mathrm{d}x_\mathcal{A} \\ &= \frac{1}{2} \log \sigma^2_{i|\mathcal{A}} + \frac{1}{2}(\log \pi + \log 2 + 1), \end{aligned} \tag{4.5}$$

where $P(\mathcal{X}_i \mid \mathcal{X}_\mathcal{A})$ is assumed as a Gaussian probability distribution. Note that the entropy is a monotonic function of its variance $\sigma^2_{i|\mathcal{A}}$, which can be evaluated ahead of making any observations (4.4).

The sensor scheduling problem becomes to select a subset of time points at $\mathcal{A}$ (out of the variable index set $\mathcal{V}$) to turn on the sensor and to keep it off at all the other time (as of indices in $\mathcal{V} \backslash \mathcal{A}$). The subset selection can be optimized by minimizing the entropy $H(\mathcal{X}_{\mathcal{V}\backslash\mathcal{A}} \mid \mathcal{X}_\mathcal{A})$. This is also equivalent to find a subset $\mathcal{A}$ that maximizes $H(\mathcal{X}_\mathcal{A})$, as the chain rule for conditional entropy holds that $H(\mathcal{X}_{\mathcal{V}\backslash\mathcal{A}} \mid \mathcal{X}_\mathcal{A}) = H(\mathcal{X}_\mathcal{V}) - H(\mathcal{X}_\mathcal{A})$. The optimization problem turns out to be a NP-hard problem. To solve it, a greedy heuristic is to iteratively find the next selection $y^*_{i+1} \in \mathcal{V} \setminus \mathcal{A}_i$ that has the highest conditional entropy given the current selection set $\mathcal{A}_i$:

$$y^*_{i+1} = \mathrm{argmax}_{y_{i+1}} H(\mathcal{X}_{y_{i+1}} \mid \mathcal{X}_{\mathcal{A}_i}), \tag{4.6}$$

Algorithm 7: Greedy algorithm for maximizing entropy $H(\mathcal{A})$ (restated in Chapter 4)

Input: covariance matrix $\Sigma_{\mathcal{V}\mathcal{V}}$, selection size k
Output: selection set $\mathcal{A}(\mathcal{A} \subseteq \mathcal{V}, \text{and } |\mathcal{A}| = k)$

```
1 begin
2     A ← ∅
3     for i = 1 to k do
4         foreach y ∈ V\A do δ_y ← σ²_{y|A}
5         y* ← argmax_{y∈V\A} δ_y
6         A ← A ∪ {y*}
7     end
8 end
```

Algorithm 7 shows the greedy algorithm based on the entropy heuristic. k is the size of the selection and the $\sigma^2_{y|\mathcal{A}}$ can be computed using (4.4). Because the log function is monotonic, $\sigma^2_{y|\mathcal{A}}$ is proportional to $H(\mathcal{X}_y \mid \mathcal{X}_\mathcal{A})$. That means finding a selection at y that maximizes $H(\mathcal{X}_y \mid \mathcal{X}_\mathcal{A})$ is equivalent to finding such a y that maximizes $\sigma^2_{y|\mathcal{A}}$.

The computation of $\sigma^2_{y|\mathcal{A}}$ is expensive. Let $|\mathcal{V}| = n$, there are n times of these computations when $i = 1$, and $(n-k+1)$ times when $i = k$. Hence, Algorithm 7 has totally $\frac{(2n-k+1)k}{2}$ times of evaluations of $\sigma^2_{y|\mathcal{A}}$.

4.3.2 The Mutual Information Heuristic

Another criterion for optimizing the subset selection is the mutual information, which was originally proposed by Caselton and Zidek in [3]. The mutual information of a subset $\mathcal{A}$

denoted as $\mathrm{MI}(\mathcal{A})$ is defined as the following, and it is actually the entropy reduction.

$$\begin{aligned} \mathrm{MI}(\mathcal{A}) &= \mathrm{I}(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}};\mathcal{X}_{\mathcal{A}}) \\ &= H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}}) - H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_{\mathcal{A}}) \\ &= H(\mathcal{X}_{\mathcal{A}}) - H(\mathcal{X}_{\mathcal{A}} \mid \mathcal{X}_{\mathcal{V}\setminus\mathcal{A}}) \end{aligned} \tag{4.7}$$

The authors in [32] demonstrated the advantage of the mutual information criterion over the entropy criterion in sensor placement for a couple of spatial monitoring applications. They found out that the mutual information criterion leaded to a more intuitive sensor placement than the entropy criterion did. The mutual information criterion placed sensors in the central areas of a sensing space, whereas the entropy placed sensors mostly at the boundaries.

In contrast to the entropy based selection maximizing only the uncertainty of the selection set $\mathcal{A}$, the mutual information criterion maximizes the reduction of the entropy over the rest of the variable space $\mathcal{V}\setminus\mathcal{A}$ before and after observing at $\mathcal{A}$. That is, to schedule a sensor, we will find a subset of time points at $\mathcal{A}^*$ such that:

$$\mathcal{A}^* = \mathrm{argmax}_{\mathcal{A}\subset\mathcal{V}}\mathrm{MI}(\mathcal{A}) \tag{4.8}$$

Optimization of the mutual information is also a NP-complete problem. A greedy algorithm was developed in [32] that selects the next variable y maximizing the mutual informa-

tion gain:

$$\Delta_y = \mathrm{MI}(\mathcal{A} \cup y) - \mathrm{MI}(\mathcal{A}), \tag{4.9}$$

The greedy heuristic chooses the next selection that provides the maximal increase in the values of mutual information.

In the context of Gaussian Process and based on the entropy equation in (4.5), Equation (4.9) can be further deduced as:

$$\begin{aligned} \Delta_y &= \mathrm{MI}(\mathcal{A} \cup y) - \mathrm{MI}(\mathcal{A}) \\ &= H(y \mid \mathcal{A}) - H(y \mid \overline{\mathcal{A}}) \qquad (4.10) \\ &= \frac{1}{2}\log_2\left(\frac{\sigma^2_{y|\mathcal{A}}}{\sigma^2_{y|\overline{\mathcal{A}}}}\right) \qquad (4.11) \end{aligned}$$

where $\overline{\mathcal{A}}$ means all the variable indices in $\mathcal{V}$ excluding $\mathcal{A}$ and y, which can also be denoted as $\mathcal{V} \setminus (\mathcal{A} \cup y)$.

An interesting notice about the mutual information gain Δ_y is that it is monotonically decreasing as the selection set $\mathcal{A}$ gets larger. It inspired the enhanced greedy algorithm with lazy evaluation [32].

The greedy algorithm with lazy evaluation for maximizing the mutual information is presented in Algorithm 8. We rewrite it as in the context of the Gaussian Process model.

Φ_{y^*} records in which iteration Δ_{y^*} is updated. The lazy evaluation saves a lot of compu-

Algorithm 8: Greedy algorithm for maximizing mutual information gain $\mathrm{MI}(\mathcal{A} \cup y) - \mathrm{MI}(\mathcal{A})$ using lazy evaluation (restated in Chapter 4)

Input: covariance matrix $\Sigma_{\mathcal{V}\mathcal{V}}$, selection size k
Output: selection set $\mathcal{A}(\mathcal{A} \subseteq \mathcal{V})$, mutual information gains Δ

begin
  $\mathcal{A} \leftarrow \emptyset$
  foreach $y \in \mathcal{V}$ **do** $\Delta_y \leftarrow +\infty$; $\Phi_y \leftarrow 0$
  for $i = 1$ **to** k **do**
    repeat
      $y^* \leftarrow \mathrm{argmax}_{y \in \mathcal{V} \setminus \mathcal{A}} \Delta_y$
      if $\Phi_{y^*} == i$ **then**
        break
      else
        $\overline{\mathcal{A}} \leftarrow \mathcal{V} - (\mathcal{A} \cup y^*)$
        $\Delta_{y^*} \leftarrow \frac{1}{2}\log_2(\frac{\sigma^2_{y^*|\mathcal{A}}}{\sigma^2_{y^*|\overline{\mathcal{A}}}})$
        $\Phi_{y^*} \leftarrow i$
      end
    until 0
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{y^*\}$
  end
end

tation of Δ_y based on the insight that the sequence of the mutual information gains on a fixed y decreases as set $\mathcal{A}$ gets bigger. It will select the y^* if the maximal Δ_{y^*} is updated in the current iteration, otherwise it will update Δ_{y^*} and Φ_{y^*} and repeat the selection process.

When $|\mathcal{V}| = n$, Algorithm 8 has $2(n+k-1)$ times of evaluations of either $\sigma^2_{y^*|\mathcal{A}}$ or $\sigma^2_{y^*|\overline{\mathcal{A}}}$ in the best case. This is more efficient and scalable than Algorithm 7 when n becomes very large.

Algorithm 8 is not only efficient, but also provides its solution with a theoretic bound to the optimal solution. Although the mutual information function in (4.7) is not always increasing, the authors in [32] proved that it is still a partially monotonic submodular function. According to [44], a greedy algorithm, such as the Algorithm 8, which optimizes a monotonic submodular function guarantees a theoretical performance lower bound of $(1-1/e)$OPT, where OPT is the optimal solution value.

After a subset of observations at $\mathcal{A}$ is selected, the conditional probability distribution of an unobserved variable i given the observations of $\mathcal{X}_\mathcal{A}$ can be computed as $P(\mathcal{X}_i \mid \mathcal{X}_\mathcal{A})$. We will use its mean computed by (4.3) as the predictive value of the observation variable $\mathcal{X}_i$.

4.4 COMPARISON EXPERIMENTS IN WIRELESS SENSOR SCHEDULING

In this section, we will compare the probabilistic graphical model based approach and the Gaussian Process model based approach in solving the subset selection problem for scheduling a sensor. Specifically we want to select a subset of time points in size k out of totally 24 time points for turning on a sensor to sense and keeping it off at the other time. For convenience, in the following experiments, we will refer to the probabilistic graphical model based approach simply as the HMM based method, and the Gaussian Process based approach simply as the GP based method. The performance is measured using predictive accuracy for the unobserved time points in terms of the Root Mean Squares (RMS) error. The methods are compared for accurate observations, as well as erroneous observations.

4.4.1 EXPERIMENTAL SETUP

A hidden Markov model and a multivariate Gaussian model were trained using the temperature time series data collected in the Intel Berkeley Research Lab [8]. All the data were pre-processed for missing samples and discretized into 10 bins of 2 degrees Kelvin. The full data set consists of temperature samples combined from three neighbored sensors for 19 days. When training the chain graphical model, we set four latent states representing the different time periods from $12am-7am$, $7am-12pm$, $12pm-7pm$, and $7pm-12am$.

The whole data set was also randomly split into the test set and the training set with the ratio 1 : 9. An small error-injected test data set was also generated. The errors were taken randomly from a normal distribution with mean zero and variance 0.25.

We use these notations in the figures: "hmm_voidp" represents the HMM based selection approach by our improved VoIDP algorithm; "gp_entropy" represents the GP based selection approach by employing the entropy heuristic, and "gp_mutual" for the GP based mutual information heuristic.

4.4.2 Results and Discussion

Fig. 4.1(a) and 4.1(b) show the results on the full data set. Generally speaking, the more observations are selected the less RMS prediction errors are achieved for both the HMM based and the GP based approaches. In Fig. 4.1(b), the mutual information heuristic holds the competition with the entropy heuristic until about 10 observations are selected. We have mentioned that the mutual information function is not always monotonically increasing as the selection set gets bigger. The mutual information gains and its values are shown in Fig. 4.2 and Fig. 4.3, respectively. It can be seen that the mutual information heuristic lost its advantage after 12 observations are selected. It explains why the mutual information heuristic only has an advantage over the entropy heuristic between the 2 and 5 observations, but loses afterwards on the test data set, as shown in Fig. 4.1(c).

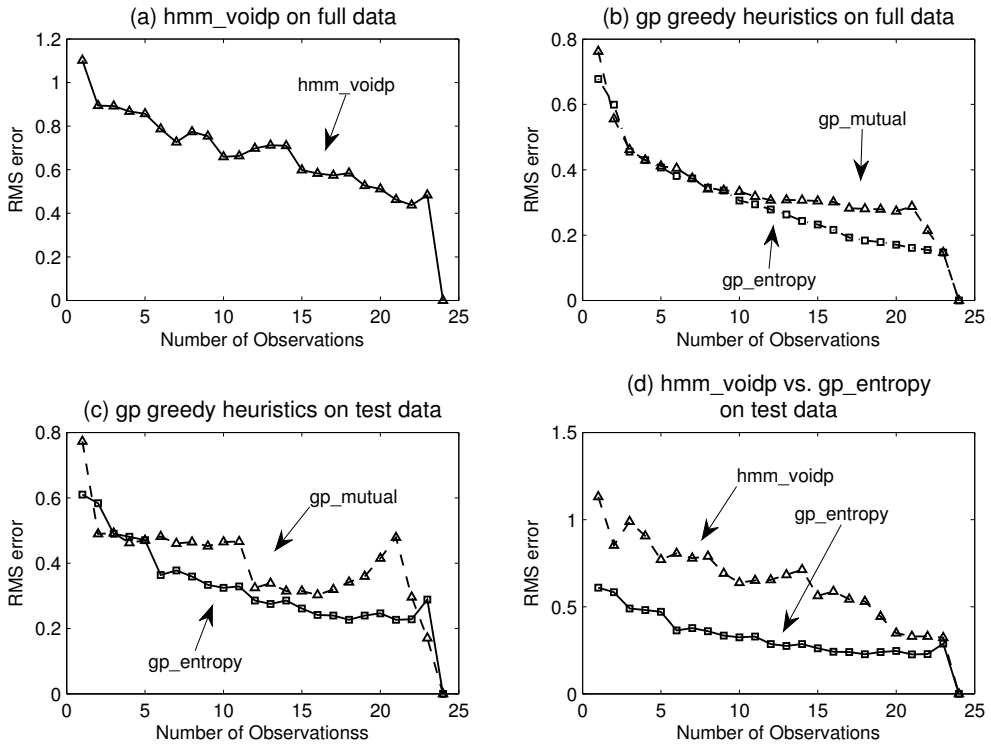


Figure 4.1: Prediction error vs. number of selected observations on full data set (a,b), and on test data set (c,d), given by the HMM based and GP based selection approaches.

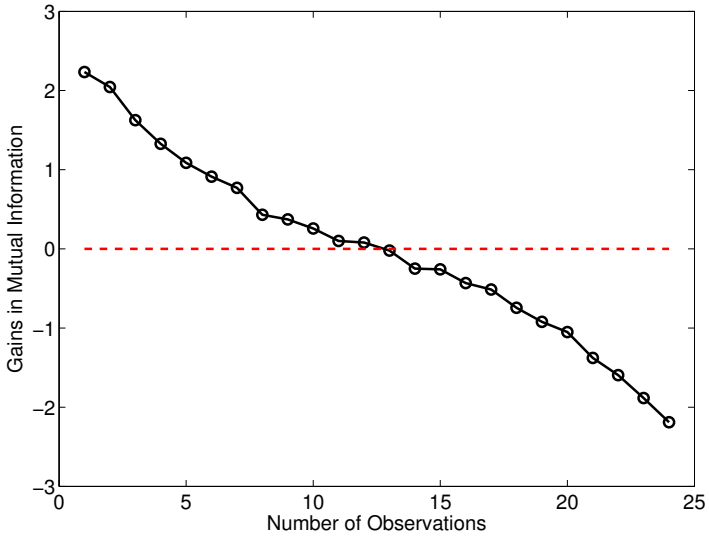


Figure 4.2: Mutual information gains on observations

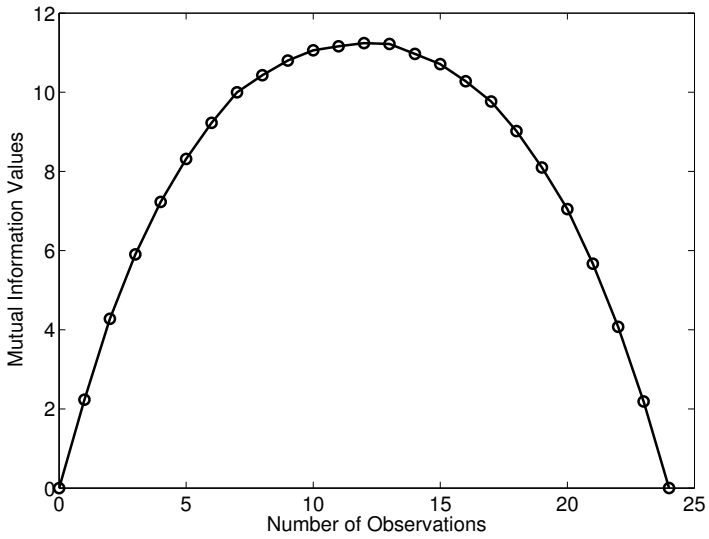


Figure 4.3: Mutual information values on observations

The GP entropy-based selection is compared with the HMM based selection on the test data in Fig. 4.1(d). The GP based approach beats the HMM based approach in terms of predictive accuracy on the test data.

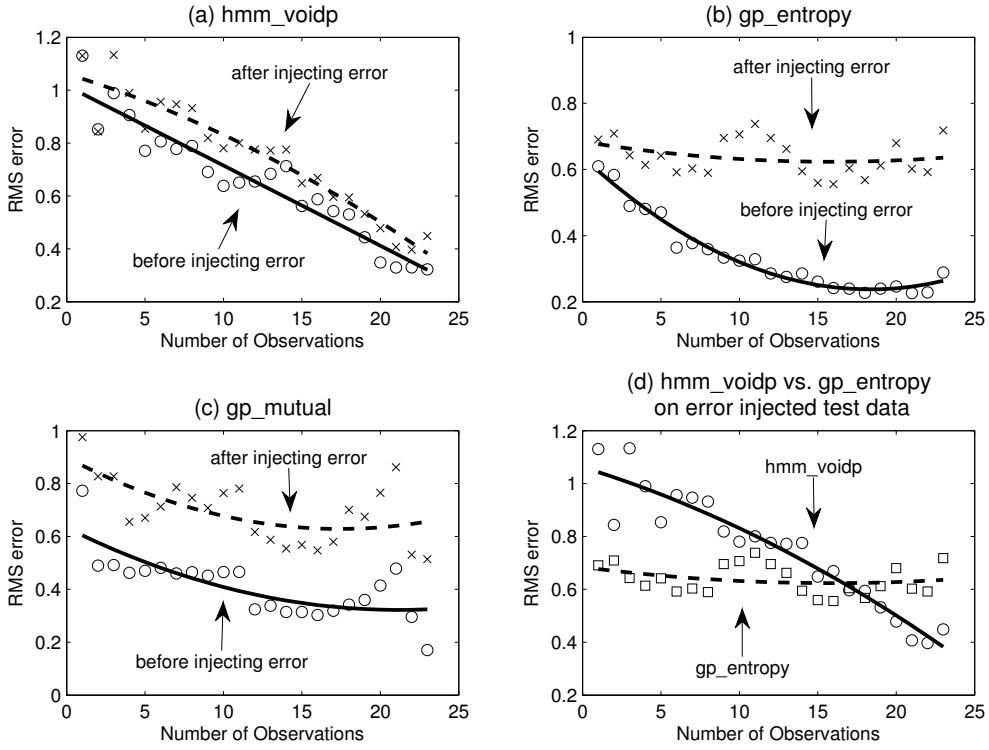


Figure 4.4: (a-c) Comparison of prediction errors on the original and error-injected test data by HMM based selection, GP based entropy heuristic, and GP based mutual information heuristic selections, respectively; (d) Comparing HMM based selection against GP based entropy heuristic selection on the test data with erroneous observations.

Fig. 4.4(a-c) examine how the erroneous observations affect these model based selection approaches. In the figures, the circles and crosses denote the results, the dash and solid lines

present the trends by doing regression based on the results. It shows that both GP based heuristic methods suffer big losses in predictive accuracy given the error-injected observations. However, the HMM based approach shows a very robust performance with a little increased of the prediction errors.

Despite both the GP based heuristics perform poorly with erroneous observations, the mutual information heuristic is slightly more stable. In Fig. 4.4(c), the two trend lines are parallel to each other. Whereas Fig. 4.4(b) shows a big difference in performances by the entropy heuristic before and after the errors are injected into the observations.

The GP based entropy heuristic is compared with the HMM based selection under the erroneous observations because of its relatively lower prediction error over the mutual information heuristic. Fig. 4.4(d) illustrates the result. It shows that the GP based entropy method tends to maintain a constant error as the number of observations increases, meaning more observations do not help the predictive accuracy. Conversely, the HMM based approach has a decreasing trend on predictive accuracy as more observations are added. After 16 observations the HMM based selection method outperforms the GP based method in terms of the RMS prediction errors.

The robustness against observation errors exhibited by the HMM based selection approach is partly attributed to the conditional independence property provided inherently by the corresponding probabilistic chain graphical model. The observations on the chain graph-

ical model cut the entire chain into smaller sub chains, and the observations on one sub chain will not directly affect the predictions on the other sub chains. This property minimizes the effect caused by the erroneous observations. But for the GP based approach, the predictions take all the observations directly into account, which therefore increases the chance of being affected by the errors in observations.

4.5 CONCLUSIONS

In this chapter, we tackle the sensor scheduling problem by selecting a subset of time points to observe in order to make the most accurate predictions for the unobserved time points. We compare two model based selection approaches, the probabilistic graphical model based, particularly with HMM, and the Gaussian Process model based employing both entropy and mutual information greedy heuristics.

The results show that the GP based approach performs better than the graphical model based approach in terms of the predictive accuracy with accurate observations. But when small errors are injected into the observations, the GP based selection method performs very poorly. In contrast, the graphical model based approach demonstrates more robust and stable performance given the erroneous observations, and outperforms the GP based approach when more observations are selected.

CHAPTER 5

APPLICATION OF THE OBSERVATION SELECTION METHODS TO FEATURE SELECTION FOR CLASSIFICATIONS

5.1 INTRODUCTION

We propose to apply the observation selection methods introduced in Chapter 3 to select features for classification problems. Selecting features for classification is similar to selecting sensor observations in wireless sensor networks. They share the similar purpose of removing redundant or noisy attributes and selecting out the valuable information that most independent to each other or beneficial to classification or prediction models. Observation selection based on optimizing submodular mutual information is efficient and effective

as shown in Chapter 3. Its computational efficiency and near-optimal theoretical guarantee motivate us to applying it in feature selection for classification problems. We explore feature selection based on submodular mutual information and entropy driven methods by comparing its performances with other feature selection methods based on attribute ranking and matrix decomposition, in varieties of classification methods and multiple classification problems.

5.2 FEATURE SELECTION AND ITS BACKGROUND

Feature selection is the process of selecting a subset of input variables or attributes and using only the subset as features fed into classification [5] methods. It servers two main purposes. First, it makes training and applying a classifier more efficient by reducing the high dimensionality of feature sets. Second, feature selection often improves classification accuracy by eliminating irrelevant or noisy features. Moreover, as high dimensional and large data sets become increasingly common in computational fields, feature selection plays an key role for machine learning or data mining algorithms to make its become efficient and scalable. The excellent surveys of and introduction to feature selection are given in [16, 64, 40, 39].

Feature selections have been utilized and proved effective to varieties of applications ranging from text mining, image retrieval, intrusion detection, genome analysis, and etc. [21,

10, 55, 37, 45]. Algorithms of feature selections can be generally divided into two categories, the filter approach or the wrapper approach [6, 24]. The filter approach depends on characteristics of training data to select features without any learning process. The wrapper approach applies a learning algorithm to evaluate feature selections. The wrapper approach usually gives better feature selection for classification accuracy than the filter approach does, but the latter is usually more computational efficient than the former.

Another way of categorization for feature selection algorithms is based on its returned results that may either be weights of all features or a subset of selected features. Accordingly, feature selection algorithms can be sorted into two types, feature weighting and subset selection. The entropy and submodular mutual information based approximate algorithms for selecting optimal information introduced in Chapter 3 will fall into the subset selection category if they are used to select features.

5.3 MOTIVATION

Mutual information [5, 2] has been used for feature selection in pattern recognition and classification. It measures how much information the presence/absence of an input variable or a subset of variables contributes to make the correct classification decision. It is essentially equivalent to an entropy reduction, and it has been seen that the mutual information criterion-based selection works better than the entropy criterion-based selection [47, 32].

However, the problem of maximizing the mutual information itself is a NP-hard problem. Due to the problem's complexity, it's not expected to find optimal solutions in polynomial time. Current methods of solving the problem are either inefficient such as by simulated annealing and hill climbing, or lack of theoretical guarantees on the performances of its solutions. Submodularity of mutual information ensures its greedy method to have properties of both efficiency and near optimality. That's our motivation of applying submodular mutual information-based method for feature selection.

5.4 SUBMODULAR MUTUAL INFORMATION-BASED FEATURE SELECTION AND OTHER SELECTION METHODS

Submodularity optimization plays a vital role in developing an efficient and near-optimal approximate algorithm for maximizing submodular mutual information. The concept of *submodularity* was originally introduced by Nemhauser et al. [44]. A set function F is *submodular*, if for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ and $i \in \mathcal{V} \backslash \mathcal{B}$ it holds that $F(\mathcal{A} \cup i) - F(\mathcal{A}) \geq F(\mathcal{B} \cup i) - F(\mathcal{B})$. It demonstrates an inherent property of a submodular function, *diminishing returns*. That is, adding another observation to a smaller subset of observations helps more than adding it to a larger subset. A greedy algorithm schema, selecting the element i^* that maximizes $F(\mathcal{A} \cup i) - F(\mathcal{A})$, guarantees that its solution is as good as at least $(1 - 1/e)OPT$, where the OPT is the function value for the optimal selection. Mutual information functions can be

considered as submodular functions [32], and thus the greedy algorithm can be applied to solve feature selection problem based on mutual information criterion. The benefit of the submodularity-based greedy algorithm is efficient and guarantees its solutions sitting within a theoretical bound to the optimal solution.

We have applied an computationally efficient version of the algorithm for observation selection in wireless sensor networks with an application to place road traffic monitoring sensors as in [48] and in Chapter 3. Its efficiency and near-optimality properties motivate us to employ it in feature selection for classification problems.

Besides the submodular mutual information-based and entropy-based selection methods, we also compare them with two other popular selection methods. One is for matrix column subset selection method, the rank-revealing QR (RRQR), and the other is the ranker attribute search method in the data mining software Weka [17].

In Column Subset Selection problem, given a matrix A and an integer k, to determine a permutation matrix P so that $\mathrm{A}P = (\mathrm{A}_1\ \mathrm{A}_2)$, and A_1 has k columns. A_1's columns are important because they should be very linearly independent. On the contrary, A_2's columns are redundant because they can be well represented by those in A_1. The resulting permutation P can be seen as a ranking of the column attributes for the matrix A. A well known method to solve the problem is called Rank-Revealing QR, a matrix factorization-based method [20, 41, 56, 15]. Essentially, whatever it is observation selection in wireless sensor networks or

feature selection for classification, the problems can all be represented by data matrices with variables or attributes as matrix columns. That's why they can be seen as a mathematical problem that can be solved by the RRQR method.

Weka is a very popular data mining software and is being widely used. It is an open-sourced java-based software that implements a lots of machine learning and data mining algorithms for classifications. It also has attribute selection methods. For example, the ranker search method that can be used with an evaluation method module to rank all the attributes in a data set by their information gains. Our experiments of comparing attribute selection methods on classification problems are heavily rely on the Weka's existing implementation of those algorithms.

Among these feature selections methods as also shown in table 5.1, the submodular mutual information-based, entropy-based, and the RRQR methods can be categorized into filter-based methods. Their evaluations are not involved with the class labels of a data set. Their directly purposes are to remove redundant attributes and find out the most independent and the best representing attributes out of the data set. Classification methods will take attribute-filtered data sets to calculate its classification accuracy. The Weka's ranker method is a wrapper-based feature selection method, because its attribute-ranking evaluation is involved with class labels. Its advantage is that the attribute selection process is directly related to classification rates, however its disadvantage is that it can not detect redundant attributes,

which means it will allow to select correlated attributes. It's interesting to compare these two types of selection methods under the same experimental setting.

In the following section, we will describe our experimental settings, and briefly introduce the classification methods in Weka that have been incorporated into the experiments.

5.5 EXPERIMENTAL SETTING

In the experiments, we employed totally 4 different attribute selection methods (as shown in table 5.1), 11 different classifiers (as shown in table 5.3), and 13 different data sets (as shown in table 5.2).

In the selection methods, the mutual information-based and entropy-based methods were introduced in the Chapter 3 where they were used to select optimal spatial observations for applications in wireless sensor networks. The RRQR method is based on matrix factorization, and it has been briefly described in the section 5.3. We used a public matlab implementation of the RRQR provided in [1]. The Weka's ranker selection method is to rank attributes based on their individual evaluation values. In running experiments, we used the "ranker" method combined with the default attribute evaluation function "ReliefF" in Weka software. It's an instance-based evaluation method. It checks neighboring instances of the same and different classes, and adjusts attribute weights accordingly. The reason we chose the ranker method is that it can return a ranked list including all the attributes, whereas other attribute

selection methods such as best-first only select a subset of the attributes. Our running experiments systematically swept attribute selection size from 1 to $N-1$, in which N is the size of whole attribute set.

Id	Selection method name
1	Mutual information-based (MI)
2	RRQR
3	Weka's ranker
4	Entropy-based

Table 5.1: Selection methods

The data sets in the table 5.2 covers a variety of situations in terms of number of classes and attributes. For those data sets not providing test sets, we divided the original data sets into two parts, two-third of which for train sets and one-third for test sets. Most of the data sets are from UCI's machine learning repository [11], some data sets are also from Chih-Jen Lin's Libsvm data web page [4]. For every data set, all its attributes including class labels were already converted into numeric values when we used them.

The classification methods used in our experiments are shown in table 5.3. Basically, we pick up one or two representatives in each of the categories of Weka's classifiers, we believe that this collection covers a wide spectrum of classification methods available in Weka. Their brief descriptions are also listed in the section 5.5.1. Since all attributes of every data set were converted into numeric values beforehand, the classification methods work here as in regression process. They take a filtered data set with selected attributes

ID	Dataset name	class num	attribute num	train instance num	test instance num
1	australian credit	2	14	460	230
2	diabetes	2	8	507	261
3	glass	6	9	142	72
4	liver disorders	2	6	230	115
5	satimage	6	36	2217	1000
6	vehicle	4	18	564	282
7	breast cancer	2	9	455	227
8	german credit	2	24	667	333
9	heart	2	13	180	90
10	pen digits	10	16	2623	1225
11	sonar	2	60	138	70
12	wine	3	13	118	60
13	dna	3	180	2000	1186

Table 5.2: Data sets

as input, and predict class variable of each test instance as a numeric value. The numeric class prediction is first converted into a relevant class label number that is closest to it by numerical distance. Then an individual classification rate or accuracy within every class of test instances can be calculated based on the number of correctly predicted instances divided by total instance number in the class. The final classification rate or accuracy for the data set is an average of all of its individual classification rates.

A batch of experiments was run on each of the data sets by applying all the listed attribute selection methods and classification methods. For each of the data sets, it applied by all the listed classification methods. Within every classifier, it tried all of the listed selection methods respectively. Also for each selection method, it tried out the selection size from 1

Id	Classifier name in Weka
1	RBFNetwork
2	GaussianProcesses
3	SimpleLinearRegression
4	PaceRegression
5	SMOreg
6	KStar
7	AdditiveRegression
8	Bagging
9	RandomSubSpace
10	DecisionTable
11	M5P

Table 5.3: Classification methods

to $N-1$ where N was the total number of attributes. All the classifiers were used with their default options provided by Weka during experiments. A resulted classification rate was an average of individual classification rates for each of the classes.

The following sub-section describes briefly the classification methods in Weka that we used in our experiments.

5.5.1 Classification Methods in Weka

Weka's classifiers are divided into several categories: Bayesian, functions, lazy classifiers, meta learner, multiple-instance classifiers, miscellaneous classifiers, rules, and trees. Within each of the categories, it includes numerous individual classifier implementations. In our experiments, we chose to use a collection of classifiers that covers the spectrum of the

categories. We give brief descriptions below for the classifiers.

1. RBF network

   It builds a Gaussian radial basis function network. It uses the k-means clustering algorithm to provide the basis functions that serves as hidden units on the hidden layer. In every cluster, the data are fit into multivariate Gaussian distributions. For numeric class problem, its prediction is given by a linear regression that combines outputs from the hidden layer.

2. Gaussian Processes

   It builds a nonlinear regression classifier using the Bayesian Gaussian process technique.

3. Simple Linear Regression

   It builds a linear regression model based on a single attribute. The attribute is chosen in terms of the smallest squared predictive error.

4. Pace Regression

   It builds pace regression linear models. It can determine the attributes to be used for the models. Under certain regularity conditions then it is provably optimal when the number of attributes tends to infinity.

5. SMOreg

   It implements a sequential minimal optimization algorithm for learning a support vector regression classifier using kernel functions.

6. KStar

   It is an instance-based classifier. It determines the class of a test instance based on the class of those training instances similar to it, as determined by some similarity function. It differs from other instance-based learners in that it uses an entropy-based distance function.

7. Additive Regression

   A meta classifier that enhances the performance of a regression base classifier. Each iteration fits a model to the residuals (The predictive errors) left by the classifier on the previous iteration. Prediction is accomplished by adding the predictions of each classifier.

8. Bagging

   An ensemble approach. It builds multiple variants of a classifier using bags of training samples, and classify a test instance using a weighted vote among the variants of classifiers. The default base classifier is a decision tree.

9. Random SubSpaces

   It builds an ensemble of base classifiers, each trained using a randomly selected subset of the input attributes. The default base classifier is a decision tree.

10. Decision Table

    It builds a decision table majority classifier. It evaluates feature subsets using best-first search.

11. M5P

    A learner to build a model tree. For a test instance, its attributes will route itself through the nodes of the tree, and make it down to a leaf. The leaf contains a linear model based on some of the attribute values, which can yield a predicted value for the test instance.

The classifiers above were those that we used in the following experiments. For more detailed information about these or other classifiers in Weka, please refers to [60, 62, 63].

The experimental results are shown in the following section. For convenience, the selection methods, classifiers and data sets are represented by their ID numbers as used in the tables above.

5.6 EXPERIMENTAL RESULTS AND DISCUSSION

For each of the data sets (see table 5.2), we ran all of the listed classifiers (see table 5.3). For each of the classifiers, we ran all of the listed attribute selection methods (see table 5.1). For each of the selection method, all possible selection size was swept through, that is, the selection size went from 1 to $N-1$ if N is the full attribute size. For comparison, using full attribute was also examined for each of the classifiers.

Given a data set and a classifier, we picked the best performed selection method in terms that it should have the highest classification accuracy rate and its attribute selection size should be as small as possible. Then we counted it as a winning for the selection method. To accommodate the very close performances, we allowed selection methods as in the same winning situation if their classification rates' difference was within 0.5% and also they shared the same selection size for scoring the classification rates.

The winning counts after all the runs are summarized in the table 5.4, and are also visualized in Fig. 5.1. It can be seen that all of the selection methods take up a variety of portions on the winning bars for each of the classifiers. It's hard to tell which classifier is favored by a particular selection method, and vice versa. But there is a strong evidence showing that the winning odds of using full attributes for classification is much lower than using an selected attribute subset. The full-attribute selection method only shows up on three of the bars.

In the summarized winning cases, we want to see how they are distributed among the

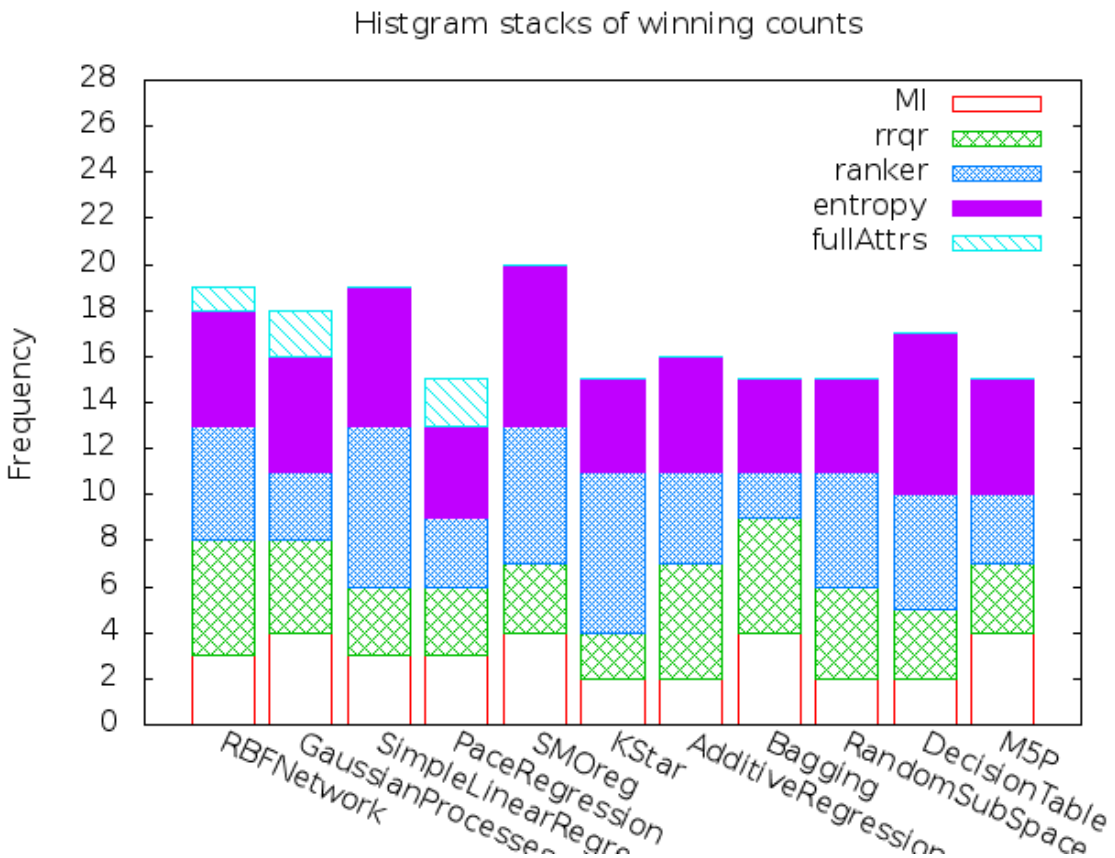


Figure 5.1: Stacked bar chart of the winning counts

classifier method	MI	rrqr	ranker	entropy	fullAttrs
RBFNetwork	3	5	5	5	1
GaussianProcesses	4	4	3	5	2
SimpleLinearRegression	3	3	7	6	0
PaceRegression	3	3	3	4	2
SMOreg	4	3	6	7	0
KStar	2	2	7	4	0
AdditiveRegression	2	5	4	5	0
Bagging	4	5	2	4	0
RandomSubSpace	2	4	5	4	0
DecisionTable	2	3	5	7	0
M5P	4	3	3	5	0
total	33	40	50	56	5
percentage %	17.9	21.7	27.2	30.4	2.7

Table 5.4: Summary of winning counts

data sets. Fig. 5.2 shows the distributions for each of the selection methods including using full attributes. It shows that data sets do affect the performance of the attribute selection methods. Based on the histogram charts, there are missing cases for some selection methods such as mutual information-based, RRQR and Weka's ranker methods. It also shows that different data sets could cause a lot of variations about performance of selection methods such as for the RRQR and the entropy-based methods.

Fig. 5.3 shows accumulated counts of winning classification methods associated for each of the selection methods. It demonstrates cooperating versatility with classifiers for every selection method, and it reveals limitation for using full attributes.

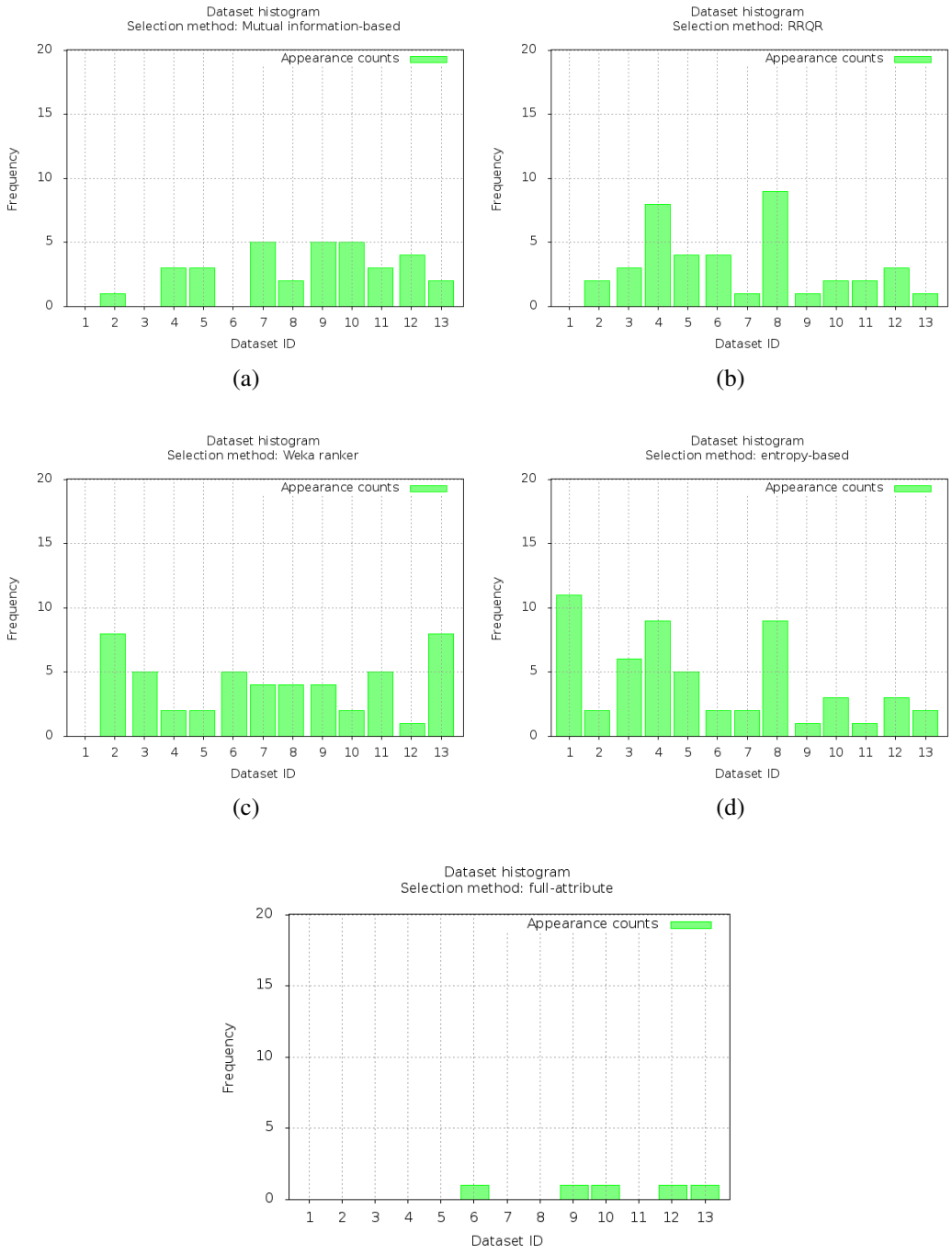


Figure 5.2: Histogram of datasets appearance in winning counts

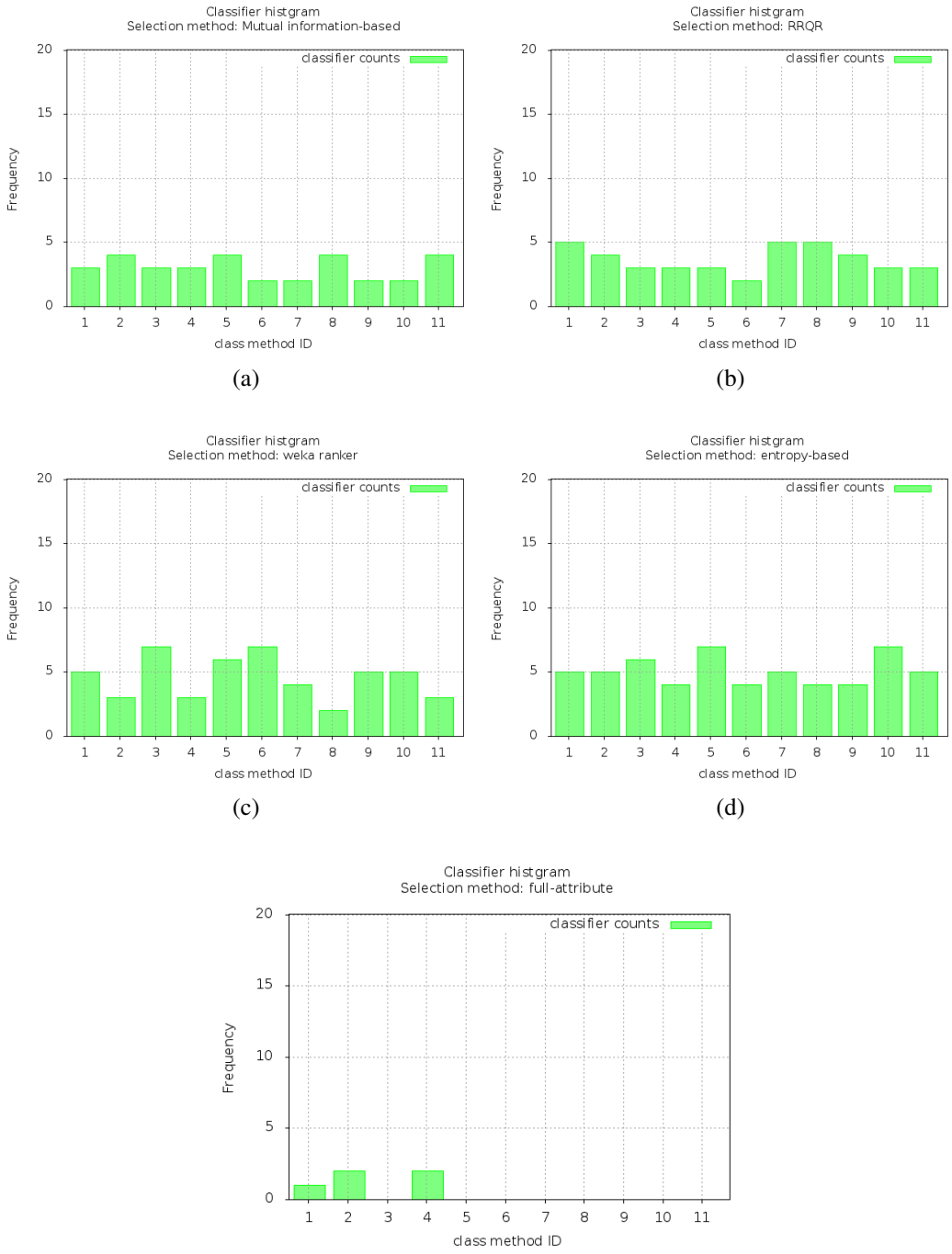


Figure 5.3: Histogram of classification method appearances in the winning counts

Selection size is a key parameter for the experimental attribute selection methods, it is interesting to learn how these attribute selection methods perform according to different selection sizes. Since we enumerated all selection size numbers for every selection method given a classifier and a data set, we summed up the selection size numbers recorded in each selection method's winning cases. Fig. 5.4 shows the distributions. For all the selection methods, it looks that their winning frequencies are varied from changing selection sizes. It's hard to guess about the reason that causes the variations. But one thing we notice is that the peak time for the mutual information-based selection method takes place on when the selection size accounts for about half of a whole attribute set. From the Fig. 5.4a, it indicates that the winning peak time corresponds to the selection size falling in between 50% and 60%.

The spike of winning counts in half-sized attribute selection given by the mutual information-based method looks phenomenal to us. It reminds us of the special property about mutual information. We have seen in Chapter 3 that when we used the submodular mutual information-based method to select traffic monitoring sensors, the mutual information gains went up until its selection size exceeded some middle point of total number of available sensors, then the gains dropped afterwards. We therefore wondered whether this is somehow related to the phenomenon displayed here in Fig. 5.4a.

We drew the mutual information values for each of the data sets, and display them on Fig. 5.5, 5.6,and 5.7. It is not surprised to see bell shapes for all of them. It says that mu-

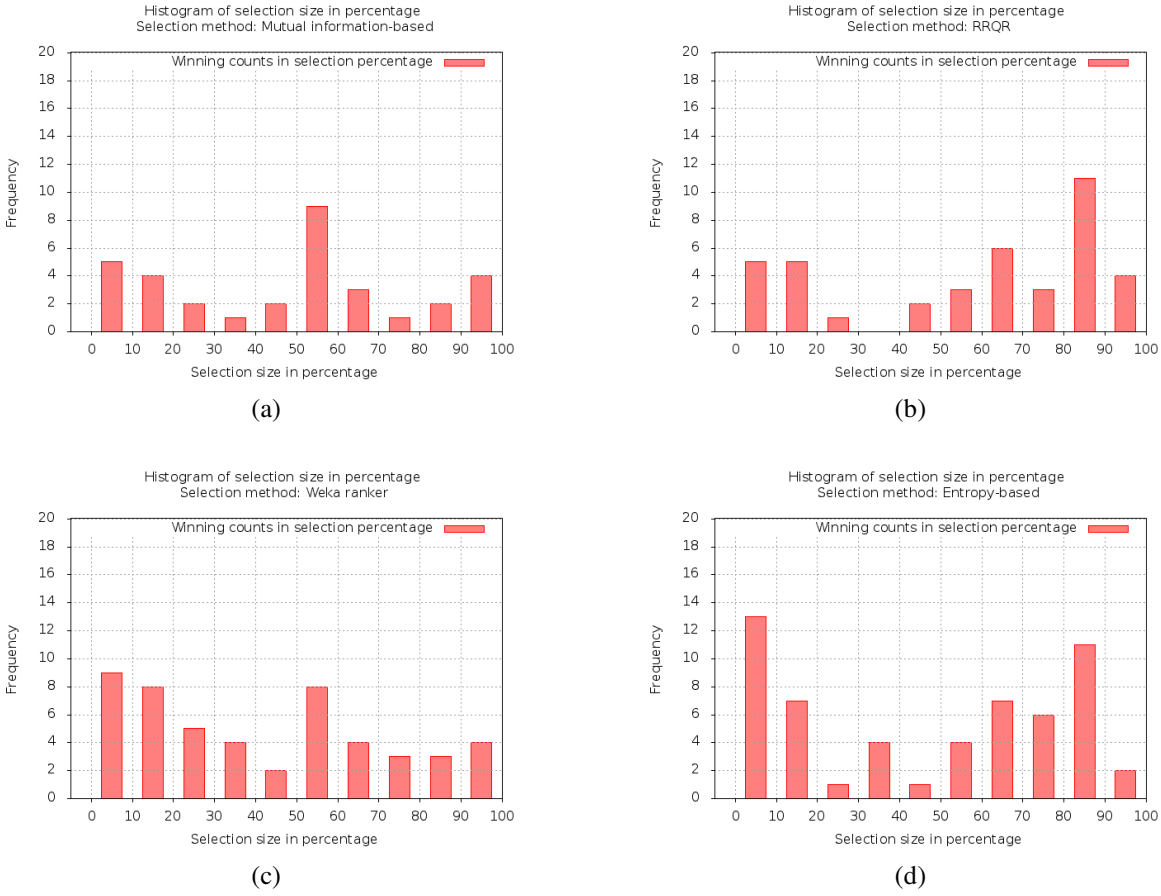


Figure 5.4: Histogram of selection size percentage in winning counts

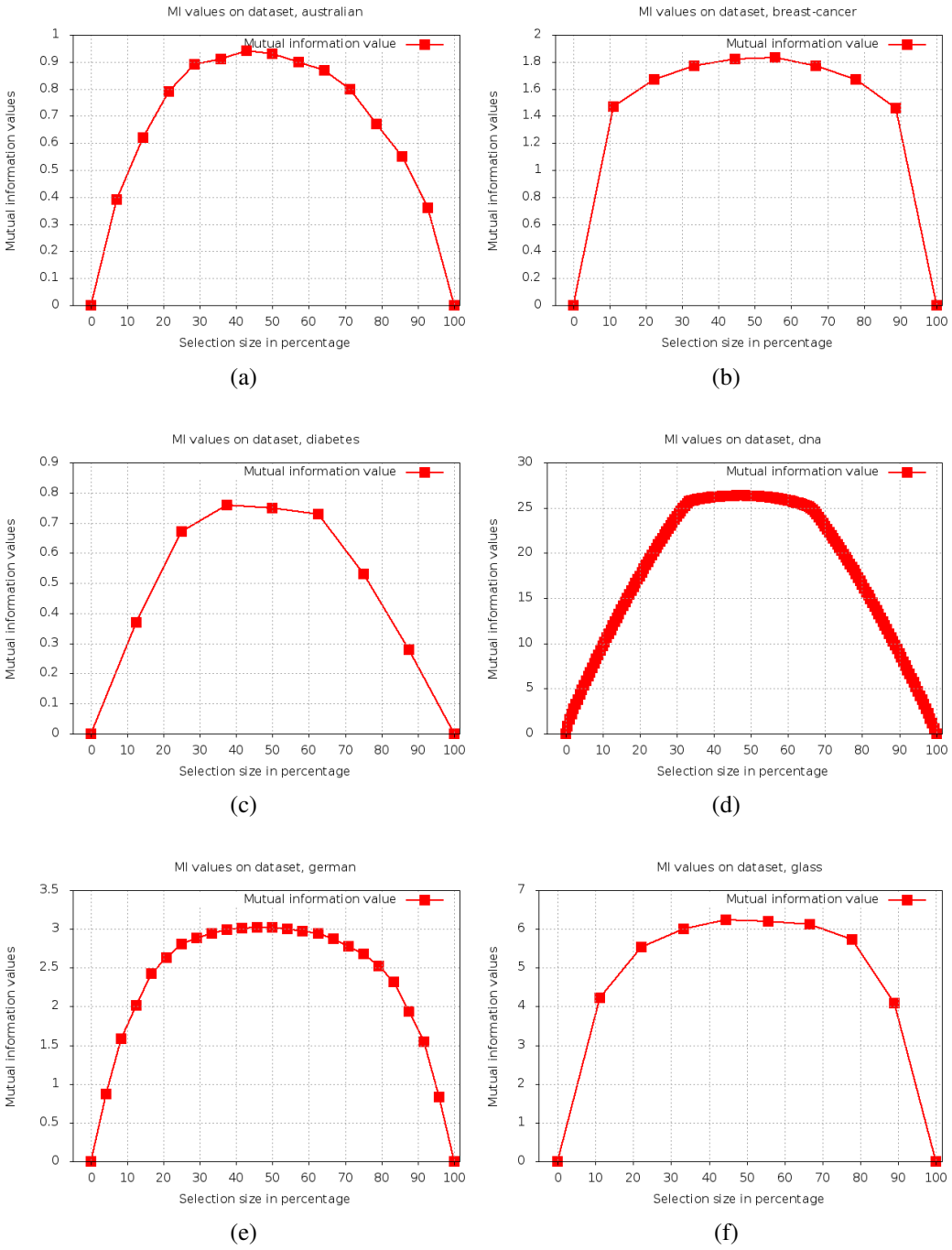


Figure 5.5: Mutual information values for datasets, part-1

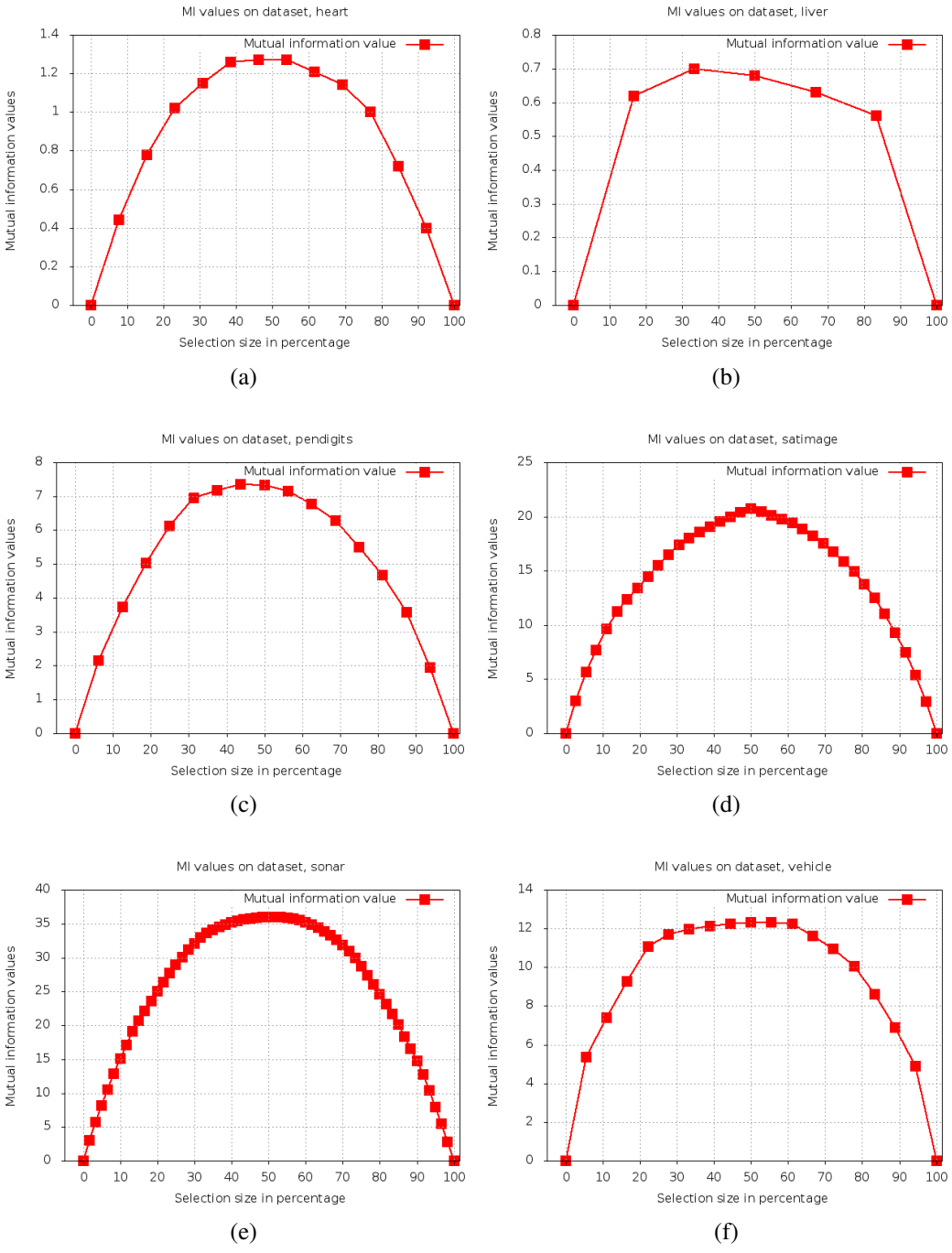


Figure 5.6: Mutual information values for datasets, part-2

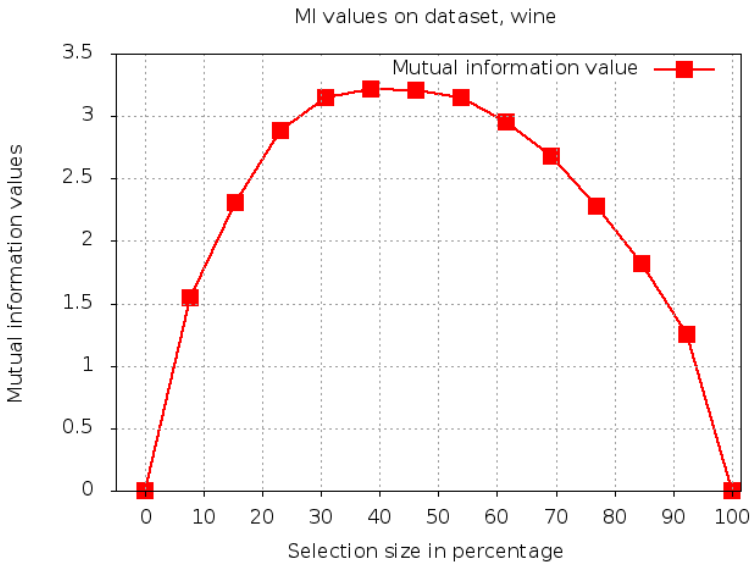


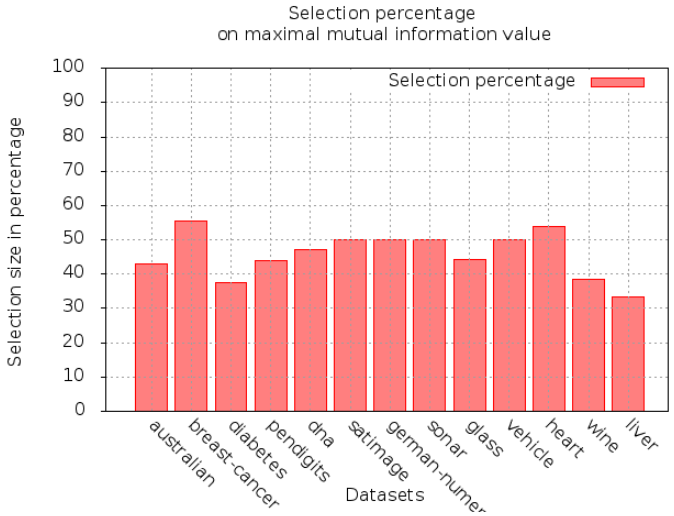
Figure 5.7: Mutual information values for datasets, part-3

tual information values keep increasing but when the selection size gets bigger and exceeds about the half in whole size, then it becomes decreasing. The selection sizes achieve the maximal mutual information values for each of the data sets are displayed in Fig. 5.8a. The selection size quantities were converted into percentages out of whole sizes for comparison convenience. They are all around 50% which is about in half size. For those data sets that contain small numbers of attributes, the percentage numbers may look lower compared to others. That's because adding or removing even a single attribute will reflect on varying differences in the percentage representation. For example, to the liver data set that has totally 6 attributes, selecting 2 attributes accounts for about 33%, whereas selecting 3 makes

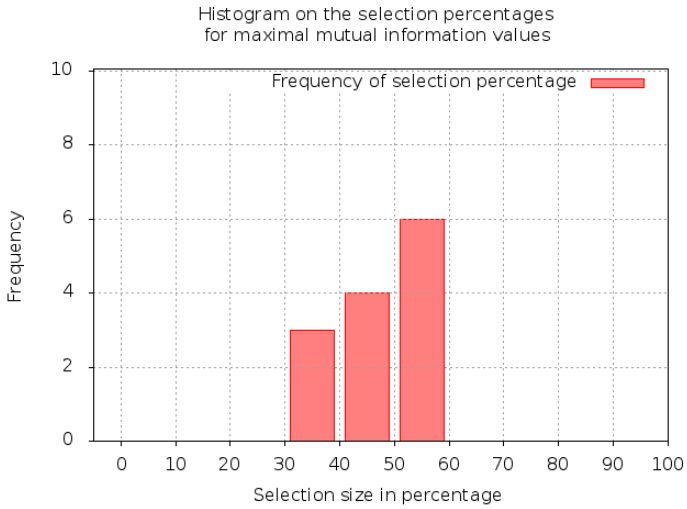
it to jump up to 50%. Fig. 5.8b shows a histogram that is a summary of accumulated counts for the data in Fig. 5.8a. The spike falls into the interval of between 50% to 60%. Does this interval have anything to do with the same interval appeared in Fig. 5.4a? We will look into detailed experimental results in the breast-cancer and heart data sets. We chose these two data sets to look further because their percentage values of selection size as shown in Fig. 5.8a fall into the $50\% - 60\%$ interval.

Fig. 5.9 and 5.10 shows all the classifier rates combined with all the selection methods on the breast cancer and the heart data sets. The selection sizes were set to 55.6% for breast cancer data set, and to 53.8% for heart data set, where the mutual information-based selection method reached its maximal mutual information values. The performances given by using full attributes were also drawn for comparison purpose.

The mutual information-based attribute selection method (It is abbreviated as MI-based in the drawings) performed very competitive on these two data sets. It can be seen from Fig. 5.9 that the MI-based method is applied on all the classifiers, and more than half of the resulted classification rates poise to reach the best compared to other selection methods given same classifiers. The best performances across all the classifiers are projected on the first two classifiers. One is with RRQR selection method on RBFNetwork classifier, and the other is with MI-based method on GaussianProcesses. Their scores are too close to tell. Using the full-attribute on GaussianProcesses also gives almost indiscernible improvement on the MI-



(a)



(b)

Figure 5.8: Summary of selection size in percentages for maximal mutual information gains

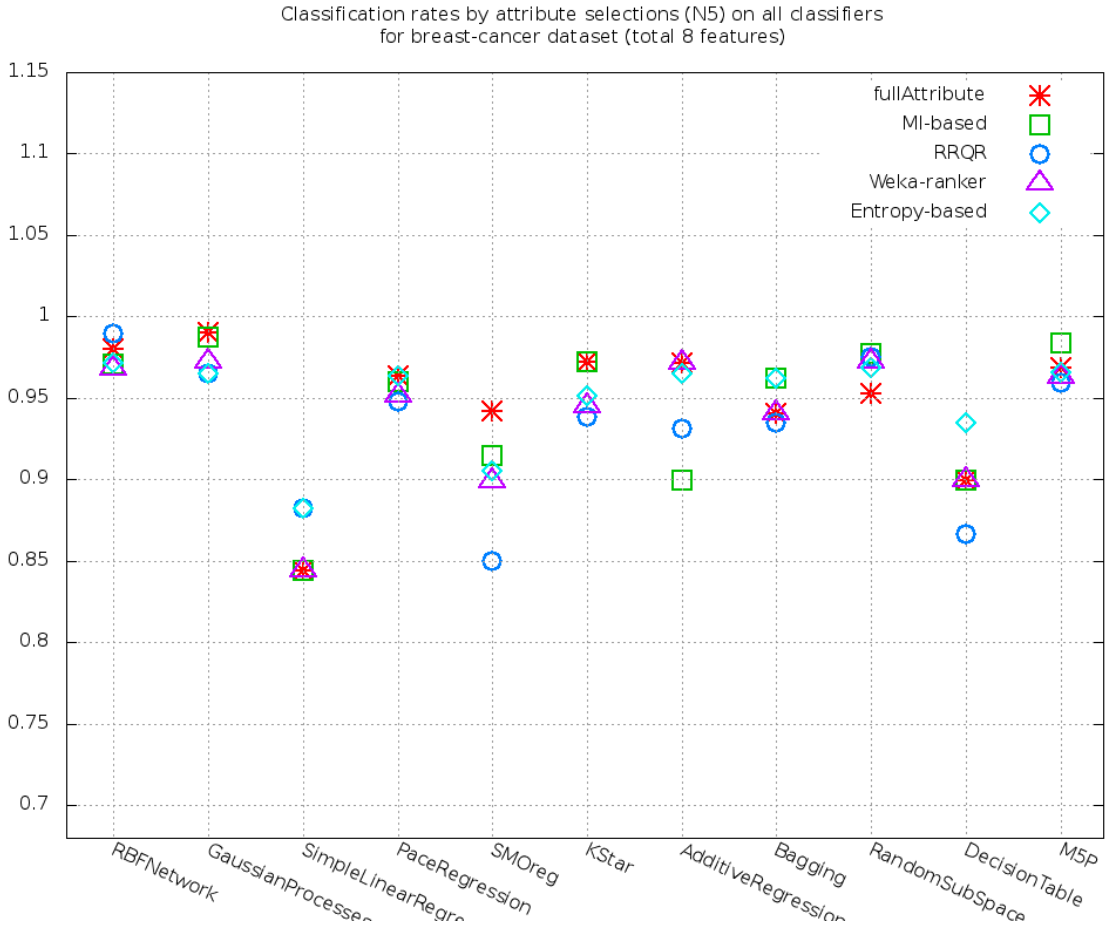


Figure 5.9: Classification accuracy rates on breast cancer data set with selection size in 55.6%

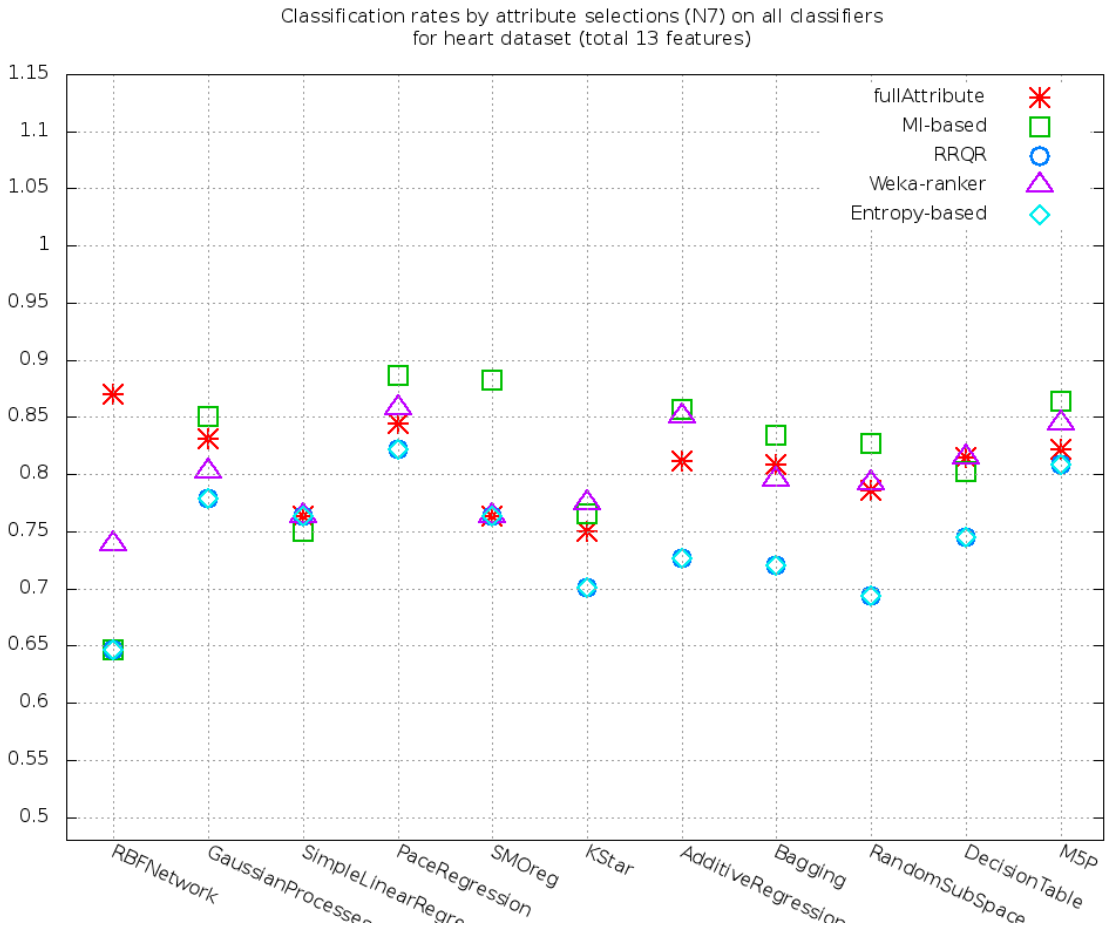


Figure 5.10: Classification accuracy rates on heart data set with selection size in 53.8%

based method. For the heart data set, Fig. 5.10 observes similar trend, more than half of the classification rates given by MI-based selection method for all the classifiers are among the best rates. The best performances are given by the MI-based method with PaceRegression and SMOreg classifiers. Their leading roles are very clear in the drawing.

Selection method	Size (%)	Attribute index	Classifier ID	Classification rate
MI	55.6	3 4 5 1 7	2	0.99
RRQR	55.6	1 6 8 4 5	1	0.99
MI	55.6	3 4 5 1 7	11	0.98
MI	66.7	3 4 5 1 7 9	9	0.98
Weka-ranker	77.8	6 1 8 4 3 2 5	6	0.98
MI	44.4	3 4 5 1	8	0.97
Weka-ranker	55.6	6 1 8 4 3	7	0.97
MI	44.4	3 4 5 1	4	0.96
Entropy-based	55.6	6 8 1 4 2	10	0.94
Weka-ranker	66.7	6 1 8 4 3 2	5	0.94
Weka-ranker	11.1	6	3	0.88
Entropy-based	11.1	6	3	0.88

Table 5.5: Best accuracy rates by classifiers on breast cancer dataset

After examining the situation with the certain selection sizes that achieves the maximal mutual information gains, we wanted to investigate throughly with varied selection sizes and different classification methods for the breast cancer and the heart data sets, and to see how performances of MI-based method with the particular selection size are being ranked. Tables 5.5 and 5.6 contain the results. It shows that the best classification scores for each of the classifiers, and the corresponding selection method with its selection size in percentage.

Selection method	Size (%)	Attribute index	Classifier ID	Classification rate
MI	53.8	10 8 2 12 4 7 3	4	0.89
MI	76.9	10 8 2 12 4 7 3 6 5 9	11	0.89
MI	53.8	10 8 2 12 4 7 3	5	0.88
Weka-ranker	61.5	3 9 12 10 13 4 6 1	9	0.87
Full-Attribute	100	1 2 3 4 5 6 7 8 9 10 11 12 13	1	0.87
Weka-ranker	38.5	3 9 12 10 13	7	0.86
MI	53.8	10 8 2 12 4 7 3	2	0.85
MI	53.8	10 8 2 12 4 7 3	8	0.83
RRQR	69.2	5 8 4 1 13 10 7 3 12	10	0.83
Entropy-based	69.2	5 8 4 1 13 10 7 3 12	10	0.83
Weka-ranker	30.8	3 9 12 10	6	0.82
Weka-ranker	7.7	3	3	0.77

Table 5.6: Best accuracy rates by classifiers on heart dataset

The results were sorted based on the classification accuracy rates, and within a same rate category it was then sorted by the selection size. The rule is that the one with the highest classification accuracy rate and the lowest selection size will be ranked on top. From both of the tables 5.6 and 5.5, the MI-based methods with about half-sized selections are ranked on the top.

Based on the results shown in Fig. 5.9 and 5.10, and in tables 5.5 and 5.6, it tells that the highest frequency in Fig. 5.8b does have something to do with the highest frequency appeared in Fig. 5.4a. For both the breast cancer and the heart data sets, their MI-based selection sizes achieving the maximal mutual information values fall into the $50\% - 60\%$ interval, and the MI-based method with these selection sizes performed not only best across

the horizon based on the same selection size, but also the best overall for all situations with various classifiers and different selection sizes.

Although these evidences have answered the question that we raised in early, using MI-based method with the chosen selection size achieving the maximal mutual information value may not provide the best classification rate for all the data sets. We will show a summary table in the following that lists the best classification accuracy rates achieved for each of the data sets we used in the experiments. However, the successful evidence demonstrated with the breast cancer data and the heart data shows this approach with the mutual information-based selection method is worthwhile, and it may grab a trophy for you on some classification problems.

Table 5.7 summarizes out of all the experiments the best classification performances for each of the data sets. It includes the corresponding attribute selection method, the selection size in percentage, and the classifier. It shows that there is no magical selection method can win out for all of the data sets. It's also interesting to see different methods could select the subsets of attributes that result in the same classification performance. For example, both the RRQR and the entropy-based methods match each other in the data set #4. The only winning case for using full attributes in the experiments happens on the data set #10, which is the pen-digits data set (refers to the table 5.2), with the KStar classifier (refers to the table 5.3). All the other winning cases are achieved by the feature selection methods along with specified

DatasetID	Best rate	Selection method	Selection size (%)	ClassifierID
1	0.90	Entropy-based	35.7	4
2	0.74	Weka-ranker	37.5	9
3	0.72	Weka-ranker	55.6	6
4	0.71	RRQR	83.3	2
	0.71	Entropy-based	83.3	2
5	0.85	Entropy-based	72.2	6
6	0.76	Weka-ranker	88.9	11
7	0.99	MI-based	55.6	2
	0.99	RRQR	55.6	1
8	0.69	Entropy-based	54.2	8
9	0.89	MI-based	53.8	4
	0.88	MI-based	53.8	5
10	0.94	Full-attribute	100.0	6
11	0.93	MI-based	56.7	2
12	0.99	RRQR	76.9	6
13	0.94	Weka-ranker	25.6	11

Table 5.7: Summary of the best classification accuracy rates for each of the data sets

Selection method	Count	Percentage
MI-based	4	25%
RRQR	3	18.75%
Weka-ranker	4	25%
Entropy-based	4	25%
Full-Attribute	1	6.25%

Table 5.8: Summary of the selection methods' winning counts in data level

classifiers.

Table 5.8 shows the accumulated appearance counts of respective selection methods from the table 5.7. Every selection method takes a fair portion except that using full attributes only accounts for once, a very small number comparably. This is again a good evidence about that feature selection can help classifiers to outperform of using full features. Based on the percentage numbers shown in table 5.8, the MI-based selection method apparently poises itself as a strong competitor compared to other selection methods. Another observation is that both filter-based selection methods such as the MI-based, and wrapper-based such as the Weka-ranker take chances to win out. It indicates that finding out independent attributes can really help to boost up classification accuracy.

Feature selection is to not only make learning process more efficient and scalable, especially for high dimensional large data sets, but also can improve classification accuracy as shown in the results reported above. Another benefit of feature selection is that it can save costs in reality. For example, table 5.9 shows descriptions for attributes in the diabetes data set. There are 8 testing items for diagnosing diabetes. Their respective costs are listed in table 5.10. It looks like testing levels of the glucose and the insulin cost a lot more than the others.

Table 5.11 compares selection methods with using full attributes for diabetes diagnosis. It shows that both the Weka's ranker and the mutual information-based selection methods

ID	Attribute detail
1	Number of times pregnant
2	Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3	Diastolic blood pressure (mm Hg)
4	Triceps skin fold thickness (mm)
5	2-Hour serum insulin (mu U/ml)
6	Body mass index (weight in kg/(height in m)^2)
7	Diabetes pedigree function
8	Age (years)

Table 5.9: Diabetes data attributes

Attribute ID	Test	Cost ($)
1	times_pregnant	1.00
2	glucose_tol	17.61
3	diastolic_pb	1.00
4	triceps	1.00
5	insulin	22.78
6	mass_index	1.00
7	pedigree	1.00
8	age	1.00

Table 5.10: Diabetes testing costs

Method	Size	Attributes	Classify rate	ClassifierID	Cost ($)	Saving
Full-attribute	100%	1 2 3 4 5 6 7 8	0.72	8	46.39	0%
Weka-ranker	37.5%	2 8 1	0.74	9	19.61	58%
MI-based	50%	4 8 2 7	0.72	8	20.61	56%
Entropy-based	62.5%	5 2 3 4 8	0.71	2	43.39	6%
RRQR	62.5%	5 2 3 4 8	0.71	2	43.39	6%

Table 5.11: Selection methods for cost saving in diabetes diagnosis

can not only achieve a same or better classification accuracy rate but also can save more than 50% of the total cost than using full attributes for diagnosis. It is remarkable because that the selection methods can promisingly improve current healthcare system. It can not only reduce patients' costs, such as for detecting diabetes, but also can help doctors to improve the diagnosis accuracy. This is just one of those examples to illustrate the advantage of feature selection in reality.

5.7 SUMMARY

In this chapter, we applied the mutual information-based selection method inspired by submodularity optimization, and the entropy-based selection method for feature selection in classification. We compared them with two other popular methods, RRQR and Weka's ranker, on multiple data sets with a variety of classifiers.

Through a batch of systematic experiments, we find out that selecting a subset of features work much better than using full attribute set on many classification problems. The selection method based on maximizing submodular mutual information poises itself to be a strong competitor among other selection methods, in terms of its comparable performance. Its selection achieving the maximal mutual information value leads to the best classification performance than by using other selections for the mutual information-based method, and also it could outperform other selection methods. Moreover, the performances of different

classifiers combined with different attribute selection methods largely depended on data sets. We also show an example based on the diabetes classification problem that how the selection methods could save medical costs, and in the meantime achieve a better accuracy rate of diabetes diagnosis in reality.

CHAPTER 6

CONTRIBUTIONS

This dissertation focuses on observation selection in wireless sensor networks, and its extended applications in feature selection for classification problems.

We presented our improved version of VoIDP algorithm for selecting optimal sensor observation on chain graphical models in chapter 2, and demonstrated its applications of wireless sensor scheduling. We also discussed a situation when assuming no reward penalties the computation of the optimal expected total reward for a sub chain can be further simplified.

In chapter 3, we solved the placement problem for traffic monitoring sensors by applying Gaussian process model-based observation selection methods in our simulated road traffic network map. We employed two greedy heuristics based on submodular mutual information and entropy under multivariate Gaussian. Our experimental results demonstrate the characteristics of these two different heuristics. It shows that the entropy-based heuristic places

sensors mainly around road intersections whereas the submodular mutual information-based heuristic disperses sensors widely across the road network. We also discover that the mutual information-based heuristic is better than the entropy-based in avoiding repeatedly selecting strong correlated locations on same road segments. However the mutual information-based heuristic only work well given the selection size is within about half of the total size, because the mutual information gains will decrease after that.

We also compared graphical model-based, particularly with Hidden Markov Models, and Gaussian process (GP) model-based selection approaches under the same scenario of wireless sensor scheduling in chapter 4. It gains us insight about these two model-based observation selection approaches. Our experimental results show that the Gaussian process-based approach performs better than the graphical model-based approach in terms of the predictive accuracy given correct observations. But when small errors were injected into the observations, the GP based selection method performs very poorly. In contrast, the graphical model based approach demonstrates more robust and stable performance given the erroneous observations, and outperforms the GP-based approach when more observations are selected.

Finally we employ sensor observation selection methods into another field of subset selection problems, feature selection for classification. In chapter 5, we apply the mutual information-based selection method exploiting submodularity optimization to filter out redundant features for classification problems. We compare the proposed method with existing

state-of-the-art attribute selection methods through extensive experiments with multiple classifiers and data sets, and show that the proposed mutual information-based feature selection method perform comparably with, or even better than, other feature selection methods.

CHAPTER 7

FUTURE RESEARCH AND APPLICATIONS

From observation selection in wireless sensor networks to feature selection for classification, the selection of the most important or valuable information plays a vital role. With restricted resources people always want to use less to do more. For example, when we design and deploy a remote sensing system to monitor an outdoor environment or an indoor manufacture factory, we'd like to maximize its capability and efficiency while still running at a low cost of monetary budge and energy consumption. Selection also means ranking or sorting things by its importance, which is the key for making decision. People have to make choices in their lives. An artificial intelligent machine needs to decide next action that maximizes its goal. A search algorithm needs to send back to users a selected and ranked list of information. For many things, selection plays an important and necessary component.

Selection of important information even becomes crucial to a new era of computing.

Today's computers need to handle large data sets and huge amount of information ranging from remote sensing, the Internet, bioinformatics, electronic health records, and to all kinds of digitized data. while traditional algorithms of machine learning and data mining scramble for sorting out and making sense of the large data sets, selection methods can be key components to build efficient models with high predictive accuracy. However, selection methods also have many challenges to deal with. We have seen that the performance of the selection methods largely depend on characteristics of data sets or specific problems. A selection method may work well on some data sets but not on others. The challenge brought by big data sets also applied to selection methods. To address these challenges, I look forward to two promising directions, one is to build customized selection methods, and the other is for parallel and distributed computing implementation of the customized algorithms.

We have learned that performance of attribute selection methods vary depending on data sets. The problem caused by the variation is amplified by large data sets in high dimensionality, which have both a large number of attributes and instances. It even becomes more challenging for selection methods to find out the most important attributes or instances. A promising approach is to customize the algorithms based on the nature of problems or characteristics of a data set. This involves a deeper understanding of a data set or a problem, and developing a suitable algorithm for it. Technically it may need dividing a large data set into a group of small data sets by either instances or attributes, and then find out a customized

learning algorithm that fits to each of the group set. The final solution will be computed by combining the sub solutions of each part. This divide-and-conquer approach is also algorithmic desirable and implementing scalable for processing large and high dimensional data sets.

Scalability and efficiency become more and more important requirements in the era of big data sets, and the parallelism and concurrency will be another key besides selection methods. Parallel and concurrent computing remains challenging nowadays. How to efficiently implement such a customized selection algorithm that we mentioned above, for a big data set with high dimensionality, in a distributed computing environment is still a challenging but also very interesting research and engineering topic.

In future, for computer science and engineering to large data sets, What I look forward to is an efficient and customized selection approach, and its parallel and distributed implementation.

Since I entered college for studying computer science in 1997, until now, based on my past 15 years of experience and observation in this field, It turns out that the Algorithms holds its position steadily compared with many other fast-paced changing technologies such as computer programming languages. The Algorithms is the hard core of Computer Science, and my passion for it has been built up over the time. Many science fields have been inspiring the advance of design and development of Computer Science Algorithms such as

mathematics and statistics. The trend has also been carried on by new technology in computer architecture and hardware such as multiple-cores and quantum computers. The design, development and analysis of Computer Science Algorithms in state of the Art is always my biggest interest in the field.

BIBLIOGRAPHY

[1] Christian H. Bischof and G. Quintana-Ortí. "Computing rank-revealing QR factorizations of dense matrices". In: *ACM Trans. Math. Softw.* 24.2 (June 1998), pp. 226–253. ISSN: 0098-3500. DOI: 10.1145/290200.287637. URL: http://doi.acm.org/10.1145/290200.287637.

[2] W. Caselton and J. Zidek. "Optimal monitoring network designs". In: *Statistics and Probability Letters* 2.4 (1984), pp. 223 –227.

[3] W.F. Caselton and J.V. Zidek. "Optimal monitoring network designs". In: *Statistics and Probability Letters* 2(4) (1984), pp. 223–227.

[4] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM Data: Classification, Regression, and Multi-label*. URL: http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

[5] Hinrich Schütze Christopher D. Manning Prabhakar Raghavan. *An Introduction to Information Retrieval*. Cambridge, England: Cambridge University Press, 2009.

[6] Sanmay Das. “Filters, Wrappers and a Boosting-Based Hybrid for Feature Selection”. In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML ’01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 74–81. ISBN: 1-55860-778-1. URL: http://dl.acm.org/citation.cfm?id=645530.658297.

[7] S. De Vito et al. “Wireless Sensor Networks for Distributed Chemical Sensing: Addressing Power Consumption Limits With On-Board Intelligence”. In: *Sensors Journal, IEEE* 11.4 (2011), pp. 947 –955. ISSN: 1530-437X. DOI: 10.1109/JSEN.2010.2077277.

[8] Amol Deshpande et al. “Model-driven data acquisition in sensor networks”. In: *VLDB ’04: Proceedings of the Thirtieth international conference on Very large data bases*. Toronto, Canada: VLDB Endowment, 2004, pp. 588–599. ISBN: 0-12-088469-0.

[9] Prabal Dutta. “Sustainable sensing for a smarter planet”. In: *XRDS* 17 (4 2011), pp. 14–20. ISSN: 1528-4972. DOI: http://doi.acm.org/10.1145/1961678.1961680. URL: http://doi.acm.org/10.1145/1961678.1961680.

[10] George Forman. “An Extensive Empirical Study of Feature Selection Metrics for Text Classification”. In: *Journal of Machine Learning Research* 3 (2003), pp. 1289–1305.

[11] A. Frank and A. Asuncion. *UCI Machine Learning Repository*. 2010. URL: http://archive.ics.uci.edu/ml.

[12] Pushkar Tripathi Gagan Goel Chinmay Karande and Lei Wang. "Approximability of Combinatorial Problems with Multi-agent Submodular Cost Functions". In: *FOCS '09: 50th Annual IEEE Symposium on Foundations of Computer Science*. Atlanta, GA, USA: IEEE, 2009.

[13] Edward I. George. "The Variable Selection Problem". In: *J. Amer. Statist. Assoc* 95 (1999), pp. 1304–1308.

[14] Michel X. Goemans et al. "Approximating submodular functions everywhere". In: *SODA '09: Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*. New York, New York: Society for Industrial and Applied Mathematics, 2009, pp. 535–544.

[15] Ming Gu and Stanley C. Eisenstat. "Efficient algorithms for computing a strong rank-revealing QR factorization". In: *SIAM J. Sci. Comput.* 17.4 (July 1996), pp. 848–869. ISSN: 1064-8275. DOI: 10.1137/0917055. URL: http://dx.doi.org/10.1137/0917055.

[16] Isabelle Guyon and André Elisseeff. "An Introduction to Variable and Feature Selection". In: *Journal of Machine Learning Research* 3 (2003), pp. 1157–1182.

[17] Mark Hall et al. “The WEKA data mining software: an update”. In: *SIGKDD Explor. Newsl.* 11.1 (Nov. 2009), pp. 10–18. ISSN: 1931-0145. DOI: 10.1145/1656274.1656278. URL: http://doi.acm.org/10.1145/1656274.1656278.

[18] Amine Haoui, Robert Kavaler, and Pravin Varaiya. “Wireless magnetic sensors for traffic surveillance”. In: *Transportation Research Part C: Emerging Technologies* 16.3 (2008). Emerging Commercial Technologies, pp. 294 –306. ISSN: 0968-090X. DOI: DOI:10.1016/j.trc.2007.10.004. URL: http://www.sciencedirect.com/science/article/B6VGJ-4R8M994-1/2/a9ad0bf584c67ed74328ae3b6e287980.

[19] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Second Edition. New York: Springer, 2009.

[20] S.R. Pope I.C.F. Ipsen C.T. Kelley. “Rank-Deficient Nonlinear Least Squares Problems and Subset Selection”. In: *SIAM J. Numer. Anal.* 49.3 (2011), pp. 1244–1266.

[21] Iñaki Inza et al. “Filter versus wrapper gene selection approaches in DNA microarray domains”. In: *Artificial Intelligence in Medicine* 31.2 (2004), pp. 91–103.

[22] Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. “A combinatorial strongly polynomial algorithm for minimizing submodular functions”. In: *J. ACM* 48.4 (2001),

pp. 761–777. ISSN: 0004-5411. DOI: http://doi.acm.org/10.1145/502090.502096.

[23] Michael I. Jordan. *Learning in Graphical Models*. Second Printing. Cambridge, Massachusetts: The MIT Press, 2001.

[24] Ron Kohavi and George H. John. "Wrappers for Feature Subset Selection". In: *Artif. Intell.* 97.1-2 (1997), pp. 273–324.

[25] Andreas Krause. "Optimizing sensing: theory and applications". PhD thesis. Carnegie Mellon University, 2008.

[26] Andreas Krause and Carlos Guestrin. "Near-optimal Nonmyopic Value of Information in Graphical Models". In: *Proceedings of the Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*. Arlington, Virginia: AUAI Press, 2005, pp. 324–331.

[27] Andreas Krause and Carlos Guestrin. "Near-optimal observation selection using submodular functions". In: *AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence*. Vancouver, British Columbia, Canada: AAAI Press, 2007, pp. 1650–1654. ISBN: 978-1-57735-323-2.

[28] Andreas Krause and Carlos Guestrin. "Nonmyopic active learning of Gaussian processes: an exploration-exploitation approach". In: *ICML '07: Proceedings of the 24th*

international conference on Machine learning. Corvalis, Oregon: ACM, 2007, pp. 449–456. ISBN: 978-1-59593-793-3. DOI: http://doi.acm.org/10.1145/1273496.1273553.

[29] Andreas Krause and Carlos Guestrin. "Optimal nonmyopic value of information in graphical models: efficient algorithms and theoretical limits". In: *IJCAI'05: Proceedings of the 19th international joint conference on Artificial intelligence*. Edinburgh, Scotland: Morgan Kaufmann Publishers Inc., 2005, pp. 1339–1345.

[30] Andreas Krause and Carlos Guestrin. "Optimal value of information in graphical models". In: *J. Artif. Int. Res.* 35.1 (2009), pp. 557–591.

[31] Andreas Krause and Carlos Guestrin. "Optimizing Sensing: From Water to the Web". In: *Computer* 42.8 (2009), pp. 38–45. ISSN: 0018-9162. DOI: http://dx.doi.org/10.1109/MC.2009.265.

[32] Andreas Krause, Ajit Singh, and Carlos Guestrin. "Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies". In: *J. Mach. Learn. Res.* 9 (2008), pp. 235–284. ISSN: 1532-4435.

[33] Andreas Krause, Ajit Singh, and Carlos Guestrin. "Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies". In: *J. Mach. Learn. Res.* 9 (2008), pp. 235–284.

[34] Andreas Krause et al. “Near-optimal sensor placements: maximizing information while minimizing communication cost”. In: *IPSN ’06: Proceedings of the 5th international conference on Information processing in sensor networks*. Nashville, Tennessee, USA: ACM, 2006, pp. 2–10. ISBN: 1-59593-334-4. DOI: http://doi.acm.org/10.1145/1127777.1127782.

[35] Andreas Krause et al. “Robust Submodular Observation Selection”. In: *J. Mach. Learn. Res.* 9 (2008), pp. 2761–2801.

[36] Andreas Krause et al. “Selecting Observations against Adversarial Objectives”. In: *Advances in Neural Information Processing Systems 20*. Ed. by J.C. Platt et al. Cambridge, MA: MIT Press, 2008, pp. 777–784.

[37] Wenke Lee, Salvatore J. Stolfo, and Kui W. Mok. “Adaptive Intrusion Detection: A Data Mining Approach”. In: *Artif. Intell. Rev.* 14 (6 2000), pp. 533–567. ISSN: 0269-2821. DOI: 10.1023/A:1006624031083. URL: http://dl.acm.org/citation.cfm?id=373612.373622.

[38] Jure Leskovec et al. “Cost-effective outbreak detection in networks”. In: *KDD ’07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. San Jose, California, USA: ACM, 2007, pp. 420–429. ISBN: 978-1-59593-609-7. DOI: http://doi.acm.org/10.1145/1281192.1281239.

[39] Huan Liu and Hiroshi Motoda. *Computational Methods of Feature Selection (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series)*. Chapman & Hall/CRC, 2007. ISBN: 1584888784.

[40] Huan Liu and Hiroshi Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Norwell, MA, USA: Kluwer Academic Publishers, 1998. ISBN: 079238198X.

[41] K. Penner (advisors: I.C.F. Ipsen M.E. Broadbent M. Brown and R. Rehman). "Subset Selection Algorithms: Randomized vs. Deterministic". In: *SIAM Undergraduate Research Online* 3 (May 2010), 22 pages.

[42] Alexandra Meliou et al. "Nonmyopic informative path planning in spatio-temporal models". In: *AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence*. Vancouver, British Columbia, Canada: AAAI Press, 2007, pp. 602–607. ISBN: 978-1-57735-323-2.

[43] Bilge Mutlu et al. "Robust, low-cost, non-intrusive sensing and recognition of seated postures". In: *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology*. Newport, Rhode Island, USA: ACM, 2007, pp. 149–158. ISBN: 978-1-59593-679-2. DOI: http://doi.acm.org/10.1145/1294211.1294237.

[44] G. Nemhauser, L. Wolsey, and M. Fisher. “An analysis of the approximations for maximizing submodular set functions”. In: *Mathematical Programming* 14 (1978), pp. 265–294.

[45] KianSing Ng and Huan Liu. “Customer Retention via Data Mining”. In: *Artif. Intell. Rev.* 14.6 (2000), pp. 569–590.

[46] Zhuang Peng et al. “Model-Based Traffic Prediction Using Sensor Networks”. In: *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*. 2008, pp. 136–140.

[47] Qi Qi and Yi Shang. *Comparing Probabilistic Graphical Model Based and Gaussian Process Based Selections for Predicting the Temporal Observations*. New York, NY: ASME Press, 2010. ISBN: 9780791859599. DOI: DOI:10.1115/1.859599.paper77. URL: http://link.aip.org/link/doi/10.1115/1.859599.paper77.

[48] Qi Qi and Yi Shang. “Submodular Mutual Information-Based Placement of Wireless Vehicle Detection Sensors for Traffic Signal Control”. In: *Transportation Letters: The International Journal of Transportation Research* (submitted, 2012).

[49] Qi Qi, Yi Shang, and Hongchi Shi. “An Improved Algorithm for Optimal Subset Selection in Chain Graphical Models”. In: *Proceedings of 2010 IEEE Congress on Evolutionary Computation*. Barcelona, Spain, 2010.

[50] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. Cambridge, Massachusetts: The MIT Press, 2006.

[51] M. C. Rodriguez-Sanchez, S. Borromeo, and J. A. Hernández-Tamames. "Wireless Sensor Networks for Conservation and Monitoring Cultural Assets". In: *Sensors Journal, IEEE* 11.6 (2011), pp. 1382 –1389. ISSN: 1530-437X. DOI: 10.1109/JSEN.2010.2093882.

[52] E. Sazonov et al. "Self-Powered Sensors for Monitoring of Highway Bridges". In: *Sensors Journal, IEEE* 9.11 (2009), pp. 1422 –1429. ISSN: 1530-437X. DOI: 10.1109/JSEN.2009.2019333.

[53] Alexander Schrijver. "A combinatorial algorithm minimizing submodular functions in strongly polynomial time". In: *J. Comb. Theory Ser. B* 80.2 (2000), pp. 346–355. ISSN: 0095-8956. DOI: http://dx.doi.org/10.1006/jctb.2000.1989.

[54] Amarjeet Singh et al. "Efficient planning of informative paths for multiple robots". In: *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*. Hyderabad, India: Morgan Kaufmann Publishers Inc., 2007, pp. 2204–2211.

[55] Daniel L. Swets and John J. Weng. "Efficient Content-Based Image Retrieval using Automatic Feature Selection". In: *Proceedings of the International Symposium on*

Computer Vision. Washington, DC, USA: IEEE Computer Society, 1995, pp. 85–90. ISBN: 0-8186-7190-4.

[56] Joel A. Tropp. "Column subset selection, matrix factorization, and eigenvalue optimization". In: *SODA*. Ed. by Claire Mathieu. SIAM, 2009, pp. 978–986.

[57] M. Tubaishat et al. "Wireless Sensor-Based Traffic Light Control". In: *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*. 2008, pp. 702 –706. DOI: 10.1109/ccnc08.2007.161.

[58] Vijay V. Vazirani. *Approximation Algorithms*. Second Printing. Germany: Springer, 2003.

[59] G. Vijay, E. Ben Ali Bdira, and M. Ibnkahla. "Cognition in Wireless Sensor Networks: A Perspective". In: *Sensors Journal, IEEE* 11.3 (2011), pp. 582 –592. ISSN: 1530-437X. DOI: 10.1109/JSEN.2010.2052033.

[60] Machine Learning Group at University of Waikato. *Weka 3: Data Mining Software in Java*. URL: http://www.cs.waikato.ac.nz/~ml/weka/index.html.

[61] M. Wiering et al. *Intelligent Traffic Light Control*. Technical Report UU-CS-2004-029. University Utrecht, 2004. URL: http://www.sf.net/projects/stoplicht.

[62] Pentaho community Wiki. *Pentaho Data Mining Community Documentation*. URL: http://wiki.pentaho.com/display/DATAMINING/Pentaho+Data+Mining+Community+Documentation.

[63] Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Third edition: Morgan Kaufmann, 2011.

[64] Lei Yu and Huan Liu. "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution". In: *ICML*. Ed. by Tom Fawcett and Nina Mishra. AAAI Press, 2003, pp. 856–863. ISBN: 1-57735-189-4.

[65] Binbin Zhou et al. "Adaptive Traffic Light Control in Wireless Sensor Network-Based Intelligent Transportation System". In: *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*. 2010, pp. 1 –5. DOI: 10.1109/VETECF.2010.5594435.

VITA

Qi Qi was born in August 21, 1978, in Beijing, People's Republic of China. After attending public schools in Beijing, he obtained his B.E. in Computer Software from Beijing University of Technology in 2001. After a few years of industrial experiences in Beijing, he joined the graduate program in Department of Computer Science at the University of Missouri in August 2005. He received his M.S. in Computer Science in December 2007, and earned his Ph.D. in Computer Science in May 2012. He has been driving his passion in the field of Computer Science for the past 15 year. He always dedicates himself to be a computer scientist, an engineer, and a teacher.