Intervention Recommendations to Minimize the 30-Day Risk-of-Readmission for Heart Failure

Rui Liu

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Computer Science and Systems

University of Washington

2015

Reading Committee:

Dr. Senjuti Basu Roy, Chair

Dr. Ankur Teredesai

David Hazel

Program Authorized to Offer Degree:
Institute of Technology - Tacoma

University of Washington

Abstract

Intervention Recommendations to Minimize the 30-Day Risk-of-Readmission for Heart Failure

Rui Liu

Chair of the Supervisory Committee:
Dr. Senjuti Basu Roy
Institute of Technology

In this thesis, we investigate the problem of designing personalized intervention strategies to minimize 30-day readmission risk for heart failure (HF) patients. In particular, we propose a novel framework that recommends personalized intervention to the patients by analyzing the complex interplay among a multitude of factors, such as, demographic factors, medical diagnoses, clinical factors, and how they contribute to readmission risk. Our proposed framework is flexible enough to include or exclude additional factors, as well as layers, or can even obey constraints provided by the domain experts (i.e., doctors) in the design of this hierarchical network. First, we propose to learn the structure and parameters of a hierarchical Bayesian network from the available patient data and use that to design rules to recommend personalized interventions. We propose scalable implementation of our proposed solution on Windows Azure, present comprehensive experimental results based on the proposed approach to demonstrate the effectiveness of our proposed framework using large scale high dimensional real patient dataset. Our work is validated with State Inpatient Data for Washington State and MultiCare Health System Data. Finally, we introduce a web-based interactive service *Pathway-Finder* to support decision making. While our primary effort is to design intervention strategies for HF, the proposed framework could be adapted in designing intervention strategies for other diseases as well.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am grateful to many people without whose help and contributions this work could not have been completed. First, I would like to express my gratitude to my advisor and committee chair Dr. Senjuti Basu Roy, who has been guiding my research and lending me courage to purse a PhD degree in the future. Next, I would like to appreciate Dr. Ankur Teredesai, who has given me valuable advice about my thesis. I also would like to thank other collaborators in Center of Data Science, David Hazel, Si-Chi Chin, Kiyana Zolfagar and Aftab Hassan for their support in this research.

Last but not least, I would like to express my deep appreciation to my family. Without support from my parents, I would never complete my Master's degree.

DEDICATION

To my parents

Chapter 1

INTRODUCTION

1.1 Background

Heart Failure (henceforth referred to as HF) is one of the leading causes of hospitalization, and studies[23, 29] reveal that many of these admissions are readmissions within a short window of time. Based on the 2005 data of Medicare beneficiaries, it has been estimated that 12.5% of Medicare admissions due to HF were followed by readmission within 15 days, accounting for about $590 million in healthcare costs [20]. Readmission can result from a variety of reasons, including early discharge of patients, improper discharge planning, and poor care transitions. In particular, studies[23, 29] have shown that targeted interventions during pre-discharge [30] and post discharge phases, like home based follow up [23], patient education [18], or administering appropriate procedures during the hospital stay can reduce the readmission rates considerably and improve the health outcome of the patients. Such factors that could be externally controlled (or administered) are construed as *interventions* and are applicable at different phases of a patient's life cycle.

1.2 Problem Definition

While research around predictive models for estimating the onset or risk of chronic conditions is increasing [34, 22, 35, 36], there is still a significant lack of tools that go beyond risk prediction to make clinical data actionable to accomplish healthier outcomes for patients. We, for the first time, attempt to go beyond risk prediction for HF, and focus on actionable intervention recommendation to aid clinicians in designing improved quality of care protocols to reduce the 30-day[1] readmission risk for HF. To recommend intervention strategies, we

---

[1] 30-day is considered *clinically meaningful* by different healthcare services and standards [1].

consider a multitude of factors, such as, socio-demographic factors, comorbidities[2] and other diagnoses, and procedures by understanding the complex interplay between these factors and how they contribute to the 30-day readmission risk. Furthermore, not all of these factors are “intervenable”. For example, socio-demographic factors such as age and gender may positively contribute to the *high* 30-day readmission risk, but these factors could not be intervened upon.

As an example, consider an elderly American African female patient with long-standing hypertension who has developed shortness of breath and significant dependent lower extremity swelling. She undergoes an echocardiogram and other outpatient testing. It is determined that she has evidence of diastolic heart failure and a treatment plan is initiated. Unfortunately, her symptoms continue to aggravate and she is admitted to the local hospital for further intervention. However, an alternate care pathway might have included the patient being referred to a heart failure specialty clinic after the initial diagnosis of diastolic heart failure was made. An alternate treatment plan was then started and stabilized the symptoms, thereby avoiding a hospital admission. We intend to investigate all these alternatives and recommend that care pathway which is most likely to minimize her readmission risk. Therefore, the problem we address is: Make personalized intervention recommendations to minimize 30-day HF readmission risk.

1.3 Approach

A Bayesian network is a probabilistic graphical model that represents a set of random variables and their dependencies using a directed acyclic graph [16]. Our proposed framework is designed by formalizing the intervention recommendation as a structure learning problem [24, 33], where the objective is to *learn the structure of a hierarchical Bayesian network* involving a multitude of factors and how they *contribute* to 30-day readmission risk. Each factor (i.e., such as, social-demographic, diagnoses, procedures, readmission) in our settings

---

[2] Comorbidities are specific patient conditions that are secondary to the patient’s principal diagnosis and that require treatment during the stay.

contributes to a node in the network, and the *causal relationship* between two nodes is represented as a weighted (weight represents probability) directed edge, giving rise to a Directed Acyclic Graph (DAG in short). The learned Bayesian network structure also reveals the causality among a sample structure may look like the one presented in Figure 1.1 for the simple case described in Example 1 in Chapter 2. The underpinning of our proposed framework relies on the following four steps: (1) Since we deal with very high dimensional data involving several hundreds of factors, we first attempt to *learn the structure of the network automatically* from the data itself. For structure learning, our designed solution appropriately adapts Constraint Based Bayesian network Learning algorithm [28, 8], *Score Based Learning Algorithm* [26] and *Hybrid Algorithm* [33] that combines both Constrained based and Score based approaches. (2) Once the structure is defined, we use parameter learning [14, 5] techniques to compute the probability of the directed edges. (3) Third, we propose novel algorithm to generate a set of intervention rules. (4) Finally, for a given patient the generated rules are summarized to offer personalized intervention recommendations to the clinician who can then review and share them with the patient.

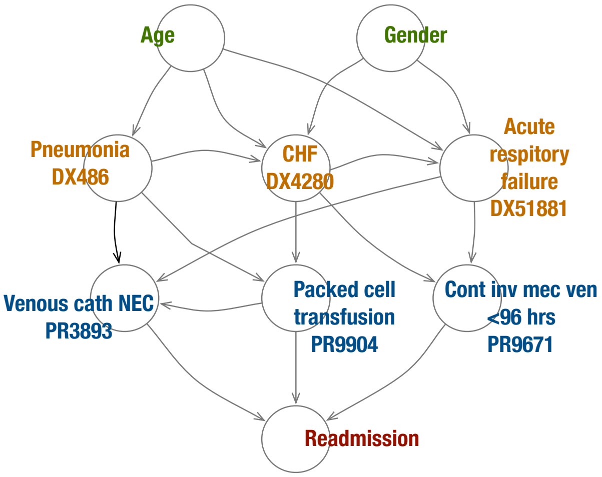


Figure 1.1: A learned structure for Example 1.

In our design, we furthermore enable the flexibility to *specify* the constraints that are known apriori or provided by the cardiac team. For example, if physicians indicate that there *can not* be any causal relationship between *ethnicity* and *length of hospital stay*, we need to learn the network while satisfying this constraint. Our structure learning algorithms are capable to handle such constraints, can easily incorporate additional factors, or even layers. After fitting the parameters to this learned structure, a non-trivial challenge is to generate recommendation rules. Bayesian networks are traditionally used for inference learning, whereas, we adapt the model to generate recommendation rules.

We present comprehensive experimental results to validate the effectiveness of our proposed method on Microsoft Azure for Research cloud platform. We use the State Inpatient Dataset (SID-WA data in short) [3] of Washington State that comprises more than $200,000$ real patient records and 200 attributes spanning across 2 years as primary study to validate our intervention recommendation framework. We also provide comparative study with MultiCare Health System Data to understand the influence of constraints provided by domain expert so as to support our validation. While our primary effort in this paper is limited to recommending interventions for HF patients, our proposed framework is generic and could be easily adapted for our diseases as well. To the best of our knowledge, ours is one of the first attempts that investigates the intervention recommendation for HF in a principled manner.

Furthermore, we develop *Pathway-Finder*, a novel interactive recommender system for clinical decision support. Through clinical pathways analysis, we identify and gather what is known about the patient (through EMR records and intake form) as well as recommend appropriate interventions that can lead to improved care quality, and report on the efficacy of those interventions. The proposed interactive system surfaces patient information relevant to that encounter, as well as supporting identification of new factors, which can then be visualized to show the connections between patient demographic characteristic, disease conditions(comorbidities or diagnoses), possible interventions, and targeted clinical outcomes.

---

[3] `http://www.hcup-us.ahrq.gov/sidoverview.jsp`

1.4 Contribution

The contributions of our work are:

- We initiate the study for recommending personalized interventions to minimize 30-day HF readmission risk.

- We formalize the intervention recommendation task as a hierarchical Bayesian Structure Learning problem. Furthermore, we propose multiple algorithms as solutions that are flexible enough to incorporate the constraints provided by the domain experts or the users. We present novel algorithms to generate and summarize rules for personalized intervention recommendation.

- We present comprehensive experimental results on Microsoft Azure, as well as case studies to demonstrate the effectiveness of our proposed techniques.

- *Pathway-Finder* visualizes the trace and predicted outcome of a patient, supporting personalized intervention recommendation.

The rest of thesis is organized as follows: Chapter 2 include architecture and technical details of our approach; Chapter 3 contains experiment settings and comprehensive evaluation of the proposed framework; Chapter 4 describe the web-based intervention recommendation service; Chapter 5 contains related work to our topic and we conclude in Chapter 6.

Chapter 2

INTERVENTION RECOMMENDATION FRAMEWORK

We describe our proposed personalized intervention recommendation framework in this chapter. A hierarchical Bayesian network effectively depicts the causal relationships between the factors and how their interplay relates to lowering the heart failure readmission risk. Thus, we model the intervention recommendation as a network learning task using the Bayesian network learning principles. After that, we perform parameter learning to compute the conditional probabilities between factors. Once we have the underlying structure and the conditional probabilities, we develop a novel algorithm to generate recommendation rules based on the fitted network learned from the given dataset. These three steps constitute the training part of the proposed framework. During testing (recommendation evaluation), we summarize the generated rules and validate the effectiveness of recommendation. Figure 2.1 describes the high level design of our intervention recommendation pipeline.

The running example is presented next to recommend interventions during hospitalization, but we note that the framework could be easily adapted to any other phase.

Example 1. Let us consider a set of two socio-demographic factors : age, and gender; three diagnoses (Congestive Heart Failure (CHF) DX4280, Acute Respiratory Failure (ARF) DX51881, and Pneumonia (PN) DX486); and three procedures (Continuous Invasive Mechanical Ventilation $<$ 96 hrs PR9671, Venous Cath NEC PR3893, Packed Cell Transfusion PR9904)[1]. These eight factors are predictors and we wish to learn how they relate to the likelihood of a 30-day HF readmission. Diagnoses (y/n), Procedures (flattened to y/n), and gender (m/f) are treated as binary variables, whereas age is a continuous variable which has

---

[1] Each diagnoses procedure has a unique code written after its name and these procedures are applicable during hospitalization.

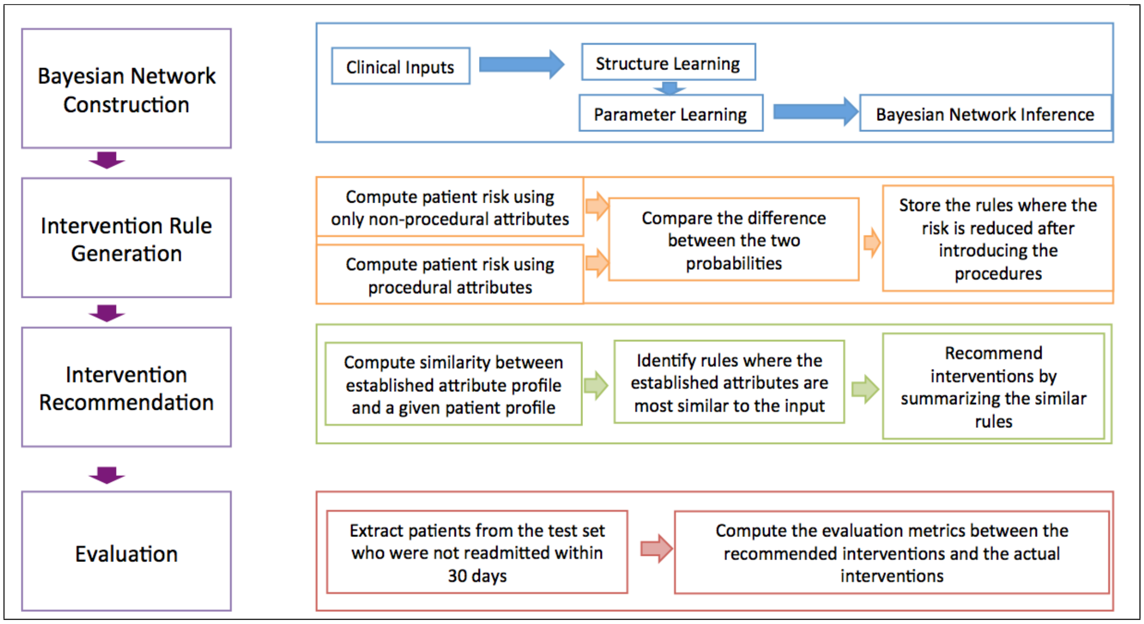


Figure 2.1: Intervention Recommendation Framework Pipeline

been discretized appropriately. Finally, the dependent variable “readmission” is a binary variable, where “Readmission=0” stands for 30 day readmission unlikely, and “Readmission=1” stands for highly-likely. For simplicity, we consider only the procedures that are intervenable. Thus, the objective is to consider the evidence about age, gender, diagnosis and possible procedures to recommend appropriate procedures that will likely minimize the overall readmission risk.

Physicians often have a series of protocol steps they follow about a patient. Hence it is safe to assume that the underlying framework needs to include additional constraints. Lets say that our physician collaborator specifies that a causal relationship exists between diagnoses and demographic factors; but not among the demographic factors themselves. Physician also specifies that a causal relationship exists between diagnosed conditions and procedures and internally between diagnosed conditions, and among the procedures administered. Finally, she indicates that procedures have a strong causal relationship with readmission risk.

As the Figure 2.1 depicts, our intervention recommendation pipeline is organized with four modules, namely, *Bayesian network Construction*, *Intervention Rule Generation*, *Intervention Recommendation* and *Evaluation*. First, we construct a Bayesian network to model the clinical data provided by health care beneficiaries. Then, we use the constructed Bayesian network to generate intervention rules base on the data provided and build a rule repository to store the generated rules. For the incoming patients, we compare the his non-intervenable features with the rules in the repository and make recommendation for the patients. Finally, we evaluate our intervention recommendation framework using the actual interventions applied on the patients.

2.1 Bayesian network Construction

Relevant notations are presented in Table 2.1. A Bayesian network is a probabilistic graphical model that represents a set of random variables and their dependencies using a directed acyclic graph [16]. It consists of the following parts:

- A set of variables and a set of directed edges between variables.
- Each variable has a finite set of mutually exclusive states.
- The variables(nodes) and directed edges form a directed acyclic graph (DAG).
- To each variable X with parents Pa^X, there is a conditional probability distribution associated with node X over nodes Pa^X. Thus, the joint probability distribution of the domain could be represented by Equation 2.1.

$$Pr(X_1, X_2, \ldots, X_N) = \Pi_{i=1}^{i=n} Pr(X_i | Pa^{X_i}) \tag{2.1}$$

In our case, the variables, which are the nodes in the DAG, are clinical attributes from State Inpatient Data or MultiCare Health System Data. Specifically, these clinical data are all discrete attributes and their respective conditional probability distributions are represented as multinomial distributions.

Table 2.1: Notations and Interpretations

Notation	**Interpretation**
$\mathcal{D}$	the dataset
N	number of points in the dataset, i.e., —$\mathcal{D}$—
X, Y, Z	three variables
x, y, z	values of X, Y, Z, respectively
P_a^X	a set of nodes that are parent of X
$\mathcal{X}$	the entire set of predictor variables, or attributes, or factors.
α_{ijk}	the hyperparameters of the Bayesian network

We perform a structure learning on the data provided by the health care beneficiaries. Here we are using three structure learning algorithms, *Constraint Based Structure Learning Algorithm*, *Score Based Structure Learning Algorithm* and *Hybrid Structure Learning Algorithm.* More detail about the structure learning algorithms will be expanded in the following subsection. Once the structure of the Bayesian network is determined, we perform the parameter learning algorithm to learn the parameters which are conditional probabilities of the nodes in the learned Bayesian network. At the end of this module, we will get a completely learned Bayesian network.

2.1.1 Structure Learning

Our structure learning solution relies upon the *Causal Sufficiency Assumption* and the *Markov Assumption*[31]. We use *Constraint Based, Score-Based, and Hybrid* methods to learn the structure of the network. We describe them one by one next.

Constraint Based Methods

Constraint Based Methods make use of the conditional independence tests using statistical tests on the data set. The conditional independence tests that are used in practice are statistical tests on the data set. Following assumptions have to be made if we are applying constraint based methods:

Causal Sufficiency Assumption There exist no common unobserved (also known as hidden or latent) variables in the domain that are parent of one or more observed variables of the domain.

Markov Assumption Given a Bayesian network model B, any variable is independent of all its non-descendants in B, given its parents.

Faithfulness Assumption A BN graph $\mathcal{G}$ and a probability distribution $\mathcal{P}$ are faithful to one another iff every one and all independence relations valid in $\mathcal{P}$ are those entailed by the Markov assumption on $\mathcal{G}$.

Regarding that we are constructing a hierarchical Bayesian network without any hidden variables, all the assumptions are satisfied.

We use a computationally efficient algorithm, *Grow and Shrink* [6] which relies on detecting the *Markov Blanket* [26] of the variables to induce the network structure. Markov blanket for a node X in a Bayesian network is the set of nodes composed of X's parents, its children, and its children's other parents. It operates by identifying the local neighborhood of each variable in the Bayesian network as a preprocessing step, in order to facilitate the recovery of the exact structure around each variable in subsequent steps. This has the added advantage of being easier to verify and possibly correct a *posteriori* by an outside expert. The overall algorithm runs in two phases : in the *grow phase*, given a node X, it starts with an empty set S and it adds variables to S as long as they are dependent (i.e., Markov Blanket property is violated) with X given the current contents of S. In this process, however, there may be some variables that were added to S that were really outside the blanket. In the *shrinking phase*, those variables are identified and removed.

Score-Based Methods

Constraint-based algorithms suffer from poor “robustness”, i.e., a large effects on the output of the algorithm is observed, for small changes of the input i.e. single errors in the independence tests. To overcome that shortcoming, we apply Score-based approaches that uses hill climbing heuristics.

Score-Based Methods creates several Bayesian network and assigns a score to each candidate of them, typically one that measures how well that Bayesian network describes the data set $\mathcal{D}$. Assuming a structure $\mathcal{R}$, its score is,

$$Score(\mathcal{R}, \mathcal{D}) = Pr(\mathcal{R}|\mathcal{D}) = \frac{Pr(\mathcal{D}|\mathcal{R}) \times Pr(\mathcal{R})}{Pr(\mathcal{D})} \tag{2.2}$$

To calculate $Pr(\mathcal{R}|\mathcal{D})$, the Bayesian approach averages over all possible parameters, weighing each by their posterior probability as Equation 2.3.

$$Pr(\mathcal{R}|\mathcal{D}) = \int Pr(\mathcal{D}|\mathcal{R}, \mathbf{p}) Pr(\mathbf{p}|\mathcal{R}) d\mathbf{p} \tag{2.3}$$

Notice that all of attributes presenting in our data are discrete and modeled using multinomial distribution, the probability distribution function is described in Equation 2.4.

$$Pr(\mathcal{R}|\mathcal{D}) = \Pi^{n}_{i=1} \Pi^{q_{ij}}_{j=1} \frac{\Gamma\alpha_{ij}}{\Gamma(\alpha_{ij} + N_{ij})} Pi^{r_i}_{k=1} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma\alpha_{ijk}} \tag{2.4}$$

We use Bayesian Information Criterion (BIC) to approximate $Pr(\mathcal{R}|\mathcal{D})$ for each Bayesian network candidate. Equation 2.5 describe how the BIC is calculated.

$$BICScore(\mathcal{R}, \mathcal{D}) = \log Pr(\mathcal{D}|\hat{\mathbf{p}}, \mathcal{R}) - \frac{d}{2} \log N \tag{2.5}$$

where $\hat{\mathbf{p}}$ denotes the the set of maximum-likelihood estimates of the parameters of the Bayesian network.

As score-based algorithms attempt to maximize this posterior probability, returning the structure $\mathcal{R}$ that maximizes it is prohibitively expensive. Since the search space of all possible structures is exponential to the number of variables n, this posses tremendous computational challenges. Our solution performs a local search and does not need the brute-force computations. The search is started from either an empty, full, or possibly random network. After that, it iteratively attempts to add, remove, or reverse every possible single-edge and consider that change which increases the score most. The process stops when there is no single-edge change that increases the score.

Hybrid Approach

We finally apply a hybrid approach to learn the network structure, namely the *max-min hill climbing algorithm* [33]. The *max-min hill climbing algorithm* combines ideas from the both Score-Based approach and Constraint-Based Approach in a principled and effective way. *MMHC* first learns the skeleton of a Bayesian network using a local discovery algorithm called *Max-Min Parents and Children*. The skeleton of a Bayesian network means it only learns edges in the network without identifying the orientation of the edges. Then the algorithm orients the skeleton using a greedy Bayesian-scoring hill-climbing search. As MMHC is based on MMPC, which is an efficient and theoretically sound local learning algorithm, it is proven to be sound in the sample limit. Notice that the first phase require a constraints based algorithm, the data should satisfy the assumptions stated in the Section 2.1.1. This algorithm has several advantages. First of all, it only requires one-dimensional tables to maintain the current list of edges considered. Once the skeleton has been identified, the graph of the network is stored as adjacency lists with very low memory requirements for sparse networks. This make the *MMHC* outperform those algorithms requiring two-dimensional tables and simultaneously considering all the possible edges in the network. A second advantage of the local approach method is that it can easily be modified to selectively reconstruct only a part of the network, when time constraints or increased complexity in a local area of the network forbids global reconstruction. Thus, this algorithm appears effective in many high dimensional real world problem (such as ours) and tackles the limitations posed by the other algorithms.

The resultant network structure generated by these algorithms satisfy any constraint that is already specified, it will include the edges that are pre-specified to exist, and vice-versa.

2.1.2 Parameter Learning

After the structure of the network is constructed, the next step is to learn the parameters of the network, given the structure. Using Example 1, this step is analogous to creating probability distribution functions, specifically to each node in the constructed network to create the conditional probability table at each node. As an example, using the sample network of Figure 1.1, this step will compute all the following probabilities at node PR 9671.

$$\begin{aligned} &\text{Pr(PR 9671= i— DX 4280 =0 \& DX 51881=0)}\\ &\text{Pr(PR 9671= i— DX 4280 =0 \& DX 51881=1)}\\ &\text{Pr(PR 9671= i— DX 4280 =1 \& DX 51881=0)}\\ &\text{Pr(PR 9671= i— DX 4280 =1 \& DX 51881=1)}\\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall_{i=0,1} \end{aligned} \tag{2.6}$$

Typically, for parameter learning, a prior distribution is assumed over the parameters of the local pdfs before the data is used (for example, this can be uniform), or it could be estimated using the given data itself. The distribution of a node X conditional upon its parents may have any form. The *conjugacy* of this prior distribution is also desirable, meaning that the posterior belongs to the same family as the prior, albeit with different parameters.

In our implementation, we use Bayesian Parameter Estimation [19] to learn the parameter θ. In this method, the prior distribution over θ (i.e., $Pr(\theta)$) is known. Now the posterior distribution of θ is calculated according to Bayes rule:

$$P(\theta|D) = \frac{Pr(D|\theta)Pr(\theta)}{\int Pr(D|\theta)Pr(\theta)d\theta} \tag{2.7}$$

Our objective is to calculate the Maximum A Posteriori (MAP in short), i.e.,

$$\hat{\theta}_{MAP} = argmax_{\theta}P(\theta|D) = argmax_{\theta}Pr(D|\theta)Pr(\theta) \tag{2.8}$$

The prior $Pr(\theta)$ is calculated using a Beta-Distribution for binary variables which gives rise to a posterior which is also a Beta distribution.

$$Pr(\theta) = Beta(\theta|\alpha_1, \alpha_0) = c\theta^{\alpha_1 - 1}(1-\theta)^{\alpha_0 - 1} \tag{2.9}$$

Whereas, for multi-valued (i.e., non-binary) discrete variables, prior $Pr(\theta)$ is a Dirichlet distribution with $Dir(\theta|\alpha)$ with hyper-parameters α_i's. The posterior would also be a Dirichlet distribution and will have the following form.

$$P(\theta|D) = cPr(D|\theta)Pr(\theta) \tag{2.10}$$

The constructed network may not be *complete* always, i.e., it may not consist of all possible edges between the nodes in two successive layers. Consider Example 1 again, and note that the variable "Gender" is not connected to all diagnosis nodes. At the same time, it is unrealistic to force the network to have all the edges, given its high dimensionality, because the search space increases exponentially by addition of edges between the nodes. Additionally, it may not be possible to track back one entire inference path of the network, because the network is not complete between two successive levels.

2.2 Recommendation Rule Generation

Then we make use of the constructed Bayesian network to generate a set of recommendation rules. In this module, our target is to select the interventions which are proven to be effective from the entire training dataset. A effective intervention should reduce the patients' readmission risk and prevent readmissions.

The key idea is to the Bayesian network inference. Using the constructed network after parameter learning, for each patient record d, we could compute the probability $Pr(Readmit = 1|d)$ and $Pr(Readmit = 0|d)$. We describe next how to make use of these inference probabilities to generate a set of recommendation rules.

Without loss of generality, let us assume that a total of $|\mathcal{X}'|$ of $|\mathcal{X}|$ factors are non-intervenable, and the remaining set $\{\mathcal{X}\} - \{\mathcal{X}'\}$ of factors could be recommended as interventions.

For each patient record d whose actual class label is 0 (i.e.,$Readmission = 0$), we use only $|\mathcal{X}'|$ attributes of record d (denoted as $d(\mathcal{X}')$) and feed it through the constructed network to obtain the inference probability p_1. Then, we use the entire patient record (with both intervenable and non-intervenable attributes, modulo the class label), and use that to make a second inference probability p_2.

$$p_1 = Pr(Readmission = 0|d(\mathcal{X}'), p_2 = Pr(Readmission = 0|d) \tag{2.11}$$

If $p_2 > p_1$ (which indicates that our constructed model infers that the set of procedures associated with the patient input is effective in further bringing down her readmission risk), we store the set of procedures $\{\mathcal{X}\}-\{\mathcal{X}'\}$ associated with d as the generated recommendation, given the values for the non-procedure attributes.Using Example 1, a recommendation rule in our case may look as follows:

- Rule-1: if Gender = Female & Age = 64 & diagnosis= PN & diagnosis= ARF & Readmit=0, recommended interventions (i.e, procedures) P1 (PR3893) = 1 & P2 (PR9904) = 0 & P3 (PR9671)= 1.

Similarly, for each patient record d' whose actual class label is 1 (i.e.,$Readmission = 1$), we check if the following condition is satisfied.

$$p_2 = Pr(Readmission = 1|d) < p_1 = Pr(Readmission = 1|d(\mathcal{X}') \tag{2.12}$$

- Rule-2: if Gender = Male & Age = 87 & diagnosis= CHF & diagnosis= ARF & Readmit=1, recommended interventions (i.e, procedures) P1 (PR3893) = 1 & P2 (PR9904) = 0 & P3 (PR9671)= 1.

In that case, Rule-2 will be also stored. Given Rule-1 and Rule-2, we select the historical health care records that will reduce the readmission risk after introducing the effective interventions. At the end of this module, we dump the rules to a rule repository for the future recommendation use.

2.3 Intervention Recommendation

The next phase of the proposed intervention recommendation framework is to recommend interventions for the incoming patients (test cases) using the rule repository generated in Section 2.2. In this module, we compute the similarity between the incoming patient with the profiles stored in the rule repository, select those who are similar to the incoming patient and summarize their intervention plans to make recommendation.

2.3.1 k Nearest Neighbor Search

For each incoming patient, given his non-intervenable features, we search a set of similar k records in rules repository using *k-Nearest-Neighbor* (*k-NN* for short) search [12]. *k-NN* is a type of instance-based learning. It finds out the most similar set of nodes comparing the target node and then summarize using majority voting. In our case, we use *Jaccard Index* to evaluate the similarity between the target patient and rules in the rule repository. *Jaccard Index* (also known as Jaccard similarity coefficients), which is defined by Equation 2.13, is a evaluation metric measures the similarity between finite sample sets.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.13}$$

Map Reduce Like Implementation

At the first run of *k-NN*, we found it is too time consuming in a sequential configuration. Consider we have $|R|$ rules in our rule repository and $|I|$ incoming patients. If each patient need s seconds to make recommendation, then we will need $|R| \times |I| \times s$ seconds in total to response. To make our recommender system more efficient, we come up with a *Map-Reduce Like Implementation* to faster the process. We observe that the rules are individually independent to each other and so are the incoming test cases. Figure 2.2 illustrates the Map-Reduce like implementation. First, we partitioned the rule repository into L_R parts. This makes it possible to parallelize the searching process. For each in the test set, we break it down to L_P map tasks. In each map task, we look for top k nearest neighbors using *Jaccard Index*. Then, in the reduce phase, all the $L_P \times k$ outputs are combined together and output the top k nearest tuples. Considering our workload is not increased while we parallelize the work into L_P subset, it is to demonstrate that the overall consuming time will be reduced to approximately as $\frac{1}{L_P}$ of the original processing time.

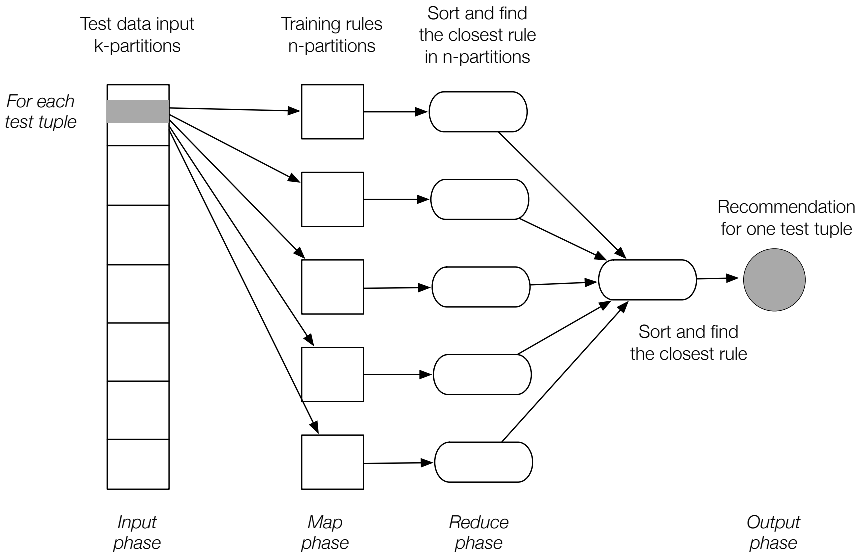


Figure 2.2: Map-Reduce Like Implementation

2.3.2 Summarization

After obtaining a set of similar rules (k nearest neighbors), we summarize the set of rules to make our recommendation. Our summarization module is using majority voting algorithm to summarize the rules. For each intervenable attribute presents in our data, we investigate the k rules, call a majority vote among the k voters and then choose the one with most votes. For example, considering procedure PR9904 when $k = 5$, if 3 out of 5 rules suggest it should be applied in Example 1, that procedure will be suggested as a intervention for the patient.

In this chapter, we describe how we construct our Bayesian network, how to obtain a rule repository utilizing the constructed Bayesian network and a parallel implementation to do the recommendation. Then we evaluate our framework in the next chapter.

Chapter 3

EXPERIMENT

In this chapter, we describe the baseline algorithms we will use to provide evaluation for our intervention recommendation framework. Then we introduce the evaluation methods to use in evaluation phase. We use two datasets to provide comprehensive experiments results in order to demonstrate effectiveness of our proposed framework.

3.1 Dataset Preparation

In this section, we present introduction to the two datasets. Notice that the two datasets are organized in two significantly different format: State Inpatient Databases for Washington State (SID-WA) dataset is in ASCII format while MultiCare Healthcare System (MHS) dataset is stored in Microsoft SQL data management tool. Then we present the preparation process for the two datasets.

3.1.1 State Inpatient Database

State Inpatient Databases (referred as SID-WA for the rest of the paper) are part of the family of databases developed for the Healthcare Cost and Utilization Project (HCUP) [1]. The dataset is a discharge abstract that includes inpatient discharge records from community hospitals in the State of Washington with all-payer, encounter-level information beginning in 2010 and 2011. SID-WA contains readmissions that occur at any hospital within the state of Washington. The interventions include health service utilizations and procedures in SID-WA. Health service utilizations present as flags and procedures present in ICD-9

---

[1] http://www.hcup-us.ahrq.gov/overview.jsp

codes. ICD-9 (The International Classification of Disease version 9) is a widely recognized international system for recording diagnoses.

Database Description

We use the State Inpatient Databases (SID) [2] of Washington State of year 2010 and 2011. SID-WA of one year comprises four files that are associated with patients and their encounters in the hospitals. The four files – core file (CORE), charges file (CHGS), diagnosis and procedure groups file (DXPRGRPS), and disease severity measures file (SEVERITY) – provide 596 attributes in total for a single patient encounter. Each inpatient encounter has an unique identifier KEY, that can be used to link records across files. Our initial dataset used KEY to join CORE, DXPRGRPS, and SEVERITY files and selected only attributes that are relevant to the clinical aspect of a patient encounter to construct heart failure cohort (described in Section 3.1.1) used for the experiments.

We use the attribute VisitLink in CORE to identify the same patients in the data. The attribute DaysToEvent is used to compute the the days between two consecutive hospital admissions for each patient. The days since previous hospital discharge is computed using the days between two hospital admissions minus the length of stay of the first admission. We exclude admission records in which the patient died in the hospital and in which the patient is younger than 1 year (age=0).

Heart Failure Cohort

We construct a heart failure cohort based on the initial dataset extracted from SID-WA, as described in the previous section. The cohort contains patients whose primary or secondary ICD9-CM diagnosis codes are listed in [22]. Initially, the cohort contains 3,908 distinct diagnosis codes and 2,049 procedure codes. In order to resolve the issue of sparsity and high dimensionality of the data, we perform *chi-square* feature selection to filter attributes that are

---

[2] http://www.hcup-us.ahrq.gov/sidoverview.jsp

less influential. Table 3.1 summarizes the 209 attributes used in the cohort. Unless otherwise stated, all 70 procedures and the attribute length of stay are construed as interventions.

Table 3.1: Attribute summary for SID heart failure cohort

Group	Category	Description
1	Demographics	Age, Gender, Race (3 attributes)
2	Comorbidity and Diagnosis	21 comorbidity and 90 diagnosis
3	Utilization and Procedures	21 health service utilization flags and 70 procedures
4	Others	Length of stay, # diagnosis, # procedures, # chronic conditions (4 attributes)

The final heart failure cohort contains data extracted from SID-WA 2010 and SID-WA 2011. Our experiments used the 2010 data (67967 patients) for training and the 2011 data (52021 patients) for testing.

3.1.2 MultiCare Healthcare System Database

MultiCare Health System dataset is provided by our collaborators in MultiCare Healthcare System which is the biggest healthcare service provider in South Sound, WA.

Database Description

The Cardiovascular data mart, which is our primary interest was developed in 2009 to support an internal clinical process improvement initiative. It is a robust analytics environment, holding approximately 8,600 patients diagnosed with HF and servicing over 14,200 hospital encounters from 2009 to 2013. In MHS dataset, we identify the medications using a MedicationID assigned by the system and identify the procedures using CPT code, which is widely used among the insurance companies.

Heart Failure Cohort

The dataset consists of HF hospitalization for patients discharged since 2009. We consider only patients who are discharged to home, excluding inter-hospital transfers. Admissions encountering in-hospital deaths are not included in our analysis. The final dataset includes 20476 diagnosed with HF from 2009 to 2013. After further exploration of the dataset we identify a total of 100 attributes to be related to CHF admission.

The key socio-demographic factors related to patients are, gender, race, marital status. Some of the other important factors pertinent to HF are ejection fraction value (EFV), blood pressure, primary and secondary diagnosis, other comorbidity variables, APR-DRG code. for severity of illness and APR-DRG code for risk of mortality. Information about the discharge disposition of patients such as the discharge status, discharge destination, length of stay and follow-up plans are also found to be correlated to HF readmissions. In addition, we include 35 cardiovascular and comorbidity attributes.

The primary challenge of working with MHS data arises due to prevalence of noise and missing values in the dataset. Our initial investigation shows that ejection fraction has the highest number of missing values (63%), followed by APR-DRG code for severity of illness (13.3%) and blood pressure (12.6%). We imputed the missing values as discussed in below Section. Our final dataset consists of 20476 instances.

Table 3.2: Attribute summary for MHS heart failure cohort

`Group`	`Category`	`Description`
1	Demographics	Age, Gender, Race .etc (15 attributes)
2	Comorbidity and Diagnosis	35 comorbidities
3	Interventions	50 procedures, 50 medications
4	Others	Length of say

3.2 Implemented Algorithms

We implement 3 different structure learning algorithms Hill Climbing (HC), Grow-Shrink (GS), Hybrid (HY) and compare it with 4 baselines to demonstrate the effectiveness of our proposed intervention recommendation framework. The 3 structure learning algorithms are explained in Section 2.1.

3.2.1 Baselines

We implement four baseline algorithms: two simple baselines, one *k-NN* baseline and one logistic regression substitution. In this section, we will expand the implementation of 4 baselines.

Simple Baselines

Considering we are initiating the intervention recommendation algorithms for HF patients to reduce their readmission risk, we use a simple baseline here which simply suggests all the procedures to be 0. In the rest of the thesis, we will refer to this simple baseline as *BL-1*. On the other hand, we also implement another baseline without using any learning phase, referred to as *BL-2* in the rest part of the thesis. *BL-2* consider the entire patient population and suggest the top-3 most frequent procedure.

k-NN Baseline

In our proposed framework, we facilitate the Bayesian network to discover effective rules out of the training dataset. The k-NN baseline consider the entire training dataset as the rule repository instead of any selection. Then it uses the map-reduce like implementation to search the k nearest neighbors and then summarize to recommend. This baseline is used to demonstrate we the rules discovered by the Bayesian network are effective.

Logistic Regression Substitution

Logistic Regression Substitution is our primary baseline. In this baseline algorithm, we substitute the *Bayseian Netowrk Construction* using logistic regression construction while the other modules of the intervention framework, namely *Intervention Rule Generation*, *Intervention Recommendation* remain the same.

Logistic regression, which is also know as logit regression is one of the most popular statistical classification models[3]. It is widely in many fields including medical and social sciences used to predict a categorical dependent value based a set of predictors. Instead of studying the dependency relationships among the mass of data as Bayesian network does, it treats all the predictors as independent variables. In logistic regression, the logit of the dependent variable is modeled using a linear combination of the predictor variables as Equation 3.1[10]. The relevant notations and their interpretations are presented in Table 3.3.

$$g(p) = \ln(\frac{p}{1-p}) = \vec{\beta}\vec{X} \tag{3.1}$$

Table 3.3: Notations for Logistic Regression

Notations	**Interpretations**
g	Logit function
p	Probability of the dependent variable
$\vec{X}$	The predictor variable vector
$\vec{\beta}$	The regression coefficients vector

In our case, we are using a multiple logistic regression because we have more than one predictor variables. Specifically, the dependent variable is “Readmit or not”, which indicates our logistic regression is binomial while the predictor variables include demographic, comorbidities, diagnosis and interventions. The regression coefficients are estimated using maximum likelihood[15]. An iterative process is used to optimize the likelihood function.

This process begin with a tentative solution. In each iterative step, the coefficients vector revises slightly and see whether the outcome will be improved. Repeat the process until converge. Once the coefficients are learned using this process, the probability of the readmission could be interpreted as Equation 3.2.

$$p = \frac{1}{1 + e^{-\vec{\beta}\vec{X}}} \tag{3.2}$$

3.2.2 Experiment Configuration

Our analyses are conducted on Microsoft Azure using 8 cores and each core with 56GB of RAM. The development and experimentation environment uses R-studio and Python. All numbers presented in the following sections are the average of three runs. We conduct our experiments primarily on SID-WA dataset and use MHS dataset as a comparative study.

3.3 Evaluation Methods

Evaluation is the final module of our proposed intervention framework as Figure 2.1 depicts. It demonstrates our proposed framework is effective and efficient. In this module, we extract the successful cases from the a dataset, which should be different from our training set, as testing set. At the very last step, we use four evaluation metrics to evaluate our recommendation framework.

3.3.1 Evaluation Data Extraction

It is necessary to emphasis that we only use the successful cases to evaluate our intervention recommendation framework. In unsuccessful cases, the intervention plan is not proper for the specific patient and it is impossible to determine a proper intervention plan. Considering this, we can not make use of the unsuccessful cases to evaluate our intervention recommendation framework, so we omit the unsuccessful cases in the evaluation phase. After the testing set is built, we compute the evaluation measures described in Section 3.3.2 to present effectiveness of our proposed framework.

3.3.2 Evaluation Measures

We use four evaluation metrics to validate our recommendation framework, namely *Hits*, *Average Accuracy*, *Average Jaccard Index* and *Average True Positive Rate(TPR).* For each pair of a set of recommended interventions and corresponding actual interventions, we could use a confusion matrix to analyze the recommendation performance. We define true positive (short as *tp*) case, which indicates the *recommended* interventions *appear* in the actual intervention list; true negative (short as *tn*) to represent the *non-recommended* interventions *do not appear* in the actual intervention list; false positive (short as fp) to represent the *recommended* interventions *do not appear* in the actual intervention list; false negative (short as fn) to represent the *non-recommended* interventions *appear* in the actual intervention list. We use Hit, *Accuracy*, $JaccardIndex$ and $TruePositiveRate$ (short as TPR), defined by Equation 3.3 to evaluate each test case. Then we summarize the metrics which are applied on each test case using sum or average.

$$\begin{aligned} JaccardIndex &= \frac{tp}{tp + fp + fn} \\ Accuracy &= \frac{tp + tn}{tp + tn + fp + fn} \\ TPR &= \frac{tp}{tp + fn} \end{aligned} \tag{3.3}$$

Hits

Hit indicate a exact match between the actual interventions and the recommended interventions. $Hits$ is computed by summing all the Hit in the entire data set.

Average Accuracy

Accuracy between the actual interventions and the recommended interventions is computed for each case. It is summarized using average function. *Averageaccuracy* is defined by Equation 3.4.

$$\begin{aligned} AverageAccuracy &= Average(Accuracy) \\ &= Average(\frac{tp + tn}{tp + tn + fp + fn}) \end{aligned} \tag{3.4}$$

Average Jaccard Index

Jaccard Index between the actual interventions and the recommended interventions is computed for each case. Then it is summarized using average. $AverageJaccardIndex$ is defined by Equation 3.5.

$$\begin{aligned} AverageJaccardIndex &= Average(JaccardIndex) \\ &= Average(\frac{tp}{tp + fp + fn}) \end{aligned} \tag{3.5}$$

Average True Positive Rate (TPR)

TPR between the actual interventions and the recommended interventions is computed for each case. Then it is summarized using average. TPR is defined by Equation 3.5.

$$\begin{aligned} AverageTruePositiveRate &= Average(TPR) \\ &= Average(\frac{tp}{tp + fn}) \end{aligned} \tag{3.6}$$

3.4 Results

We conduct our experiments on two datasets: SID-WA dataset and MHS dataset. We expand the experiments results on these two datasets in this section.

3.4.1 Primary Study on SID dataset

We implement 3 different structure learning algorithms (Hill Climbing (HC), Grow-Shrink (GS), Hybrid (HY) and compare it with a baseline implementation using Logistic Regression (LR) technique. We vary the number of diagnosis attributes (30, 60, 90). The LR

Table 3.4: Attribute Layers

`Layer`	Description
Base Layer	Demographic attributes and other attributes.
Diagnosis Layer	Diagnosis attributes and comorbidity attributes.
Intervention Layer	All the intervenable attributes. (In the SID-WA dataset, we have health service utilization and procedures. In MHS dataset, we have medications and procedures.)
Class-label Layer	"Readmission", which is our class label.

algorithm does not "discover" any causal relationship between the variables, but "learns" the association between different interventions and readmission risk. Based on that, it ranks the interventions and returns them. We note that other ranking based methods, such as, Odds Ratio [12] does not lend itself naturally to our problem settings. We also implement two natural baselines for summarization module (referred to Section 3.2.1).

Learning Phase

Bayesian networks place each variable (e.g. demographics, diagnosis, and procedure) in a directed acyclic graph. In a constructed Bayesian network, nodes represent variable nodes and edges represent causal relationships among them. The structure learning and parameter fitting is performed using the R statistical software environment with the *bnlearn* package. The training phase is akin to what is described in Chapter 2.

In order to ensure constraints provided by the domain experts and reveal the causality relationship of the clinical attributes, we partitioned the attributes into $4 layers$ as Table 3.4. They are demographic layer, diagnosis layer, intervention layer and readmission layer. Each layer contains a set of attributes in Table 3.1. Then we incorporate the constraints provided by the domain experts. Our implementation enables us to specify such constraints by creating a white-list and a black-list. Any black-listed edge must not be present in the network, whereas, the white-list ones must be present in the constructed network. In order to observe the relationship between procedures(i.e, interventions) and readmission, we white-list edges from influential procedure nodes (based on the results of feature selection) to readmission node. We use the following heuristics to construct the blacklist:

- Prohibit any out-links from Class-label layer.
- Prohibit any in-links to Base layer.
- Prohibit any links from Intervention layer to Diagnosis layer.

After we apply the constraints from domain experts, we obtain 12657 edges in the black-list and 5 edges in the white-list.

The network structure is learned on Microsoft Azure using 10,000 records randomly sampled from the training data and the parameters are using all 68K records. We generate the recommendation rules, as described in Section using the complete SID-WA 2010 heart failure cohort.

Table 3.5 and Table 3.6 shows the complexity of the constructed networks and the number of distinct rules discovered for each experiment setting. From Table 3.5 we could observe that *HC* always obtain Bayesian network with highest complexity while the complexity of *GS* and *HY* are much lower. Combining the two tables, we find their is no dependency relationship between the number of edges in constructed network and the number of rules discovered. *30*, *60* and *90* denote the number of diagnosis used to construct the Bayesian network. As we could get from the table, a higher dimensional data always get a more complex network under our configuration.

Validation Phase

After a Bayesian network is learned in the previous section, the next phase is to validate our recommendation using complete SID-WA 2011 heart failure cohort using 52K records.

Table 3.5: SID-WA:Number of edges in the constructed network

	`Num of diagnosis`		
`Alg`	`30`	`60`	`90`
HC	277	306	341
GS	63	67	75
HY	107	108	119

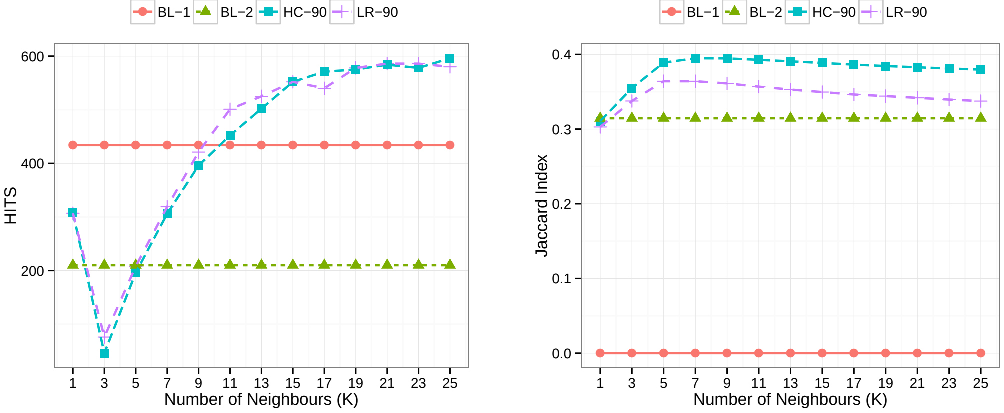
Table 3.6: SID-WA:Number of distinct rules discovered in each experiment setting

	`Num of diagnosis`		
`Alg`	`30`	`60`	`90`
HC	58,864	37,028	51,479
GS	46,501	46,501	46,531
HY	42,465	24,322	24,322

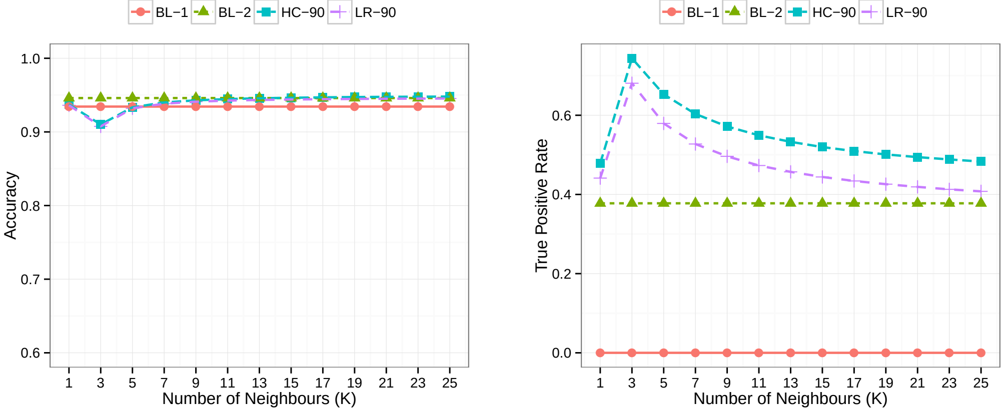
We compute metrics in Section 3.3.2 to evaluate our proposed intervention recommendation framework.

Effectiveness of Summarization Summarization is one of the module in our proposed framework. It is explained in Section 2.3. We compare our intervention recommendation framework using implementation of *Score Based Structure Learning Algorithm* with other baselines in Section 3.2.1 to evaluate the effectiveness of summarization. We vary the k, which is the number of neighbors in $k-NN$ algorithm then compute the metrics in Section 3.3.2. Figure 3.1 include the experiments results of our proposed implementation and baseline implementations. As the figure suggest, the recommendation quality is improved only to certain extent. It also implies that a small number of similar rules (a small k) is sufficient for effective intervention recommendation.

Recommendation Effectiveness Then we evaluate the recommendation effectiveness in this paragraph. Figure 3.2 present our experiment results for recommendation effectiveness at $k = 7$. In this set of experiments, we vary the structure learning algorithms and the number of diagnosis used from the SID-WA data. We perform a paired t-test to understand the significance of the results in Table 3.7. We set the significance level to $p - value < 0.05$. In Figure 3.2, x-axis varies the number of diagnoses attributes and y-axis captures the average of a respective quality measure. In Table 3.7, * denotes that the result is significantly better than the others among the three variations of attribute numbers. † denotes whether the result is significantly better than the others among the three structure learning algorithms and logistic regressions. First of all, we could observe that algorithms with 90 diagnosis significantly outperform those with 30 or 60 diagnosis. Second, we could observe that *Hill-Climbing* which is a *Score Based Structure Learning Algorithm* is significantly better than others in terms of *Jaccard Index* while *Max Min Hill Climbing* which is a *Hybrid Structure Learning Algorithm* performs better in *True Positive Rate.* We could also observe that Bayesian network Implemented Algorithms are significantly better than the Logistic Regression in terms



(a) Hits (b) Jaccard Index



(c) Accuracy (d) True Positive Rate

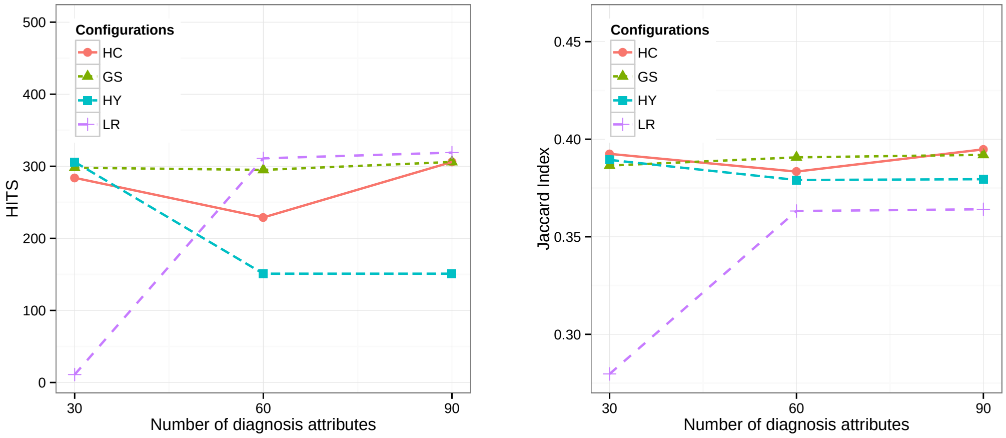
Figure 3.1: Effectiveness of summarization module; x-axis varies k and y-axis captures the average of a respective quality measure.

Table 3.7: Statistical significance of quality results of Figure 3 is further explained using paired-t-test.

`Alg`		`HITS`	`JAC`	`ACCY`	`TPR`
`HC`	30	284	0.3925†	0.9394	0.5960
	60	229	0.3834†	0.9372	0.5983
	90	306	0.3948*†	0.9400*†	0.590*
`GS`	30	298	0.3865	0.9370	0.5979
	60	295	0.3907	0.9380	0.5995
	90	306	0.3920*	0.9383*†	0.6002*
`HY`	30	306	0.3894*	0.9444*†	0.6020†
	60	151	0.3791	0.9344	0.6145†
	90	151	0.3794	0.9344	0.6164*†
`LR`	30	11	0.2797	0.9152	0.5149
	60	311	0.3632	0.9380	0.528*4
	90	319	0.3641*	0.9384*	0.5274

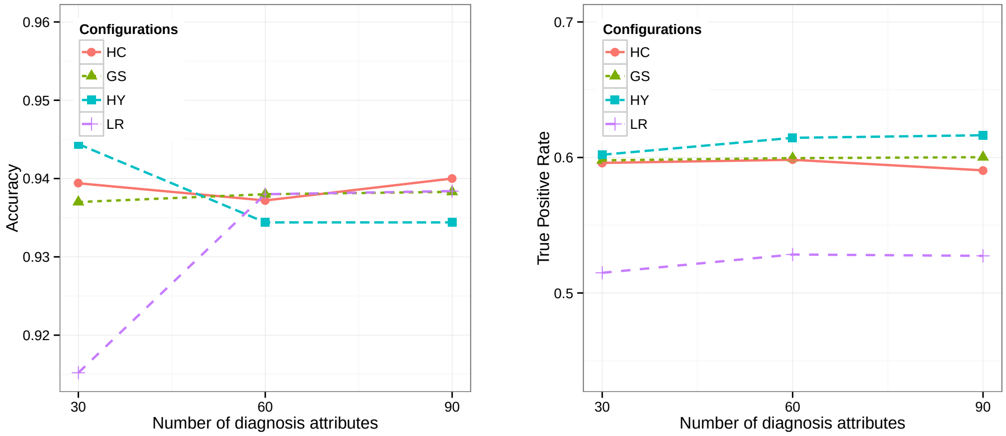
of *Jaccard Index*, *Accuracy* and *True Positive Rate*. None of the algorithms are doing well in terms of *Hits* which means a total match (all of them get around 300 out of around 50,000 test cases).

Figure 3.3 shows our experiments results at $k = 1$. In this set of experiments, besides varying the structure learning algorithms and the number of diagnosis used from the SID-WA data, we also compare our recommendation results with the $k - NN$ baseline and logistic regression substitution. In the figure, x-axis varies the number of diagnoses attributes and y-axis captures the average of a respective quality measure. We could observe that all the algorithms provide comparable outcomes. HY algorithm suffers from lower *Hits*, $JACC$ and ACC comparing to other four algorithms. This means rule repository constructed by this



(a) Hits

(b) Jaccard Index



(c) Accuracy

(d) True Positive Rate

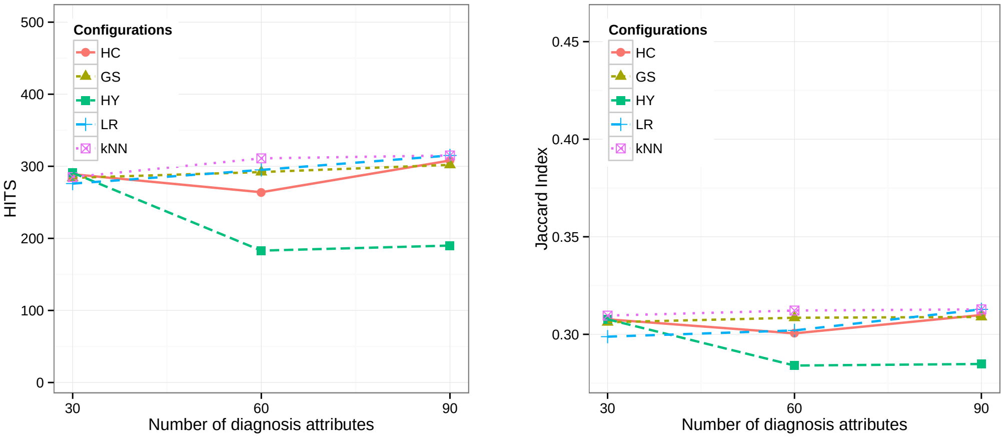
Figure 3.2: Effectiveness of different algorithms for intervention recommendation ($k = 7$)

algorithm contains less effective rules than others. Meanwhile, kNN and *HC* outperform other algorithms significantly and provide very comparable results. *HC*'s rule repository (contains 44,852 rules while kNN's rule repository contains the 51,084 rules. This result suggests the *HC* framework remove the ineffective rules from its repository while kNN accepts all.

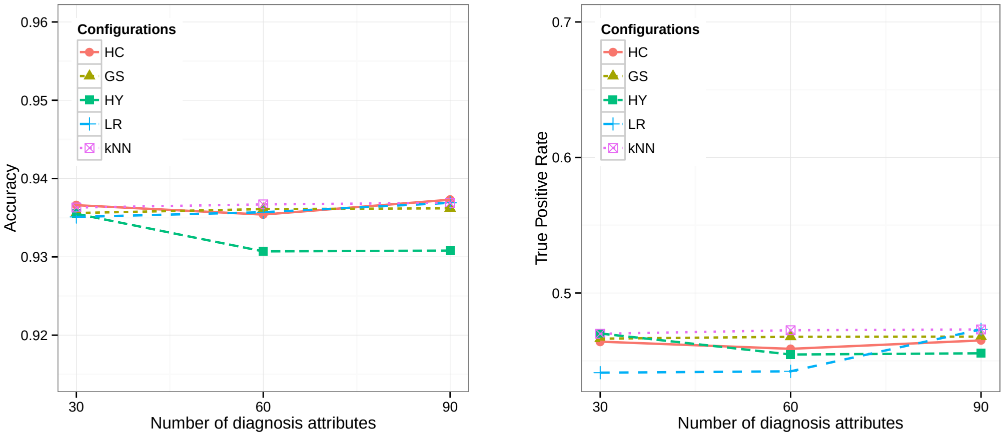
Effectiveness of Prior Knowledge In this paragraph, we conduct experiments with or without whitelist to evaluate the influence by prior knowledge provided by domain experts. We compare the performance of *Grow Shrink* which is a *Constraint Based Structure Learning Algorithm* and *Max Min Hill Climbing* algorithm in the setting of with/without whitelist provided by domain experts. All of the experiments are under the setting of 30 diagnosis. Considering that the *Logistic Regression* does not assume any causality exist among the attributes, we don't include *LR* in this set of experiments. In Table 3.8, the *ori* means the original algorithm without any domain knowledge while the *alt* refer to the altered algorithm with whitelist provided. The whitelist includes a number of edges we presume their existence in the *Bayesian network*. We could observe from the results that altered algorithms outperforms in TPR under the setting of *HY*. In terms of *Hits*, *Jaccard Index* and *Accuracy*, we observe that the original algorithms are presenting better results in both structure learning algorithms.

3.4.2 Comparative Study on MHS dataset

We also conduct comparative study to demonstrate that our proposed framework is generic and it could be applied to different systems. In our comparative study on MHS dataset, we implement three structure learning algorithms. As we have already written in Section 3.4.1, we partition the attributes into 4 layers as Table 3.4. The MHS data is partitioned into 2 equal parts randomly. One for learning the structure and parameter of the Bayesian network while the other for validating. In order to evaluate the influence from the blacklist, we conduct experiment of 3 algorithms with or without blacklist constraints. MHS dataset



(a) Hits (b) Jaccard Index



(c) Accuracy (d) True Positive Rate

Figure 3.3: Effectiveness of different algorithms for intervention recommendation ($k = 1$)

Table 3.8: Recommendation evaluation for the alternative whitelist. * denotes the value that is significantly higher

`Alg`		`HITS`	`JAC`	`ACCY`	`TPR`
`GS`	ori	284	0.3067*	0.9355*	0.4796
	alt	217	0.3044	0.9307	0.4876*
`HY`	ori	291	0.3082*	0.9355*	0.4834
	alt	217	0.3031	0.9341	0.4840

provide more clinical attributes than WA-SID does. In addition to procedures, MHS also provide medication order data for each patient, which also plays an important role contributing to minimize the risk of readmission for patients. Although the Map-Reduce Like implementation mitigate the scalability issue, it remains impossible if we are recommending the procedures as well as medications at the same time. The computation space requirement will grow significantly, thus prohibiting us to achieve the two recommendation objectives at the same time. Instead of recommending two fields at the same time, we implement recommendation for medications in a parallel manner. Rather than constructing one Bayesian network, we construct two Bayesian networks: one network to address the recommendation for procedures, the other for medications.

Procedure Experiments Results & Analysis

Table 3.9 show the comparative experiments results for procedure recommendation on MHS dataset. In Table 3.9, the *non-black* means the Bayesian network is constructed without blacklist while *black* means the Bayesian network is constructed utilizing blacklist. The structure learning algorithm will prohibit the links in blacklist if provided. If our proposed framework, the blacklist ensures our constructed Bayesian networks are organized in a hierarchical manner as we have already described in Section 3.4.1. First thing we could observe

Table 3.9: MHS:Procedure Experiments Results

`Alg`		`Number of edges`	`Number of rules`	`JACC`	`ACC`	`TPR`
HC	non-black	169	1338	0.7987	0.8091	0.9035
	black	162	1186	0.6993	0.7185	0.9145
GS	non-black	42	2009	0.7994	0.8097	0.9309
	black	42	2170	0.8084	0.8177	0.9017
HY	non-black	70	2233	0.7908	0.8019	0.9047
	black	68	2027	0.7903	0.8012	0.9054

is that we achieve comprehensive high *Jaccard Index*, *Accuracy* and *True Positive Rate* as we do on SID-WA dataset. We could also observe that the number of edges in learned Bayesian network will change when we add or remove blacklist. In the case of *HC* or *HY*, when we force constraints to the Bayesian network it will be smaller (with fewer edges) and the performance will drop especially in *HC*. It's not the case in *GS* where the Bayesian network will grow bigger and provide better performance in terms of *Jaccard Index* and *Accuracy*. Above all, the better we understand the Bayesian network structure of the data, the better performance the framework will provide.

Medication Experiments Results & Analysis

Table 3.9 shows statistics evaluating medication recommendation. In Table 3.10, as in the previous section the *non-black* means the Bayesian network is constructed without blacklist while *black* means the Bayesian network is constructed utilizing blacklist. Comprehensively, our recommendation framework obtains TPR greater than 96% in every algorithms and with or without blacklist. However, we could also see that all of three algorithms are suffering low JACC and ACC. The reason for low JACC and ACC is that some medications have several alias. For example, *Bumex* is a medicine treating fluid retention (edema) and high blood

Table 3.10: MHS:Medication Experiments Results

`Alg`		`Number of edges`	`Number of rules`	`JACC`	`ACC`	`TPR`
HC	non-black	112	2204	0.272	0.402	0.9822
	black	111	7448	0.242	0.378	0.9845
GS	non-black	28	877	0.402	0.509	0.9635
	black	27	534	0.418	0.517	0.9628
HY	non-black	60	932	0.359	0.474	0.9740
	black	56	534	0.418	0.517	0.9628

pressure. In the database, we have 3 *Bumex* 0.5 mg/1 mg and 2 mg. Our framework treat the three as three different medicine because they are attached with different IDs, which is the only way we identify the medications. However, things are different considering procedures, because they don't have any alias. This cause a great drop of performance in medication recommendation phase. This is considered to be a limitation of our framework.

In this chapter, we evaluate the effectiveness of summarization, prior knowledge using SID-WA dataset. Then we conduct comparative study on MHS dataset. Both datasets show our proposed framework performing well regarding the Hits, JACC, ACC and TPR. The experiments results demonstrate our proposed intervention recommendation framework to be effective to minimize 30-day readmission risk for HF patients.

Chapter 4

PATHWAY-FINDER IMPLEMENTATION

In this chapter, we propose a web-based interactive recommender system, called *Pathway-Finder*. The recommender system efficiently collects and displays patient information in a meaningful way to support an effective personalized treatment plan. *Pathway-Finder* implements a Bayesian network to discover casual relationships among factors and a *Key-Value* structure to traverse the network efficiently.

4.1 System Overview

Pathway-Finder is a cloud based web-service hosted on Microsoft Azure for Research platform. The objective is to interactively discover more about the user health conditions and adaptively recommend care-pathways to minimize 30-day readmission risk for heart failure patients. The majority of the proposed system components in this demonstration are pre-computed and stored to increase the speed of the application. *Pathway-Finder* implements a Bayesian network to discover casual relationships among factors and a *Key-Value* structure to traverse the network efficiently. Figure 4.1 provides the overview of the system that comprises of offline and online layer. The UI enlists simple socio-demographic factors and the user (i.e. clinician) selects respective values for those from the drop-down. After that, the system alternatively suggests a set of diagnoses and interventions to the user and then she/he selects some of them. In Section 4.2 we describe the offline computations, and Section 4.3 is used to describe the computations that take place, once the user starts interacting with the system.

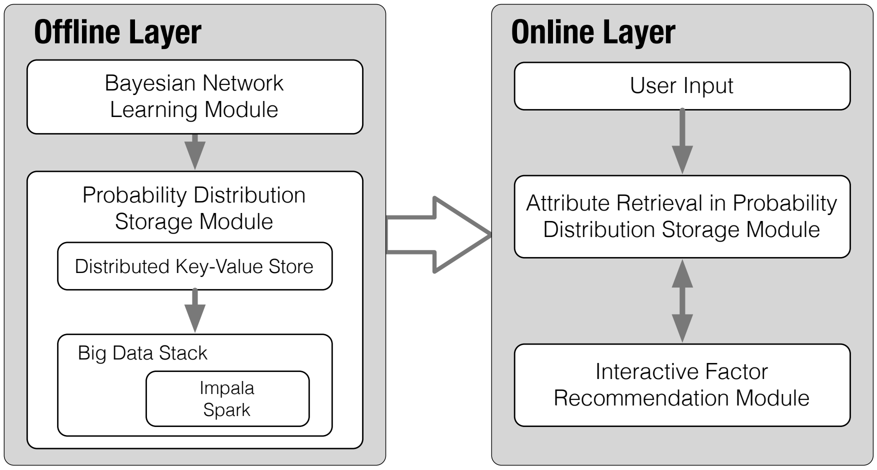


Figure 4.1: Pathway-Finder System Architecture

4.2 Offline Layer

The offline layer contains two modules, Bayesian network Learning Module and Probability Distribution Storage Module. In the offline layer, we input clinical data and output *Key-Value* storage.

4.2.1 Bayesian network Learning Module

In this module, we model the clinical data from a health care system as a Bayesian network in this module. As we have described in Section 2.1, we use structure learning algorithms to get a structure of the Bayesian network. Then, we perform parameter learning to learn the probability distribution function inside the Bayesian network.

4.2.2 Probability Distribution Storage Module

In this module, we transform the conditional probability function to a scalable and efficient storage. The keys are multiple set of composite keys consisting of the minimum number of combination of the attributes required to maintain the uniqueness and have a cascading relationship with each other. The values are the probability of diagnoses, given key and the rest of the attributes.

4.3 Online Layer

Online layer is an interactive layer between user and our recommendation system.

4.3.1 User Input

The interface accepts social-demographic inputs from users.

4.3.2 Probability Distribution Lookup Module

In this module, the system suggest a set of diagnosis and interventions based on the inputs. The module may be invoked multiple times to do look up either for the diagnoses or for the interventions. Based on the user input, the search goes inside the Probability Distribution Lookup Module to retrieve either the diagnoses or the interventions that the user is most likely to have.

4.3.3 Interactive Exploration/Recommendation Module

There is an iterative interaction between this module and the lookup module. If the user alter the suggested diagnosis, the output recommendation and readmission risk score will change correspondingly.

Chapter 5

RELATED WORK

To the best of our knowledge, no prior work has investigated the intervention recommendation problem for heart failure. In this section, we compare and contrast existing healthcare research efforts to enable decision support using data mining techniques at scale.

Bayesian network for decision support in healthcare settings: Several recent research efforts have applied Bayesian network to enable decision support in a clinical and healthcare settings. For example, a recent work has studied the problem of deciding a treatment plan for dental cares based on inter-causal association between different sign-symptoms using Bayesian network [2]. The work considers a rather low dimensional data. Similar modeling effort has been observed to enable decision support for generating treatment plan for other diseases as well, such as, coronary diseases [4], ulcers [9], sepsis [11], and depression [17].

Unlike us, none of these work deal with the problem of high dimensionality, scale, or multi-layer modeling. At the same time, for them, deciding the appropriate treatment plan for a given patient is simple and could be obtained by calculating the inference probabilities. Unlike our problem, they do not generate recommendation rules, nor do they perform large scale validation.

Risk prediction: A related important problem studied in several research effort is to predict risk [22]. For example, [32] proposes predictive modeling techniques to predict suicide risk. Several effective solutions (as well as distributed implementation) to predict the risk of readmission for congestive heart failure [34, 22, 35, 36] is known. However, none of these solutions has any easy extension for intervention recommendation problem.

Big data mining in healthcare settings: A recent work [7] has leveraged “big” healthcare claims data for the knowledge discovery process and demonstrated how the task of analyzing healthcare data could be translated into some of the most well-known analysis problems in the data mining community, social network analysis, text mining, temporal analysis, higher order feature construction, etc. Although our studied problem is fundamentally different; nevertheless, our proposed framework could benefit from the proposed large scale data analysis solutions.

A recent research [13] studies the problem of identifying risk signals of potential adverse drug reactions (ADRs) through Bayesian network. However, unlike us, their main contribution is in proposing solutions for large scale data mining. In particular, they propose an empirical Bayes modeling to mine ADR signals from 5 million adverse event reports collected by FDA and 46 million patient records.

Another recent work proposes a PARAllel predictive MOdeling (PARAMO) platform [25] which, (1) constructs a dependency graph of tasks from specifications of predictive modeling pipelines, (2)schedules the tasks in a topological ordering of the graph, and (3) executes those tasks in parallel using cluster computing environment.

It is easy to observe that our effort is orthogonal to these related work. Nevertheless, we wish to explore more advanced large scale data mining techniques in the future.

Chapter 6

CONCLUSION

In this work, we investigate the problem of recommending interventions to minimize 30-day risk of readmission for heart failure patients. Our proposed solution relies on learning the structure and parameters of a hierarchical Bayesian network from the given data to capture the complex interplay between multitude of factors related to heart failure, such as, demographic, diagnoses, and procedures and how they contribute to the 30-day heart failure readmission problem. After that, we propose novel algorithms to generate rules and summarize them that could be used to recommend interventions. Our implementation addresses the scalability and high dimensionality issues through parallel implementation on Windows Azure and our experimental results. We use comprehensive experiments to evaluate our proposed framework on SID-WA dataset and MultiCare dataset.

Furthermore, we propose an interactive system called *Pathway-Finder*, with the objective to visually explore, discover, and recommend clinical pathways for health conditions. We demonstrate *Pathway-Finder* to interactively recommend interventions to minimize the readmission risk due to Heart Failure (HF). We use a Bayesian network to model the clinical data and then use a key-value storage to enables us to perform real time lookup to interactively recommend interventions.

In this thesis, we design a framework to minimize the 30-day readmission risk for heart failure patients. However, minimizing the length-of-stay (also know as LOS), mortality risk and hospitalization costs remain open and challenging. In order to achieve better health care service, these problems must be addressed. Besides heart failure, we are also investigating other diseases, such as COPD and diabetes, to construct intervention recommendation framework to minimize readmission risk for patients in those cohorts.

BIBLIOGRAPHY

[1] Ruben Amarasingham, Billy J Moore, Ying P Tabak, Mark H Drazner, Christopher A Clark, Song Zhang, W Gary Reed, Timothy S Swanson, Ying Ma, and Ethan A Halm. An automated model to identify heart failure patients at risk for 30-day readmission or death using electronic medical record data. *Medical care*, 48(11):981–988, 2010.

[2] Ajay Bhatia and Rajeshwer Singh. Using bayesian network as decision making system tool for deciding treatment plan for dental caries. *Journal of Academia and Industrial Research (JAIR)*, 2(2):93, 2013.

[3] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

[4] John A Bittl, Yulei He, Alice K Jacobs, Clyde W Yancy, and Sharon-Lise T Normand. Bayesian methods affirm the use of percutaneous coronary intervention to improve survival in patients with unprotected left main coronary artery disease. *Circulation*, pages CIRCULATIONAHA–112, 2013.

[5] George EP Box and George C Tiao. *Bayesian inference in statistical analysis*, volume 40. John Wiley & Sons, 2011.

[6] Facundo Bromberg, Dimitris Margaritis, and Vasant Honavar. Efficient markov network structure discovery using independence tests. *Journal of Artificial Intelligence Research*, pages 449–484, 2009.

[7] Varun Chandola, Sreenivas R Sukumar, and Jack C Schryver. Knowledge discovery from massive healthcare claims data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1312–1320. ACM, 2013.

[8] Jie Cheng and Russell Greiner. Comparing bayesian network classifiers. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 101–108. Morgan Kaufmann Publishers Inc., 1999.

[9] Insook Cho, Ihnsook Park, Eunman Kim, Eunjoon Lee, and David W Bates. Using ehr data to predict hospital-acquired pressure ulcers: A prospective study of a bayesian network model. *International journal of medical informatics*, 82(11):1059–1067, 2013.

[10] David A Freedman. *Statistical models: theory and practice.* cambridge university press, 2009.

[11] Eren Gultepe, Jeffrey P Green, Hien Nguyen, Jason Adams, Timothy Albertson, and Ilias Tagkopoulos. From vital signs to clinical outcomes for patients with sepsis: a machine learning basis for a clinical decision support system. *Journal of the American Medical Informatics Association*, pages amiajnl–2013, 2013.

[12] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: Concepts and techniques.* Morgan kaufmann, 2006.

[13] Rave Harpaz, William DuMouchel, Paea LePendu, and Nigam H Shah. Empirical bayes model to combine signals of adverse drug reactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1339–1347. ACM, 2013.

[14] David Heckerman, Dan Geiger, and David M Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

[15] David W Hosmer Jr and Stanley Lemeshow. *Applied logistic regression.* John Wiley & Sons, 2004.

[16] Finn V Jensen. *An introduction to Bayesian networks*, volume 210. UCL press London, 1996.

[17] Michel CA Klein and Gabriele Modena. Estimating mental states of a depressed person with bayesian networks. In *Contemporary Challenges and Solutions in Applied Artificial Intelligence*, pages 163–168. Springer, 2013.

[18] Todd M Koelling, Monica L Johnson, Robert J Cody, and Keith D Aaronson. Discharge education improves clinical outcomes in patients with chronic heart failure. *Circulation*, 111(2):179–185, 2005.

[19] Stuart C Kramer and Harold W Sorenson. Bayesian parameter estimation. *Automatic Control, IEEE Transactions on*, 33(2):217–222, 1988.

[20] Harlan M Krumholz, Eugene M Parent, Nora Tu, Viola Vaccarino, Yun Wang, Martha J Radford, and John Hennen. Readmission after hospitalization for congestive heart failure among medicare beneficiaries. *Archives of internal medicine*, 157(1):99–104, 1997.

[21] Wentian Li and Dale R Nyholt. Marker selection by akaike information criterion and bayesian information criterion. *Genetic Epidemiology*, 21:S272–7, 2000.

[22] Naren Meadem, Nele Verbiest, Kiyana Zolfaghar, Jayshree Agarwal, Si-Chi Chin, and Senjuti Basu Roy. Exploring preprocessing techniques for prediction of risk of readmission for congestive heart failure patients. In *Data Mining and Healthcare (DMH), at International Conference on Knowledge Discovery and Data Mining (KDD)*, 2013.

[23] Mary D Naylor, Dorothy Brooten, Roberta Campbell, Barbara S Jacobsen, Mathy D Mezey, Mark V Pauly, and J Sanford Schwartz. Comprehensive discharge planning and home follow-up of hospitalized elders: a randomized clinical trial. *Jama*, 281(7):613–620, 1999.

[24] Richard E Neapolitan et al. *Learning bayesian networks*, volume 38. Prentice Hall Upper Saddle River, 2004.

[25] Kenney Ng, Amol Ghoting, Steven R Steinhubl, Walter F Stewart, Bradley Malin, and Jimeng Sun. Paramo: A parallel predictive modeling platform for healthcare analytic research using electronic health records. *Journal of biomedical informatics*, 48:160–170, 2014.

[26] Thomas Dyhre Nielsen and Finn Verner Jensen. *Bayesian networks and decision graphs*. Springer Science & Business Media, 2009.

[27] Judea Pearl. *Causality: models, reasoning and inference*, volume 29. Cambridge Univ Press, 2000.

[28] Judea Pearl, Thomas Verma, et al. *A theory of inferred causation*. Morgan Kaufmann San Mateo, CA, 1991.

[29] Michael W Rich, Valerie Beckham, Carol Wittenberg, Charles L Leven, Kenneth E Freedland, and Robert M Carney. A multidisciplinary intervention to prevent the readmission of elderly patients with congestive heart failure. *New England Journal of Medicine*, 333(18):1190–1195, 1995.

[30] Joanne Kraenzle Schneider, Susan Hornberger, Jane Booker, Alyce Davis, and Randy Kralicek. A medication discharge planning program measuring the effect on readmissions. *Clinical nursing research*, 2(1):41–53, 1993.

[31] Peter Spirtes and Christopher Meek. Learning bayesian networks with discrete variables from data. In *KDD*, volume 1, pages 294–299, 1995.

[32] Truyen Tran, Dinh Phung, Wei Luo, Richard Harvey, Michael Berk, and Svetha Venkatesh. An integrated framework for suicide risk prediction. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1410–1418. ACM, 2013.

[33] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

[34] Kiyana Zolfaghar, Jayshree Agarwal, Deepthi Sistla, Si-Chi Chin, Senjuti Basu Roy, and Nele Verbiest. Risk-o-meter: An intelligent clinical risk calculator. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1518–1521. ACM, 2013.

[35] Kiyana Zolfaghar, Naren Meadem, Ankur Teredesai, Senjuti Basu Roy, Si-Chi Chin, and Brian Muckian. Big data solutions for predicting risk-of-readmission for congestive heart failure patients. In *Big Data, 2013 IEEE International Conference on*, pages 64–71. IEEE, 2013.

[36] Kiyana Zolfaghar, Nele Verbiest, Jayshree Agarwal, Naren Meadem, Si-Chi Chin, Senjuti Basu Roy, Ankur Teredesai, David Hazel, Paul Amoroso, and Lester Reed. Predicting risk-of-readmission for congestive heart failure patients: A multi-layer approach. *arXiv preprint arXiv:1306.2094*, 2013.