University of Cincinnati

Date: 7/17/2015

I, Weizeng Ni, hereby submit this original work as part of the requirements for the degree of Doctor of Philosophy in Mechanical Engineering.

It is entitled:
Ontology-based Feature Construction on Non-structured Data

Student's name: Weizeng Ni

This work and its defense approved by:

Committee chair: Hongdao Huang, Ph.D.

Committee member: Lawson Wulsin, M.D.

Committee member: Nao Xu, Ph.D.

Committee member: Sundararaman Anand, Ph.D.

Committee member: David Thompson, Ph.D.
Ontology-based Feature Construction on Non-structured Data

A dissertation submitted to the
Graduate School
of the University of Cincinnati
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the Department of Mechanical Engineering
of the College of Engineering and Applied Science
by
Weizeng Ni
M.S. University of Cincinnati
July 2015

Committee Chair: Samuel H. Huang, Ph.D.
Abstract

Data mining on non-structured data is a relatively under-researched area because most efforts in the KDD community in the last decades are devoted to mining relational structured data. Thanks to the information explosion in the big-data era, the majority of knowledge is emerging in various forms of non-structured data. This necessitates new methodologies of constructing meaningful features from non-structured data to facilitate knowledge learning. Most existing data-driven methods only serve the objective of improving feature discriminative power, while severely underestimate the importance of interpretability. In many domains, the discovery and learning of new hypotheses and knowledge in a meaningful and understandable form from non-structured data is the prime aim.

In this study, an ontology-based feature construction framework is proposed. The framework presents the structural relations embedded with domain knowledge in the form of ontology. Features of 3 levels are defined based on the granularity in ontology. A feature, representing a domain hypothesis, can be readily constructed by evolving ontology. Support and confidence are two criteria proposed to evaluate the usefulness of the features in support of searching for optimal ones. Furthermore, in an interactive way, domain experts are involved to explore new hypotheses with the aid of data-driven heuristic algorithms. Also, ontology is highly flexible to be reconstructed in order to accommodate different hypotheses.

A comprehensive case study is conducted in which the proposed methodology is applied on a miscellaneous medical claim data to build features that are both interpretable and highly predictive for hospitalization forecast. A medical professor are constantly
consulted to bring in domain insights to aid ontology evolution and assess meaningfulness of the constructed features and prediction. The constructed features outperform those based on initial hypotheses in terms of prediction accuracy. Moreover, the ability of discovering new and useful knowledge is demonstrated by the meaningfulness of the new features and evolved ontology.
Acknowledgements

The dissertation will never be finished without the guidance of my advisor and committee members, and helps from my colleagues, friends and family.

Firstly I would like to express my deepest appreciation to my advisor, Dr. Samuel Huang, who has oriented and supported me with immense knowledge, patience and steadfast encouragement even when he was putting up a fight with disease. Doing research and studying with the guidance of Dr. Huang is an extremely enriching experience for me. My sincere thanks also go to the rest of committee members: Dr. Nuo Xu, Dr. Lawson Wulsin, Dr. Sam Anand, and Dr. David Thompson, for their insightful comments and persistent helps.

I also would like to thank all my colleagues and friends including Hua Li, Yi Zhang, Sury Yaddanapudi, Vishnu Madhvarayan, and Chuan Jiang.

Finally, many thanks to my parents, brother and sister. Supports and love from my family are the greatest fortune in my life.
# Table of Contents

Abstract ............................................................................................................................... ii

Acknowledgements ............................................................................................................ v

Table of Contents .............................................................................................................. vi

List of Figures .................................................................................................................... viii

List of Tables ..................................................................................................................... ix

1 Introduction ..................................................................................................................... 1

2 Literature Review ......................................................................................................... 6
   2.1 Literature search strategy ......................................................................................... 6
   2.2 Methodologies review ............................................................................................. 8
      2.2.1 Feature construction on structured data .............................................................. 8
      2.2.2 Feature construction on unstructured data ......................................................... 19
      2.2.3 Feature construction on semi-structured data .................................................... 37
   2.3 Data-driven V.S. knowledge-based .......................................................................... 42

3 Methodology Proposal .................................................................................................. 45
   3.1 Ontology constitution .............................................................................................. 45
   3.2 3-level ontology-based features .............................................................................. 49
   3.3 New hypotheses exploration ................................................................................... 52
      3.3.1 Genetic algorithm introduction ........................................................................... 53
      3.3.2 Genetic algorithm application on hypothesis evolution ..................................... 56
   3.4 Ontology reconstruction .......................................................................................... 57
4 Case Studies

4.1 Hospitalization prediction on miscellaneous medical claim data

4.1.1 Data overview

4.1.2 Evaluation criterion

4.1.3 Ontology construction

4.1.4 Predictive model

4.1.5 Feature construction

4.1.6 Ontology reconstruction

5 Conclusions

Reference
List of Figures

Figure 2-1: Flow chart of search strategy in the first Scopus database .......................... 7
Figure 3-1: Transitive property example. ................................................................. 48
Figure 3-2: Initial ontology for medical claim data .................................................. 49
Figure 3-3: Genetic algorithm (GA) workflow ......................................................... 53
Figure 3-4: Genetic algorithm - initial population creation ..................................... 54
Figure 3-5: Genetic algorithm – crossover ............................................................... 55
Figure 3-6: Genetic algorithm – mutation ............................................................... 55
Figure 3-7: A chromosome representing a candidate hypothesis ............................. 56
Figure 3-8: Elevated ontology .................................................................................. 59
Figure 3-9: Simplified ontology ............................................................................... 59
Figure 4-1: Initial ontology ....................................................................................... 64
Figure 4-2: Ontology specification in R ................................................................. 65
Figure 4-3: A chromosome representing an initial Sub_popu_POS ......................... 69
Figure 4-4: Comparison of optimal Diag_pro and derived Diag ............................. 75
Figure 4-5: Ontology reconstruction ....................................................................... 76
List of Tables

Table 2-1: Summarization of selected paper in the first Scopus search .......................... 7
Table 3-1: Variables in a medical claim data........................................................................ 45
Table 3-2: Classes and properties in the medical claim data example .............................. 46
Table 3-3: Relations and domains in the medical claim data example ............................ 47
Table 4-1: Variables of medical claim data ........................................................................ 61
Table 4-2: Evaluation criterion benchmarks ..................................................................... 63
Table 4-3: Relations and domains ...................................................................................... 64
Table 4-4: Classes and properties ....................................................................................... 65
Table 4-5: Comparison of the optimal and initial constructed feature Count_POS .......... 70
Table 4-6: Comparison of the optimal and initial constructed feature Diag_proc ........... 72
Table 4-7: Subset of population of diagnosis, procedure and month ............................... 73
Table 4-8: Classes and new properties ............................................................................. 74
1 Introduction

1.1 Problem Statement

In the problem of learning knowledge and patterns from data, most of learning algorithms (e.g., classification models, regression models, and clustering methods) are only applicable to relational structured data where all samples are uniformly characterized by a set of features. This type of data is generally referred to as structured data. Most efforts in the Knowledge Discovery and Data Mining (KDD) community in the last decades are devoted to mining it. Thanks to the information explosion in recent years, however, the majority of knowledge does not necessarily resident perfectly in the structured data. In this information era, the most prevalent formats of raw data include web sites, images, plain texts, time-series signals, and so forth. The data of any representation except for relational representation are collectively termed non-structure data. The need of learning knowledge from non-structured data gives rise to an unprecedented challenge of bridging the gap between raw dataset and learning algorithms. Transforming raw non-structured data to easy-to-learn structured data is typically realized by constructing meaningful features from raw data. The prime concern in feature construction is, specific to a leaning problem, maintaining genuine and comprehensive information to the greatest extent, which guarantees that it can be fully exploited in subsequent analyses including feature selection, modeling, and validation. As the first step in pattern learning problem, feature construction is viewed as the most pivotal step because analyzing on “bad” constructed features is pointless.

For the topic of feature construction on non-structured data, an extensive body of literature review reveals the lack of unified and robust framework. The domain-specific
structural relation in the non-structured data necessitates the integration of domain knowledge and data in the construction process, which is widely neglected in the existing approaches. The data-driven approaches without aid of domain knowledge do not fully capitalize the domain-specific information in the data, and the resulting features usually cannot be interpreted. On the other hand, knowledge-guided approaches are likely missing substantial data-embedded information and incapable of discovering new knowledge. This study aims to develop a unified feature construction framework on non-structured data that allows interactive incorporation of domain knowledge and data information to build highly discriminative, hypothesis-based, and interpretable features. Moreover, new knowledge in format of hypothesis and feature can be discovered and assessed.

1.2 Background

Non-structured data can be divided into two subclasses, namely unstructured data and semi-structured data. For an unstructured dataset, a sample is essentially comprised of, instead of explicitly expressed features, a set of basic elements and no distinct structure (e.g. hierarchical relationship) among the basic elements exists. Spatial relationship among basic elements is the only relationship to be concerned. For instance, digitized image can be represented as tonal values of pixels, and all pixels are distributed uniformly without any hierarchical relationship. The spatial relationship is reflected on the locations of pixels within an image. A text document is comprised of a sequence of ordered terms like words, characters, and numbers. The spatial relationship is reflected on the term order. Note that if the spatial relationship is removed, it leads to a simpler data representation, namely, “bag of basic elements”. Many traditional feature construction
methods on text assume that term order has little useful information thus view a
document as a “bag of words”. Semi-structured data is represented in a pre-defined non-
relational data model. The data model is normally defined manually based on domain
knowledge. An illustrative example is Extensible Markup Language (XML) data, where
the data model is defined by a set of rules for document encoding. An element in XML is
a logical document component in form of Tag markup construct. The content in the
element may contain markup and other elements, which are called child elements. This
format makes XML data both human-readable and computer-readable. Another example
is transaction-like data (i.e. medical claims data, and credit report). The instances are
characterized by multiple features and events, and events can be further specified by a set
of characteristics, such as event type, event time. Dataset of this type is named event-
centric data.

In banking industry, credit report is an important event-centric data based on
which credit worthiness is assessed, loan products are constructed, and lending decisions
are made. Traditionally, the features are constructed in ad hoc fashion by experts, which
has resulted over the last several decades a “standard set” of several hundred features.
However, it is believed that these features only include 10-20% usable information from
raw credit report. So a systematic approach automatically constructing features that
contain most useful information will go a long way in providing effective lending
decision support. In addition, the approach can be steadily applied to other data of similar
event-centric representation such as healthcare insurance claims.
1.3 Objectives

The objectives to be achieved require the proposed ontology-based feature construction framework to be capable of:

1) Designing a concise and meaningful representation for non-structured data where domain knowledge and data information can be interactively integrated;

2) Constructing features that have competitive discriminative power for prediction and high interpretability;

3) Discovering new knowledge from data and evaluating the legitimacy of new and existing hypotheses.

1.4 Intended Contributions

Should the proposed framework be successfully implemented and the above objectives be achieved, the impact would be profound and far-reaching. First, the framework is highly general so its application can be readily extended to the data with various structural relations in different fields such as medical, finance, and social media. Second, the framework can substantially reduce the time and efforts of searching for optimal features for modeling. The intuitive rationale behind the framework allows domain practitioners to easily apply, so they can focus on tackling the domain-specific problems which is what they are good at. Finally, equipped with this framework, domain experts are able to explore new knowledge within the flexible constrain of interest, so the understanding for a certain area can be notably advanced.
1.5 Dissertation Layout

Section 2 describes the extensive body of literature review about the existing approaches. First, a literature search strategy is depicted to show that the best efforts have been made to exhaustively investigate existing approaches. The review is presented in the order of 3 data structures: structured data, unstructured data, and semi-structured data. Finally, the primary drawback for the existing approaches is pointed out, namely, the lack of integration of domain knowledge and data information, and it motives this study.

Section 3 introduces the proposed ontology-based feature construction framework. It starts with how the concept of ontology is employed as the representation for non-structured data. Then the ontology-based representation leads to the 3-levels features to be constructed. The biggest focus is the application of heuristic algorithms to build relation-level features and explore new knowledge. Lastly, the functionality of reconstructing ontology based on hypothesis evolution is illustrated.

Section 4 presents a case study where the proposed framework is applied on a miscellaneous medical claim data. Other than the proposed method, the choices of evaluation criterion, predictive model, and result benchmarks are discussed. The resulting features and hypotheses are evaluated in terms of prediction power and interpretability. The results verify the feasibility and advantages of the proposed framework.

Section 5 concludes this dissertation with conclusions and remarks on future research.
2 Literature Review

2.1 Literature search strategy

Across different fields, the concept of feature construction is expressed in different terms. Specifically, “feature” can be called “variable” and “attribute”, and “construction” equals to “creation”, “generation”, or “extraction” in some scenarios. Thus, we include combinations of these terms as keywords for literature searching. In the bibliographic database Scopus, the query -- “feature construction” OR “feature creation” OR “feature generation” OR “variable construction” OR “variable creation” OR “variable generation” OR “attribute construction” OR “attribute creation” OR “attribute generation” is searched in article title, abstract, or keywords, and totally 1329 documents are returned (as of Nov 2013). The returned documents are sorted by citation. The abstracts of the top 200 papers with citations ranging from 505 to 9 are looked into. The papers explicitly describing methodologies of feature construction are selected for further investigation. The search strategy is summarized as the flow chart in Figure 2-1. 48 papers are finally selected, and the numbers of papers belonging to different data type and application domains are summarized in Table 2-1. Note that no paper is found to be related to semi-structured data.

As the semi-structured data is the data type that this dissertation primarily targets, several more Scopus searches specific for semi-structured data are conducted. The new queries include such keywords as “websites”, “financial transaction”, “healthcare insurance claims” and so forth. The most cited papers in the result are selectively reviewed and most relevant ones will be described in the following section. Note that apart from the papers in the Scopus search result, more than one hundred papers showing
up in bibliographies are also examined, and the most relevant ones are included in this review. In addition, some relevant patents are searched and one of them named “data spider” is found to be a good reference, which will be discussed in detail.

Table 2-1: Summarization of selected paper in the first Scopus search

<table>
<thead>
<tr>
<th>Application domain</th>
<th>Paper number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structured data</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>digitized image</td>
</tr>
<tr>
<td></td>
<td>acoustic signal</td>
</tr>
<tr>
<td></td>
<td>text</td>
</tr>
<tr>
<td></td>
<td>biology sequence</td>
</tr>
<tr>
<td>Unstructured data</td>
<td></td>
</tr>
<tr>
<td>Semi-structured data</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 2-1: Flow chart of search strategy in the first Scopus database
2.2 Methodologies review

2.2.1 Feature construction on structured data

Feature construction on structured data is essentially realized by constructive induction (Michalski 1983), which has mainly been used with decision trees learning (Rizoiu, Velcin et al. 2013). Constructive induction is defined as follows (Matheus and Rendell 1989): a set of constructive operators \( \{ o_1, o_2, ..., o_n \} \) are applied on a set of primitive features \( \{ f_1, f_2, ..., f_m \} \), resulting in the construction of one or more new features \( \{ f'_{1}, f'_{2}, ..., f'_{N} \} \). For the sake of simplicity, the terms of constructive induction and feature construction on structured data are used interchangeably.

In a pattern learning problem, preparing data representation is the very first and vital step. Transformation of data representation is generally comprised of feature construction, feature selection, and feature abstraction (Motoda and Liu 2002). Feature construction differs from feature extraction and selection in that no feature or information is lost, so it is normally undertaken prior to the other steps. Feature construction is essentially motivated by the difficulty of models learning some complex patterns. For a specific application or a concept to be learned, some concept-related patterns are not explicitly expressed by the original features. For example, in a XOR problem, two features \( f_{1}^{\text{xor}}, f_{2}^{\text{xor}} \) jointly determine the class of a instance while each one has no individual discriminative power. Few predictive models have the ability of discovering this interaction pattern without aid of feature construction. So a new feature defined as \( f_{1}^{\text{xor}} \text{ XOR } f_{2}^{\text{xor}} \) can help predictive models make correct prediction. In finance domain, it is believed that the financial ratios of accounting variables are more meaningful and effective in prediction than original variables. Thus, constructive induction can be
summarized as: with or without the help of domain knowledge and hypothesis, exploit useful patterns and underlying structures in the data and express them more explicitly in terms of new features to make the concept to be learned more effectively and easily.

Based on the criteria by which the newly constructed features are evaluated, constructive induction can be roughly categorized as *interleaving* approach and *preprocessing* approach (Otero, Silva et al. 2003, Espejo, Ventura et al. 2010). This categorization strategy is analogous to classification of *wrapper* and *filter* in feature selection. In a interleaving approach (Zheng 2000), the processes of constructing features and concept learning are intertwined in a such way that the learning performance of the a predictive model is the fitness function which evaluates the discriminative power of the newly constructed features. In contrast, preprocessing approach (Bensusan and Kuscu 1996, Kuscu 1999, pOtero, Silva et al. 2003, Muharram and Smith 2005) isolates the process of feature construction from learning process, and adopts some entropy-based or statistical criteria as the fitness functions. In general, interleaving approach creates features more tailored to the models used for evaluation, at expense of more intensive computation.

Based on the information used to guide the search in construction space, constructive induction can be classified into 3 groups: *hypothesis-driven*, *data-driven*, *knowledge-based* (Bloedorn and Michalsi 1998). In a hypothesis-driven method, a hypothesis is firstly chosen (e.g. decision tree); new features are constructed based on this specific hypothesis; and then are added to current feature set. This augmented feature set derives a new hypothesis which can be used for constructing next new feature. For instance, Fringe (Pagallo 1989) and DC Fringe (Yang, Rendell et al. 1991) make use of
hypotheses previously generated by a similarity based learner. Their details will be discussed later. Free from the limitation of any hypothesis, data-driven approach only looks into the inherent relationship conveyed by the instances. Illustrative examples can be found in (Hu and Kibler 1996, Bloedorn and Michalsi 1998). In the case of knowledge-based approach, the knowledge provided by experts determines the search in the constructive space.

During the 1980s and early 1990s, the early general feature construction systems are typically based on hypothesis of decision tree, including FRINGE (Pagallo 1989), CITRE (Matheus and Rendell 1989), DC FRINGE (Yang, Rendell et al. 1991), MIDOS (Gamberger and Lavrac 2002), EXPLORA (Klösgen 1996), and TERTIUS (Flach and Lachiche 2001). Three of them will be discusses in the following.

The FRINGE (Pagallo 1989) methodology is essentially motivated by the replication problem in decision tree learning. For a decision tree, some sub-trees with same structure may be frequently replicated within a tree. Thus in this complicated tree, the instances are distributed over multiple sub-trees with same structures. Fewer samples may compromise the accurate prediction probability estimate. Replication problem can be alleviated or eliminated by building composite features with the implementation of FRINGE. In the problem of classifying a dataset with a set of boolean features, a decision tree is built and pruned. Define paths from the root to positive leaves as “merging disjunctive regions”. For each feature \( f_{curr} \) in this region, conjoint current feature \( f_{curr} \) and its predecessor \( f_{pred} \) in form of \( f_{curr} \cdot f_{pred} \) if positive value of \( f_{pred} \) leads to testing \( f_{curr} \); otherwise, in form of \( f_{curr} \cdot \overline{f_{pred}} \). The composite features are added to the current feature set and a new decision tree is subsequently built. This process
is repeated until no feature can be generated. The novel algorithm is experimentally proven to outperform standard decision tree for data with boolean features.

CITRE (Matheus and Rendell 1989) also adopt decision tree for feature construction, but it differs from FRINGE in its use of domain knowledge, feature generalization, and information-theoretic evaluation. The operator used in CITRE is binary \textit{and}(operand, operand), where operands are boolean expressions of \textit{equal}(feature, value). For a decision tree with nominal features in nodes, the links are the boolean expression, thus are possible operands. The operand selection is also biased by the concept of “merging disjunctive regions”. All possible links in the “merging disjunctive regions”, the branches from the root to positive leaves, are selected as candidate operands. Applying the binary operator on all operand pairs produces new features. Candidate operands set could also be further narrowed down by applying, if available, domain knowledge.

DC Fringe (Yang, Rendell et al. 1991) is a variant of FRINGE with two improvements: 1) both conjunction and disjunction are used; and 2) the node types (positive or negative) of siblings and predecessor’s siblings are taken into account. If the sibling of predecessor of the current feature \((f_{curr})\) are positive leaves, disjoint the \(f_{curr}\) and \(f_{pred}\), otherwise, conjoint them. In the experiments with small DNF and CNF concepts, DC Fringe shows better performance than FRING in terms of accuracy, conciseness, and efficiency.

In (Piramuthu, Ragavan et al. 1998), DC Fringe system has been used to transform original financial data to a new representation easing the concept learning of neural networks. This work is motivated by the challenges associated with both financial
data and neural network. The financial data is normally characterized by frequent data noise, lack of functional structure, and high attribute interactions. These inherent properties will cause learning difficulty in the sense that genuine concept may be missed by the predictive models which cannot capture feature interactions or is susceptible to excessive noise. Back-propagation (BP) neural network is highly popular in financial classification because it is tolerant to noise and requires little statistical assumptions. But it is slow due to the enormous size of search space and the iterative manner. After new features are constructed by DC Fringe, the underlying concept will be described more concisely and comprehensively. Moreover, few hyperplanes separating different classes will be needed, thus the degree of dispersion on the search space is reduce. The empirical studies on three real-word financial datasets demonstrate the improvement in both learning speed and classification accuracy. It also shows that the gap between training and test accuracy is narrowed down, which implies the improvement of generalization ability.

These general feature construction systems are normally working without considering domain-specific knowledge and under many restrictions. With the aim of incorporating domain-specific knowledge and hypothesis into data-driven construction and releasing restrictions, (Markovitch and Rosenstein 2002) proposes a generalized and flexible framework compatible to any given set of constructor functions. Feature space specification (FSS) is firstly defined by user, which specifies 1) primitive features, 2) constructor functions, 3) domain and range of constructor function, and 4) constraints over constructor functions. Specifically, a FSS is made up of three arguments: type, basic-feature, and constructor-function. The types are either nominal, ordered-nominal, or
continuous. Basic-features are defined by feature ID and type. Constructor-function is defined by ID, return type and the specification of their input arguments. For examples in the domain of tic-tac-toe end games, a type of “slot” is the ordered-nominal value of a board slot. A constructor-function can be defined as

\[ \text{Max} := \text{Slot}, \langle \text{set of Slot}, \{\text{NoConst}\} \rangle, \]

where “set of Slot” is the set of basic-features, and “\{NoConst\}” is an argument constraint function forbidding constant features. This constructor-function receives the input of a set of slots without constant, and returns the maximal value of the slots. After the search space is defined by feature space specification, four types of search operators are applied on one or two features to produce new features. Compose operator composes new features using all proper constructor in FSS; inset operator inserts one feature into the other; replace operator replaces component of one feature with the other; and interval operator creates a new feature based on whether a feature lies within a specified range. This feature construction algorithm is applied on a variety of classification problems. It is shown that predictive performances are significantly improved, and it sheds light on that the underlying concept is better described by the new feature representation.

With the substantial improvement of computing capability in computer, Genetic programming (GP) is emerging as the most widely used tool for inductive construction. As opposed to the early systems, GP performs more exhaustive global search on candidate solutions. It also has an advantage of better coping with interactions, and being less likely to get trapped in local maxima (pOtero, Silva et al. 2003). In a GP procedure, the individual program is mostly encoded as tree structure. Other common representations include linear GP, Cartesian GP, and grammatical GP (Neshatian, Zhang
et al. 2012). In a tree-based GP procedure, each individual tree consists of a root, internal nodes, and terminal nodes. Correspondingly, the root stands for constructed feature; terminal nodes are operands (primitive features); and internal nodes represent operators. The initial population of individuals comprised of a set of trees is first created randomly. Then it undergoes revolutionary process repeatedly until terminate criteria are met. The initialization mechanism is characterized by the scheme of limiting tree size, candidate operators, etc. The revolutionary process generally consists of generation selection, crossover, mutation, fitness evaluation, and so forth. As the search in construct space is autonomously guided by GP and normally no domain knowledge and hypothesis is involved in evolutionary process, this class of methods belongs to data-driven approach. Some studies using different tree-based GP approaches in terms of different characteristics are presented in the following.

(Smith and Bull 2005) develops GAP algorithm which uses GP for feature construction and genetic algorithm (GA) for feature selection. Given a dataset with $m$ numeric features, each individual is composed of $m$ GP trees. In the step of initializing GP trees, the chance of choosing a primitive feature on a certain node is given by:

$$P = 1 - \frac{1}{(d+1)^2},$$

where $d$ is the depth of the node. In this way, a root has depth of 1, and therefore has 50% chance of having no descendants. If it is determined that a primitive feature would be on a node, a feature will be chosen randomly from the pool of $m$ primitive features. Otherwise, an operator will be randomly selected from the set \{*, /, +, −, \%\}, and two child nodes are produced. This initialization mechanism implicitly limits the tree size. For each generation in evolutionary process, the fitness of an individual ($m$ composite features) is evaluated by the predictive accuracy of C4.5 in the 10-fold cross
validation manner. So this belongs to interleaving approach. Tournament selection is used to choose parents for the next generation. On two selected parents, two-point crossover is applied and it occurs both on individuals level (whole trees are exchanged between individuals) and tree level (sub-trees are exchanged between trees at the same position in each parent). Mutation is realized by replacing randomly generated subtree to selected subtree with a use-defined mutate rate. Additionally, the order of the trees between two randomly chosen loci may be reversed with the user-defined inversion probability. After a set of new features are created, genetic algorithm based feature selection method is applied, and the fitness of a feature subset is also evaluated by the predictive accuracy of C4.5 in the 10-fold cross validation manner. The GAP algorithm is evaluated on ten datasets from UCI repository, and the experimental results show that this approach outperforms C4.5 alone.

In (Otero, Silva et al. 2003), the tree size is limited by explicitly specifying a parameter representing the maximal number of nodes, instead of depth. The choice stems from the observation that if a maximum tree depth is specified, the GP will probably be somewhat biased to produce balanced trees. The operators used include arithmetic operators \{\ast, /, +, -, \%\}, and two relational comparison operators \{\leq, \geq\}. The outputted features have values of either continuous or boolean. Moreover, it uses information gain ratio as the fitness function to evaluate constructed features. Thus, this method can be classified as preprocessing approach.

In another study (Lin, Ke et al. 2008), the author applies a variant of basic tree-based GP with some specific configurations. Other than primitive features and operators, constants are also involved as tree nodes. The tree size is explicitly restricted as the
maximal tree depth $d$ which is defined by a user. In the wrapper manner, the fitness of generated feature set is evaluated by the formula: $f = \text{Sensitivity} \times \text{Specificity}$. For each generation, only two individuals with highest fitness score will be reproduced. Crossover and mutation are performed on remaining individuals to produce offspring. In order to increase population diversity and avoid local optimum, the probability of mutation $P_m$ is increased as generation increases. The evolution process is iterated until the maximal iteration number is reached or fitness score of an individual equals to 1.0.

For the tree-based GP methods belonging to preprocessing category, the fitness criterion in feature construction process can be the same fitness criterion of decision tree classifier in modelling process. A question can be raised as to how the relationship between the fitness criteria used in GP and decision tree modelling affects the performance. This question is investigated in (Muharram and Smith 2005). The experiment is conducted by basic tree-based GP methods, with four different fitness functions, namely, information gain (IG), gini index (IG), IG + GI, and $\chi^2$. The classifiers employed are C5, CART, CHAID, and artificial neural network (ANN). The first three classifies use, correspondingly, IG, IG, and $\chi^2$ as splitting criteria. ANN separates instances by constructing nonlinear hyperplane. Different combinations of fitness measures and classifiers are applied on five dataset to compare prediction performance. It is shown that GP-based feature construction helps improve the performance of a classifier, sometimes significantly. More importantly, it is demonstrated that the improvement is not significantly dependent on which fitness measure is used in GP.
In addition to these GP-based feature construction methods, some non-GP feature construction methods have also been developed specific to some domains. Following are two illustrative examples.

In the domain of microarray classification, (Hanczar, Zucker et al. 2007) proposes a dimension reduction method termed \textit{FeatKNN}, whereby every new feature is created from two most informative gene pair. This is essentially motivated by the observation that much discriminative power of genes is conveyed by the interaction between multiple genes. These interaction effects are easy to be overlooked by various models if they are not expressed explicitly. In the implementation of \textit{FeatKNN}, the most informative gene pairs are firstly identified based on mutual information criterion in a sequential forward search (SFS) procedure. For a selected gene pair \((g_i, g_j)\), an instance \(x\) can be projected to the 2-dimensional space defined by \(g_i\) and \(g_j\). The probability of \(p_a(x)\) and \(p_b(x)\) denote the probabilities of instant \(x\) belonging to class \(a\) and class \(b\) respectively. With the application of \(k\)-nearest neighbor algorithm, \(p_a(x)\) can be approximated by formula:

\[
p_a(x) = \frac{n_a(x)}{k}, \quad \text{where } n_a(x) \text{ is the number of neighbors belonging to class } a, \text{ and } k \text{ is the total number of neighbors.}
\]

Such that the new feature is defined as \(f_{ij}(x) = p_a(x) - p_b(x) \approx -1 + 2 \frac{n_a(x)}{k}\). In essence, feature \(f_{ij}(x)\) reflects the ability of \(g_i\) and \(g_j\) to jointly differentiate two classes. The experiment result demonstrates the performance improvement on models predictions and the prevalent interaction effect embedded in genes.

In the application of chemical process control (Piramuthu and Sikora 2009), all possible pairs of \(m\) primitive features are considered as operands, and four arithmetic operators \(\pi = \{+, -, \%, /\}\) are applied on all operands. This scheme leads to \(4 \times \left(\begin{array}{c} m \\ 2 \end{array}\right)\)
new features in each iteration. All new features and primitive feature are individually evaluated by $\chi^2$ statistic. Only a set of features with highest ranks are carried over to the next iteration. The iteration terminates when criterion concerning $\chi^2$ statistic is satisfied. This method explores all possible pairwise interactions in each iteration at expense of highly computation burden.

In summary, inductive construction methods are generally being developed from early general-purpose systems to the evolutionary algorithm-based paradigms and some domain-specific approaches. Early general-purpose systems are only capable of searching in limited feature space and dealing with features of certain value types (e.g. boolean and nominal). While evolutionary algorithm-based approach expands the search space substantially and adapt to features of various value types. For some domains, domain-specific approaches introduce domain knowledge directly to assist feature searching and they are found to be especially effective. In general, the most common practice in inductive construction is using tree-based GP algorithm. From the previous applications, it shows the following advantages. First, tree structure is highly understandable, so new features are more likely to be accepted by experts. Second, the complexity of constructed features can be easily controlled by constraining tree depth or node number. Third, domain knowledge can be steadily introduced by choosing domain-specific construction operators. For example, it is believed in financial community that financial ratios are more effective than the original variables in identifying potential problem banks, the operator “/” is included in GP algorithm.
2.2.2 Feature construction on unstructured data

In the Scopus search result, digitized image, time-series signal, text, and biology sequence are the four types of raw data where feature construction methodologies are explicitly illustrated and applied. The basic elements for digitized image are tonal values of pixels; for time-series signal are magnitude values at a given time; for text are words, punctuations, and numbers; for biology sequence are sites representing amino acids or nucleotide bases. Evidently, most machine learning algorithms and statistical models cannot directly learn patterns from basic elements since usable information embedded in an individual basic element is extremely limited. Thus, constructing easy-to-model features is emerging as an indispensable and significant step which proceedings such steps as feature selection, feature extraction, and concept learning. In the following four subsections, methods corresponding to these four types of unstructured data are illustrated.

2.2.2.1 Digitized Image

Extracting features from digitized images has served to address a broad spectrum of problems including object detection (Rothwell, Zisserman et al. 1995, Chen, Man et al. 2005), object tracking (Collins, Liu et al. 2005), visual recognition (Gader and Khabou 1996, Georgieva and Jordanov 2009).

In the object detection problem, a digitized image is usually represented as a 2-D array, where each entry corresponds to a pixel with a binary, discrete or continuous tonal value. Before being fed to some form of learning algorithms, raw image data is normally transformed to conceptual information by applying transform methods on a portion of image within sliding windows (Agarwal and Roth 2006). The constructed features are
normally termed *pixel statistics*, which are generally calculated in terms of such factors as the zone’s shape, orientation, fields representing size, and statistics. Some low-level features can be easily extracted such as color, texture, shapes, etc (Tsai 2007). To further interpret an image, spectral, textural, and contextual information are considered as three fundamental pattern elements, and the corresponding three groups of features are generated by certain transform methods.

Spectral features reflect the variations of tonal values in different spectral bands (e.g. RGB and NIR color space). Corresponding transform methods include color histograms and spectral moments (Weinbach and Grinnell 1998). Color histogram takes into account the occurrences of pixels having colors within one of a series of discrete color intervals, and no spatial information among pixels is concerned. Spectral moments consider the spatial information of various colors. The four low-order moments are defined as

\[
\mu = \frac{1}{N} \sum_{i=1}^{N} s_i, \quad \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (s_i - \mu)^2}, \quad sk = \left( \frac{1}{N \times \sigma^3} \times \sum_{i=1}^{N} (s_i - \mu)^3 \right)^{1/3},
\]

and

\[
k_u = \left( \frac{1}{N \times \sigma^4} \times \sum_{i=1}^{N} (s_i - \mu)^4 \right)^{1/4};
\]

where \(s_i\) is tonal value of the \(i\)th pixel for one spectral band, \(N\) is number of pixels within a window, \(\mu\) is arithmetic mean, \(\sigma\) is standard deviation, \(sk\) stands for the third moment skewness, and \(k_u\) standards for the fourth moment kurtosis. In a study of classifying object-based vegetation (Li, Hayward et al. 2010), spectral moment features compare favorably to that of other state-of-the-art textual features in term of discriminative power.

Textural features contain the information about the spatial distribution of tonal variance within a band (Haralick, Shanmugam et al. 1973). The most widely used approaches extracting textual features are Gray-Level Co-Occurrence Matrix (GLCM) (Haralick, Shanmugam et al. 1973), Local Binary Patterns (LBP) (Ojala, Pietikainen et al. 2002).
2002), and Laws’ filter masks (Laws 1980). Some studies have showed that GLCM is the best one in terms of high texture discrimination accuracy (Tso and Mather 1999, Mather and Tso 2010). A brief description of GLCM is given below. This method is based on the assumption that the texture information is contained in the overall or “average” spatial relationships which the gray tones in the image have to one another. In this sense, gray-tone spatial-dependence matrices are generated which are assumed to contain all the texture information. Given a discrete 2-D image array, the gray-tone spatial-dependence matrices are in form of $DTV \times DTV$, where $DTV$ are number of discrete gray-tone values. Each matrix is associated with two parameters, namely, distance $d$ and angular relationship $\text{angl}$ between neighboring pixels. The value of entry $(i,j)$ in a matrix is the number of times gray tones $i$ and $j$ have been neighbors with distance of $d$ and angular relationship of $\text{angl}$. For each gray-tone spatial-dependence matrix, 14 textural features are constructed based on specific equations. One textural feature named angular second-moment feature (ASM) is detailed for illustration. ASM feature is defined as $f_{ASM} = \sum_{i=1}^{DTV} \sum_{j=1}^{DTV} \left( \frac{P(i,j)}{R} \right)^2$, where $P(i,j)$ is the value of entry $(i,j)$ in a gray-tone spatial-dependence matrix. ASM feature is a measure of homogeneity of an image. Given a homogeneous image, there are few dominant gray-tone transitions. So there are fewer entries having large values, and ASM feature will be smaller. Other than homogeneity, other specific textural characteristics of the image such as contrast and complexity are related to other textual features. In a study of classifying cork tiles of seven types (Georgieva and Jordanov 2009), a set of textural features generated by GLCM and Laws’ filter masks have been adopted and combined to describe the cork tiles. Coupled with PCA and LDA for data preprocessing, neural networks for classification, the
experimental results show competitive performance of the textural features compared to other similar methodologies.

The context in object-based image can be defined as how the probability of presence of one object is affected by its neighbors (Tso and Olsen 2004). A simple way of obtaining contextual feature is majority voting within a predefined window. A more sophisticated method is Markovian Random Fields model (MRF) (Tso and Mather 1999). In a scene classification study (Tso and Olsen 2004), it is demonstrated that incorporating contextual features extracted by MRF with spectral and textural features helps further enhance classification performance.

In the field of handwritten digit and character recognition, images are often characterized by binary pixel tonal values. Some domain-specific features have been explored and investigated by researchers. For instance, three groups of features are proposed by (Favata and Srikantan 1996), namely, gradient features, structural features, and concavity features. They emphasize the relationship between a pixel and the neighbors at ranges of, respectively, local, intermediate, and global. A binary image is first imposed a 4×4 grid, leading to 16 regions. Then features are independently extracted from each region. The gradient of a central pixel is a function of its derivatives with respect to its eight nearest neighbors. It can be computed by employing 3*3 Sobel operators. The direction of a gradient falls into one of 12 predefined direction ranges. For each direction range, a histogram is built for all pixels, counting the number of pixels whose gradient direct within the given range. A threshold is applied to the histogram, and then a binary gradient feature is obtained. This scheme leads to 12×4×4=192 binary gradient features. Structural features capture “mini-strokes” pattern in the gradient map.
They are obtained by applying 12 manually-built rules to each pixel, operating on its eight nearest neighbors. It also produces $12 \times 4 \times 4 = 192$ binary features. Concavity features include three subclasses of features: coarse density, large stroke, and concavity. Coarse density features are derived from counting the pixels falling into a given grid. Large stroke features are computed from run lengths of horizontal and vertical black pixels across the image. Concavity features are extracted by applying a starlike operator. Concavity group has total $4 \times 4 \times (1 + 2 + 5) = 128$ binary feature. The effectiveness of these three groups of feature is evaluated by $k$-nearest neighbor algorithm in the applications of digit string segmentation and handwritten word recognition. The constructed features show high capability of recognizing valid class members.

Object tracking problem is quite distinctive in that feature construction and object identification are conducted iteratively as frames are updated and it is normally conducted on-line. For a frame corresponding to a given time, new discriminative features are created and selected based on the object identified in the previous frame, and then object for this frame is identified based on features newly obtained. In a study (Collins, Liu et al. 2005), seed features are firstly constructed for all pixels in the formula:

$$f_{seed} = \{w_1R + w_2G + w_3B \mid w_\ast \in [-2, -1, 0, 1, 2]\},$$

where $R$, $G$, and $B$ are three additive primary color values. After ruling out redundant coefficients, 49 seed features can be obtained for each pixel. Within a new frame, a rectangular set of pixels covering object and an outer margin of pixels covering background are chosen in the previous frame. Given a seed feature $f_{seed}(i)$, calculate $H_{obj}(i)$ and $H_{bg}(i)$ as histograms of feature values for pixels on, respectively, object and background. The empirical discrete probability distribution $p(i)$ for the object, and $q(i)$ for the
The new features are created as the log likelihood ratio of class conditional seed feature distributions. The formula is given as: 

\[ F(i) = \log \frac{\max(p(i))}{\max(q(i))} \]

The features have positive values for colors distinctive to the object, and negative values for colors associated with the background.

From the studies noted above, one strong pattern emerges is that most transformation methods of feature construction are conducted manually based on domain knowledge, while classification is conducted automatically by computer (Lillywhite, Lee et al. 2013). These basic transformation methods are typically derived based on specific considerations with respect to certain domains, and they are implemented independently. So a feature set successfully addressing a problem is not necessarily also applicable to another problem or even a different data for the same problem. It gives rise to a need of, when constructing features, further expanding potential feature space by choosing and combining multiple transformations. Hence, a wealth of efforts is increasingly devoted to integrate these transformation methods in an automated fashion to create higher-level conceptual features with higher discriminative power. For example, in order to more extensively explore potential pixel statistics with high predictive power, some studies (Krawiec and Bhanu 2003) (Lin and Bhanu 2003) try to extract pixel statistics automatically with implementation of genetic programming (GP). Each inner node in a GP tree represents a transformation method, and the root is the constructed feature. Another study (Roberts and Claridge 2004) presents a system evolving derivation of pixel statistics and normal tree based object detector simultaneously -- in data mining terms, feature construction step interleaves model selection process.
The most highly data-driven feature construction methods we find are Evolution-COnstructed (ECO) method (Lillywhite, Lee et al. 2013) and a Genetic Programming-based (GP-based) image classification system (Atkins, Neshatian et al. 2011). In the former study (Lillywhite, Lee et al. 2013), Every ECO feature is represented by three elements, namely subregion \((x_1, y_1, x_2, y_1)\), the transforms \((T_1, T_2, \ldots, T_n)\), and the parameters for each transform \(\emptyset_i\). Totally 27 Transforms \(\emptyset_i\) are considered, and they include Fourier transform, Sobel operator, histogram equalization, pixel statistics, Gabor filter, gradient, and so forth. Each transform \(\emptyset_i\) has 0-6 possible parameters. The choices with respect to these three elements are explored by genetic algorithm with functions of cross over and mutation. The fitness of constructed features is evaluated by the performance of perceptron classifying the test data. Specifically, the fitness score is given as: \(s = \frac{500t_p}{f_n + t_p} + \frac{500t_n}{f_p + t_n}\), where \(t_p\) stands for the number of true positives, \(t_n\) stands for the number of true negative, \(f_p\) is the number of false positive, and \(f_n\) is the number of false negative. Compared to accuracy, the fitness score is insensitive to class unbalance. This approach creates a finite set of highly discriminative. The experiment shows that ECO features compare favorably to hand-crafted state-of-the-art object recognition algorithm.

For the GP-based image classification system (Atkins, Neshatian et al. 2011), features and classifier evolve in an individual program interactively and automatically, and no expert intervention is needed. From root to leaves, an individual program is comprised of classification tier, aggregation tier, and filtering tier. The input to filtering tier includes raw image, window, and integer. After applying functions (i.e. Mean, Add, Div), a single transformed image is outputted. Aggregating tiers apply some forms of
operators (i.e. StDev, Max, Min) on input (made up of image, window center, and window size), and return double-precision real number. Lastly, output from aggregating tiers and random number are fed to classifying function to make a prediction. Filtering and aggregating steps are regarded as the process of feature construction. This system is compared to baseline system based on human-extracted features, and they show similar performances although it does not necessarily use the same features extracted by human experts.

2.2.2.2 Time-series Signal

Acoustic signal is the time-series signal that researchers mostly encounter, and in the Scopus search result, classifying acoustic signals is the only problem on time-series signal. Therefore, we confine ourselves to the classification problem on acoustic signal. A notable variety of applications are related to this problem according to the sources emitting the sounds, such as human beings, music instruments, and manufacturing machines. In analogy to leaning from images, it is necessary to extract high-level conceptual features from signals to facilitate learning process. The orthodox transformation methods applied on signal include Time Fourier Transform, Wavelets, Dual Tree Wavelets, moving average, Hartley Transforms, Hilbert-Huang Transforms, Linear Predictive Coding Coefficients, and so forth (Jorgensen, Lee et al. 2003). The details of these transformation methods are out of scope of this review, and it is only focused on various schemes of integrating these transformation methods in an automated fashion for different applications.

The most straightforward and commonly used scheme is “bag of frames” (West and Cox 2004), on which many previous works are based. Essentially, acoustic signal is
partitioned into a sequence of consecutive, possibly overlapping frames, where short-term features are independently extracted. Short-term features with respect to all frames are then aggregated to create a set of new features. The dominant downsides of “bag of frames” approach are the isolation from domain knowledge and its incompatibility with all of performance boosting techniques such as feature selection and parameter tuning (Pachet and Roy 2009). So its learning performance will be significantly compromized when faced with difficult problems.

There exist a huge volume of transformation methods dealing with acoustic signals in different domains. As handling acoustic signal is an especially domain-specific problem, in the majority of studies, choosing transformation methods is driven or partially driven by domain knowledge. A distinctive application (Omid, Mahmoudi et al. 2009) is to sort pistachio nut based on the sounds made by sliding nuts impacting a steel plate. A pistachio nut can be classified as split-shell or closed-shell. The feature construction is conducted manually by applying FFT to convert the original time-series signal to 1024-point Discrete Fourier Transform (DFT). Three groups of features, namely, magnitude, phase angle, and Power Spectral Density (PSD) of each data sample can be readily extracted from DFT. These constructed features subsequently undergo feature extraction with the application of PCA, which reduces more than 98% dimension. In this example, only three aspects (magnitude, phase angle, and PSD) of produced acoustic signal are considered and assumed to sufficiently distinguish sources of two types.

A paradigm of integrating domain knowledge and data-driven search in feature space is firstly extracting primitive domain-specific features from raw signals and then
feeding them to a heuristic-based search mechanism to construct features with higher discriminative power. Normally the heuristic-based search is realized by evolutionary algorithm. Two works (Guo, Jack et al. 2005, Schuller, Reiter et al. 2006) employing this paradigm are stated in detail.

The work of (Guo, Jack et al. 2005) investigates the acoustic signals from rotating bearings to identify fault bearings. It is devoted to extract discriminative features from the raw vibration signals collected from bearing rotation. The extracted features are subsequently applied on machine learning algorithms to predict bearing conditions. An original time-series signal can be converted to a $N$-vector in which each element value is the signal magnitude at a given time. Thus, the dimensions can be normally arbitrarily large in order to approximate original signals. The method transferring raw data to single scalar value (primitive features) is termed terminator. In this approach, terminator is computing the estimate of four statistical moments. The $k$th-order moment is defined as:

$$m_{x}^{(k)} = \frac{1}{N} \sum_{i=1}^{N} x_i^k,$$

where $x_i$ is the $i$th data point for a sample, and $k = 1, 2, 3, \text{ and } 4$. These four features are used as primitive features in the subsequent GP algorithm. The operators used include mathematical, logical, and probabilistic operators. Fisher criterion is adopted as fitness function to evaluate each individual. This approach is compared to some manually developing features which contain conventional measurements of shock pulse, crest factor, and kurtosis; high and low filtering; signal difference and sums; and normalization. For instance, kurtosis measurement is defined as:

$$kurt = \frac{1}{N} \sum_{j=1}^{N} \left( \frac{x_j - \bar{x}}{\sigma} \right)^4,$$

where $\bar{x}$ is the point average, and $\sigma$ is their standard deviation. As expected, the features generated by tree-based GP are very robust and improve the classification performance.
The author also points out a promising avenue of future research, that is, employing more terminators and operators to expand search space.

The work of (Schuller, Reiter et al. 2006) applies this paradigm in the speech emotion recognition problem. In this first phase, the primitive features are generated by applying functional derivations on a set of low-level contours. These base contours include such manually calculated measurements as elongation, intensity, MFCC coefficients, spectral centroid, and so forth. Each base contour reflects a specific aspect of a speech. For instance, contours of elongation, intensity and intonation are representative of prosodic information of a speech. Note that these base contours are obtained by applying transformation methods such as Hamming window and FFT. As demonstrated by (Schuller, Rigoll et al. 2003), derived functionals boost classification performance in contrast to base contours, thus they are used as primitive features for subsequent genetic algorithm process. The descriptive statistics applied to derive functionals are the linear momentums of first four orders of speed and acceleration coefficients of all base contours. A part of functionals are selected based on domain-knowledge. In this respect, constructing and selecting primitive feature set is dominated by domain knowledge. In the following implementation of genetic algorithm, primitive features serve as operands; and alteration methods (reciprocal value, addition, subtraction, multiplication and division) serve as operators. In each evolved generation, a set of operators are randomly selected and applied to a chromosome that denotes a subset of current feature set. A new feature is generated and added to the current feature set to be involved in next evolution.

One can see that the effectiveness of whole process of the feature construction paradigm stated above is dominated by the discriminative ability of primitive features
created based on domain knowledge, since the following heuristic search is totally based on the primitive ones. The main drawback of this paradigm is that a great deal of information may be missed in the process of converting raw signal to primitive features if methods are not chosen appropriately by experts, and they are in no way recovered in heuristic search. In order to exploit the potential useful information to the greatest extent, one can feed raw signal to heuristic search directly so that the search and selection of transform methods applied on raw signals is driven by both knowledge and data.

According to this idea, a study (Mierswa and Morik 2005) presents a unifying framework of feature construction for time-series audio data. A repository of abundant elementary extraction operators is adopted to extensively explore useful feature sets in the large search space. The repository encompasses basis transformations, filters, markup of intervals, generalized windowing, and functions. Basis transformations map the data from a given space into another space like frequency space, correlation space, and phase space. For example, FFT maps the data from time space into frequency space. Filters transform elements of a series to another location within the same space. Examples include moving average, exponential smoothing and frequency passes. Note that many transformations often couple with weighting functions such as Hanning window, Blackman-Harris window, and linear functions. Detection of appropriate markup intervals of signals can be realized by clustering. Windowing transformation shifts a window with a step size and a certain function within the window is calculated. Functions are defined as the transformations converting a series into another series, which includes normal statistics like average, variance, and standard deviation. Tree-based genetic programming (GP) is used to guide the search in operator space and determine the
representation of constructed features. The fitness functions are the performance of classifiers applied on the data which are resulted from mapping original data to generated features. This framework is distinctive in that a wide array of transformations are considered, which includes the ones converting raw signals to scalar values. The heuristic approach used for searching in operator space guarantees that automatically constructed features are highly meaningful and domain-specific to accommodate different problems.

Extraction Discovery System (EDS) (Pachet and Zils 2004, Pachet and Roy 2009) is another unifying framework of integrating data-driven search and domain knowledge for feature construction. Apart from the automated fashion realized by genetic programming, the cores of the EDS are the predefined typing rules and heuristics which introduce relevant domain knowledge and expert opinions for guiding search of optimal features. In the manner of genetic programming, EDS constructs high-level discriminative features by automatically applying operators on acoustic signals. The operators include a variety of basic transformation methods including mathematical operators, signal processing operators, temporal such as correlation, spectral such as FF, and split. Each constructed feature can be represented by a tree with multiple operators in the nodes. Since each operator is only compatible with the inputs of certain value types, value types and typing rules are first defined. Data type is comprised of atomic value type, function, and vector. The atomic value types are defined in terms of physical dimensions, like frequency (f), time (t), and amplitude (a). Functions are defined as objects mapping one dimension to another. For example, an acoustic signal which is represented as time to amplitude is denoted by function of “t:a”. Analogously, the a spectrum represented as frequency to amplitude is defined as “f:a”. Vector associates an
index to a value and its symbol is “V”. For instance, the data type of a signal split into frames is “Vt:a”. The typing rules pose constraints on functions of operators and inputs of different value types, in order to guarantee the syntactical correctness. For example, given a input signal, a typing rule can be given as 
\[ \text{Type}(\text{Min}(\text{Max}(\sqrt{\text{Split}(\text{FFT}(\text{Split (SIGNAL, 3, 100)), 2, 100)))), 2, 100))) = "a" . \]
Lastly, function patterns can be defined in terms of data type, typing rules, and three generic operators, namely, “*”, “!” , and “?” . An example in (Pachet and Zils 2004) is presented for illustration. The pattern \( ? \_ a (\_ V a (\_ \_ t: a (SIGNAL))) \) stands for 1) apply some signal transformations in the temporal domain (\( \_ t: a \)); 2) split the resulting signal into frames (\( \text{Split} \)); 3) find a vector of characteristic values -1 for each frame (\( \_ V a \)); and 4) \( ? \_ a \) `find one operation to find one relevant characteristic value for the entire signal (\( ? \_ a \)). Under the constraints defined by function patterns, the construction process is iterated in GP manner. Additionally, some domain-specific heuristics are taken into account to facilitate quick construction of “interesting” features. For instance, functions containing useless operations (like too many repetitions of an operator) are normally avoided. Other mechanisms like rewriting rules and caching are designed to further speed up the learning and reduce computation cost. The experiment showed that this system is capable of detecting a Sinus wave that is mixed with a powerful colored noise in another frequency range. This type of complex patterns cannot be discovered by linear combination of basic features.

2.2.2.3 Text

Extracting features from texts is specifically for the text categorization problem. Currently, the majority of text categorization systems are employing “bag of word”
(BOW) paradigm (Gabrilovich and Markovitch 2007). This orthodox paradigm extracts features in terms of unordered words or phrases to describe semantics of documents. Each document is steadily mapped to a structured format characterized by these extracted features, from which statistical models can be easily learned. Given a collection of documents, feature construction process is named document indexing, and generated features are termed term weights. Therefore, a document is represented as a vector of term weights \( d_j = <w_{1j}, w_{2j}, ..., w_{|T|j}> \), where \( T \) is the set of terms that occur in at least one document (Sebastiani 2002). The most common representation of the term weight is frequency distribution. It takes the times of occurrence of term \( t \) in a document as term weights. This representation can be simplified to binary weight where 1 and 0 indicates, respectively, presence and absence of term \( t \) in a document. Frequency distribution is also extended to a widely used variant, namely term frequency-inverse document frequency (tf-idf) (Salton and Buckley 1988). Tf-idf is defined as \( \text{idf} = tf_{t,d} \times \log \frac{N}{df_t} \), where \( tf_{t,d} \) is the frequency of term \( t \) appearing in document \( d \), \( df_t \) is the number of documents containing term \( t \), and \( N \) is number of documents. It reflects an intuition that the importance of a term with respect to a document is also impacted by document frequency: the more documents a term occur in, the less discriminative it is. Additionally, a number of other representations have been proposed and explored. For instance, some researchers (Fuhr and Buckley 1991) propose to use phrases, instead of individual words, as terms. (Wibowo and Williams 2002) recommends to only take into consideration words at the beginning of the document.

Although “bag of word” approach has achieved quite a few successes, its inherent limitation of having no access to domain knowledge hinders its effective application on
some difficult problems. For example, in the case that some words conveying highly
discriminative information do not appear or appear only once in the training collections,
their useful information will be ignored and underestimated by BOW approach. This
observation motivates some studies of exploring new feature space with the assist of
common or domain knowledge.

For a certain domain, domain-specific features can be explicitly constructed and
demonstrated. (Lan, Tan et al. 2009) proposes and compares various feature generation
schemes regarding protein-protein interaction (PPI) classification. Given a corpus data
that is a collection of document abstracts, the task is to train a classifier from the training
corpus and classify an unseen document as PPI relevant or irrelevant. Four feature
generation paradigms are presented, namely bag-of-word (term weighted as binary or
tf.rf), trigger keywords, protein named entities (PNEs), and interact-PNE. Except for bag-
of-word, the other three are all domain-specific approaches. Trigger keywords are
selected by biological domain experts to indicate an interaction between protein entities.
PNEs are the biological entities like gene, protein and virus, and their names are extracted
by a PNEs recognition system. The use of PNEs is assumed to complement the
information loss in the general bag-of-words representation. Three PNEs-related features
are adopted, and they represent respectively 1) if the document has at least one PNE, 2) if
the document has at least two PNE, and 3) if the document has more than two PNEs.
Lastly, three interact-PNE-related features are constructed in terms of the pairwise
occurrence of interactive indicators and PNE on a single sentence. The combinations of
these feature representations are compared by applying linear SVM on a competition
corpus, and the performance is evaluated by precision, recall, $F_1$, and accuracy. It is
observed that the combination of bag-of-word (tf.rf weighted) and interact-PNE outperforms other counterparts. It is concluded that interaction information on the sentence level contributes substantially to the predictive power. In addition, tf.rf weight method performs better than binary method in this context.

(Gabrilovich and Markovitch 2005) proposes a unifying framework which incorporates the common-sense and domain-specific knowledge into the bag of word (BOW) to create new knowledge-based features. The knowledge base can be built from abundant resources regarding most domains on the Internet, so this approach can be applicable to various domains. In this study, author chooses Open Directory Project (ODP) as the knowledge base that is composed of a machine-readable hierarchy of 600,000 categories (termed concepts in this scenario) over 4,000,000 websites. The contextual descriptions and content of corresponding website constitute the training data for each concept. The most characteristic words are also identified to describe a concept. In this configuration, nearest neighbor algorithm is employed to compare the segment of text with all concepts, and best-matching ones are selected as new features for this text segment. This feature construction methodology is assessed on four test collections, and it shows a significant improvement of categorization performance.

2.2.2.4 Biology Sequence

DNA, RNA and protein sequence represent the spectrum of biomedical data. Each one of them is made up of a sequence of ordered sites. For DNA and RNA sequence, each individual site represents one of four nucleotide bases of a DNA strand, denoted by A, C, G, and T; whereas the sites in a protein sequence are 20 amino acids. In contrast to texts, biology sequences have limited number of “terms”, and the orders of sites bear
significant meanings in conveying biological information. From Scopus search result, three methodologies of extracting features from DNA and protein sequences are selected and briefly stated below.

For the problem of predicting translation initiation sites (TIS) in mRNA and cDNA, various machine learning algorithms have been proposed and used, among them is the study (Zeng, Yap et al. 2002) which uses feature construction and feature selection to create features applicable to general purpose models. In the DNA case, TIS is ATG sites, so the task is to predict if a ATG site is a true TIS. A raw sequence is typically defined by a window centered at each ATG with an upstream and a downstream of 100 bases long. Each base takes on value of A, T, G, or C. The first group of constructed features is called frequency-grams, denoted as K-grams (k = 1, 2, 3, 4 and 5). Since upstream and downstream k-grams are distinguished, it gives total $2 \times 4^k$ features for this group. The frequency of each k-gram occurring in raw sequence is the value of corresponding feature. The second group of constructed feature is in-frame frequency-grams. The k-gram (k = 3, 4) is aligned with the 3-gram and the histogram window extends downstream for (k-3) positions. The third group is Stop-codon frequencies. It is defined as the frequencies of the in-frame stop codons (i.e. TAA, TAG, TGA) within 50 bases downstream and within 100 bases downstream from the ATG. The constructed features are subsequently selected by correlation based feature selection (CFS) (Hall 1999). Then four general models (SVM, Naïve Bayes, neural network, and decision tree) are applied to assess the discriminative power. This strategy shows favorable performance to those involving special configuration and engineering on predictive model.
Another study of predicting translation initiation sites is using, instead of DNA patterns, amino acid patterns (Liu, Han et al. 2004). For the case of amino acid sequence, \( k \)-gram (\( k = 1, 2 \)) amino acid patterns are used. Each position may take on 20 standard amino acid letters and a special stop symbol, so a \( k \)-gram pattern leads to \( 2 \times 21^k \) features. Additionally, three boolean features “DOWN4-G”, “UP3-AorG” and “UP-ATG” are also added, corresponding to three knowledge in terms of identifying TIS from previous works. For instance, the feature “UP3-AorG” is inspired by the finding that A or G tends to be found 3 nucleotides upstream of a true TIS.

In (Vucetic, Radivojac et al. 2001), 23 features are created from protein sequence to predict protein disorder. For each position \( i \) within a sequence and an amino acid \( a \) the feature \( f_{ia} \) is defined as the fraction of \( a \) within a widow of fixed length. In this way, 20 features corresponding to 20 amino acids are constructed. The other two features are based on, respectively, the assumptions that the flexibility of amino acids and sequence complexity are significantly related to disorder.

Compared to unstructured data of other types, biology sequence is much simpler in that the number of basic elements is small, so no sophisticated search mechanism is necessary. However, there are abundant biology assumptions and knowledge available, based on which other features specific to a given problem can be extracted.

2.2.3 Feature construction on semi-structured data

Compared to structured data and unstructured data, semi-structured data is a much more under-researched field in terms of feature construction. Since no paper describing feature construction methodologies on semi-structured data is found in the top 200 cited papers in Scopus search result, new Scopus searches need to be conducted. First, a query
is made which combines the previous query of feature construction with the names of some types of semi-structured data such as websites, medical claim, firm financial statement, credit bureau report, credit card transaction, and user generated contents like Facebook. In the result, few relevant papers are returned. To loose the query restriction, a new query is conducted which combines the names of semi-structured data with the keywords related to concept learning such as data mining, machine learning, statistical, prediction, and classification. In the similar manner, the most cited papers and their reference papers are selectively examined. The most representative methodologies of constructing features on semi-structured data are presented below. Additionally, a patent describing a transaction spider that automatically extracts features from transactions is also introduced in detail.

Web pages are generally organized in such human-readable structure as comprising of full-text, title/heading, image, hyperlink, and so forth. To convert them into computer-readable formats, many efforts have been made to develop information extraction systems, an example being the web spider built by the WebKB group (Craven, DiPasquale et al. 2000). In the WebKB, user firstly defines features associated with web pages. Those features belong to either classes (e.g. department, faculty, student, and project) or relations (e.g. advisor_of_, and course_taught_by). After being trained on a set of labeled web pages, the web spider is capable of extracting values of defined features from web pages of similar type. The knowledge bases is enriched and populated by the extracted content, and then can be used for computer inference. Other efforts are also made to extend work of WebKB with applications of mining web pages of retails (Ghani and Fano 2002) and companies (Ghani, Jones et al. 2000). It is worthwhile to note that all
the features are defined beforehand based on domain knowledge. For example, the wrapper features in (Ghani, Jones et al. 2000) includes revenue (revenue data for up to the last 10 years), net-profit (net profit data for up to the last 10 years) and same-state (companies in the same state as this company). These features are subsequently used to discover association rules.

Most of features extracted from medical claims data for computer inference are based on combination of clinical judgment and empirical evidence (Kolodner, Lipton et al. 2004), and no autonomous constructive method is found in the literature. Being knowledge-based, constructed features in various studies differ substantially since different domain-specific considerations and empirical evidences are taken into account. For example, to improve the prediction performance of medical expenditures, a study (Huber, Schneeweiss et al. 2013) assumed that the total health care costs are related to such factors as sex, age, language area, managed care, deductible, accident coverage, and CDS (chronic disease score). On the other hand, 9 features are assumed to potentially influence patient medication utilization, namely number of claims, region, sex, age, total amount charged to the patient, total amount paid by the patient, enrollment continuity, medication types, and comorbidity (Huang, Wulsin et al. 2009, Huang, Li et al. 2011). The features used to discovery temporal behavioral patterns are total benefits paid within a certain time window (Tsoi, Zhang et al. 2005). They are specifically defined as: 

\[ y_t = \sum_{i=t}^{t+W-1} z_i \quad t = 1, 2, ..., D - W + 1, \] 

where \( D \) is 365 (one year period), \( W \) is the size of time window, and \( z_i \) denotes the benefits paid to a claimant on the \( i \)th day. Every profile can be characterized by \( y_1, y_2, ..., y_{D-W+1} \). (Liou, Tang et al. 2008) employs only 9 highly conceptual, cost-related features in pursuit of detecting claim fraud. The partial
list of features is: average days of drug dispense, average diagnosis fees, average medical expenditure per day, and average amount claimed. Another study (Wan and Shasky 2012) whose aim is identifying care patterns and detecting bill fraud uses 12 hospital characteristics, market competition, and area characteristics as features to feed to exploratory statistical methods. Examples of hospital characteristics include average length stay, bed size, total inpatient days, and Medicare case mix index.

In the scenario of generating meaningful financial features, (Kirkos, Spathis et al. 2007) complies 27 financial ratios from firm financial statements and describes the details of these features. Each feature is derived based on prior studies concerning one aspect associated to fraudulent falsification. For instance, continuing growth is regarded as an important motivation for fraud as companies may engage in fraudulent activities to keep similar growth trend as past (Stice, Albrect et al. 1991), Sales Growth Ratio is derived to measure growth.

From credit card transactions, (Sahin, Bulkan et al. 2013) derived features of 5 types: all transactions statistics, regional statistics, merchant type statistics, time-based amount statistics and time-based number of transactions statistics. The full list of features and the derivation approach are not given due to privacy and confidentiality concern. These statistics jointly features the usage profile of one credit card, and a cost-sensitive decision tree is applied on these features to classify transactions as normal or fraudulent. The credit transactions are also used for credit scoring which supports decision making for approval of a credit card application. Throughout paper search, only a few of relevant datasets are found for credit scoring, and only limited number of features (ranging from 14 to 38) are contained. The German and Australia datasets from UCI Machine Learning
Repository ([http://archive.ics.uci.edu/ml/](http://archive.ics.uci.edu/ml/)) are mostly employed (Hsieh 2005, Huang, Chen et al. 2007, Wang and Huang 2009, Chen and Li 2010) for exploration of effective credit scoring engine. Some applications on other datasets like UCSD (Marqués, García et al. 2012), Iranian (Marqués, García et al. 2012), and Turkish credit card data (Akkoç 2012) are found, and all the efforts are still devoted to predictive models with little discussion of feature derivation from transactions.

The patent of Data Spiders [Data spiders] is the only document we find that discusses data-driven approach of feature construction from event-centric data. Specifically, this approach is jointly driven by data and domain knowledge in the sense that transaction variable templates are built based on domain knowledge and the subsequent search is guided by intrinsic characteristics of dataset with implementation of genetic algorithm (GA). An example of template can be expressed in following format: # event = A in time = B based on rollup C. A, B, and C are parameters coming from pre-defined value sets, and a combination of parameters on a template constructs a feature. A notable variety of templates and possible values of parameters lead to prohibitively enormous number of features, which requires heuristic search for choosing discriminative feature set of acceptable size. Data Spider employs genetic algorithm, subjected to Naïve Bayes score ranking, to guide the heuristic search. Each group of features is regarded as a chromosome, and chromosomes are initialized by randomly choosing template parameters at the beginning of GA. The feature groups evolve in loop fashion: 1) fitness of each group is assigned by Naïve Bayes score ranking; 2) children are created and mutated from parents with highest fitness; 3) a new generation is created by merging top parents and children; 4) assign fitness to each group in new generation; 5) and check if
the stop condition is satisfied, if not, go to step 2. Since it is a patent, no experiment is conducted to examine the effectiveness of this approach, and no discussion of compiling templates for different types of data, or the reason of choosing GA, Naïve Bayes is made. This idea sheds light of one direction toward realizing automatic data-driven feature construction from semi-structured data.

In summary, the majority works of learning patterns and concepts from semi-structured data is focused on the learning process rather than feature creation. In the applications on event-centric data, typically a small number of features are generated by human experts based on domain knowledge and subjective judgments. Even in the same field, different data and different considerations possibly lead to totally disparate features. One can see that only a small fraction of information will be retained if other impact factors are overlooked by experts. Hence methods of exploring features in an automatic and thorough manner merit in-depth investigation by researchers. The patent of Spider is the only methodology found to conduct automatic feature construction on transactions.

2.3 Data-driven V.S. knowledge-based

The process of supervised learning can be generally divided into a sequence of three stages – representation (feature construction), feature selection/extraction, and prediction (Kittler 1975). In general, the approaches used in these 3 stages distribute gradually from problem-based and expert-involved to data-driven and computer-automated. Representing a raw data in a meaningful way with respect to a specific application often necessitates taking domain knowledge into consideration, while prediction can be conducted automatically by computers on well-structured data. This pattern is also manifested in application of object detection (Lillywhite, Lee et al. 2013).
Based on the literature presented above, one can see that quite a few of studies have tried to bring automatic data-driven approaches to the first stage – data representation. Moreover, we are presented an evident pattern that features can be constructed from structured data and unstructured data by a wide variety of data-driven approaches, whereas few data-driven methods are applied on semi-structured data. Similar to structured data, the distribution of values in unstructured data is homogeneous in the sense that no hierarchical or grammatical relationship is present. This homogeneity eases the direct use of data-driven constructive methods without any expert intervention. In contrary, the non-relational data models in semi-structured data commonly imply some form of hierarchical or grammatical relationship, and this calls for involvement of domain knowledge. However, knowledge-based approaches are slow, labor intensive, and most importantly, only capable of generating so few features as a substantial part of useful information may be missed. In light of these observations, we realize that a method integrating both domain-knowledge and data-driven is essential for the objective of exploiting nugget from semi-structured data to the greatest extent. Also, event-centric data coming from various sources (credit report, medical claim, retailer data) have similar structure (event-centric) and common characteristics, so the new developed methods can be possibly used on this field. The data spider [Data spiders] is a good reference and illustrative example manifesting the way of integrating domain-knowledge and data-driven and the application on event-centric data. The compilation of templates and possible values for each element are driven by domain knowledge, and searching for optimal set of features is automatically realized merely based on data inherent characteristics. After pondering these findings and examples, we propose a feature
construction engine serving as an interface between domain experts and data, and can readily apply to event-centric data coming from various sources.
3 Methodology Proposal

3.1 Ontology constitution

In this study, an ontology-based feature construction framework is proposed with the aim of constructing meaningful and interpretable features from semi-structured data by interactively integrating domain knowledge, expert judgments, and data-conveyed information. The constructed features are representative of general domain knowledge and expert hypotheses. The framework not only facilitates the discovery of new knowledge and hypotheses, but also presents knowledge in an understandable and succinct way. Under this framework, ontology is employed to describe the hierarchical and procedural structures underlining a semi-structured data. A semi-structured medical claim data is used as an example for illustration (some variables are shown in Table 3-1).

Table 3-1: Variables in a medical claim data

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAT_ID</td>
<td>Patient ID</td>
<td>PROC_CDE</td>
<td>Procedure code</td>
</tr>
<tr>
<td>MEM_SEX</td>
<td>Member sex</td>
<td>NDC</td>
<td>National drug code</td>
</tr>
<tr>
<td>MEM_AGE</td>
<td>Patient age at date of service</td>
<td>DAYSSUP</td>
<td>Days supply</td>
</tr>
<tr>
<td>REGION</td>
<td>Region</td>
<td>QUAN</td>
<td>Quantity dispensed</td>
</tr>
<tr>
<td>PROV_ID</td>
<td>Provider ID</td>
<td>PROD_TYPE</td>
<td>Product type</td>
</tr>
<tr>
<td>POS</td>
<td>Place of services</td>
<td>PAY_TYPE</td>
<td>Payor type</td>
</tr>
<tr>
<td>FROM_DT</td>
<td>Service begin date</td>
<td>CHARGE</td>
<td>Segment Charge Amount</td>
</tr>
<tr>
<td>TO_DT</td>
<td>Final date of service delivery</td>
<td>PAID</td>
<td>Segment Paid Amount</td>
</tr>
<tr>
<td>DIAG1</td>
<td>ICD-9 diagnosis code</td>
<td>RECTYPE</td>
<td>ETG record type</td>
</tr>
<tr>
<td>DIAG2</td>
<td></td>
<td>NDC</td>
<td>National drug code</td>
</tr>
<tr>
<td>DIAG3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIAG4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Ontology in this framework is essentially comprised of 3 elements: class, relation, and property.
Class is a group of individuals sharing same characteristics. In the example of medical claim data, Patient and Provider are two classes, and John can be an instance of class Patient and Cincinnati Children’s hospital can be an instance of Provider.

Property is a characteristic of a class. For example, age and sex are two properties of class Patient. The data type of a property is categorized into nominal, ordinal, ratio, and interval. Table 3-2 lists the classes and the corresponding properties for the medical claim data example.

Relation is a link between classes, which stands for how classes are related. The classes that a relation can directs from are defined as from-domain, and the classes that a relation can directs to are defined as to-domain. The group of instances in the same domain is called instance cohort, and each domain has cardinality bound that is the bound of the number of instance a domain has. Table 3-3 lists the relations, the corresponding from-domains, to-domains, and cardinality bounds of the medical claim data. In this example, the relation MakeDiagnosis has from-domains of Patient and Provider, and to-domain of Diagnosis. The 4 diagnoses made by a provider for a patient is an instance cohort of domain Diagnosis, and the cardinality bound for this to-domain is [1, 4].

Table 3-2: Classes and properties in the medical claim data example

<table>
<thead>
<tr>
<th>Class</th>
<th>Property</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient</td>
<td>sex</td>
<td>nominal</td>
</tr>
<tr>
<td></td>
<td>age</td>
<td>ratio</td>
</tr>
<tr>
<td></td>
<td>region</td>
<td>nominal</td>
</tr>
<tr>
<td>Provider</td>
<td>region</td>
<td>nominal</td>
</tr>
<tr>
<td></td>
<td>expertise</td>
<td>nominal</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>diagnosis_code</td>
<td>nominal</td>
</tr>
<tr>
<td>Procedure</td>
<td>procedure_code</td>
<td>nominal</td>
</tr>
<tr>
<td>Hospitalization</td>
<td>from_date</td>
<td>interval</td>
</tr>
</tbody>
</table>
Table 3-3: Relations and domains in the medical claim data example

<table>
<thead>
<tr>
<th>Relation</th>
<th>Domain</th>
<th>Class</th>
<th>Cardinality limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>MakeDiagnosis</td>
<td>from-domain 1</td>
<td>Patient</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>from-domain 2</td>
<td>Provider</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>to-domain</td>
<td>Diagnosis</td>
<td>[1, 4]</td>
</tr>
<tr>
<td>MakeProcedure</td>
<td>from-domain</td>
<td>Diagnosis</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>to-domain 1</td>
<td>Procedure</td>
<td>[1, +]</td>
</tr>
<tr>
<td>PrescribeDrug</td>
<td>from-domain</td>
<td>Procedure</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>to-domain</td>
<td>Drug</td>
<td>[1, +]</td>
</tr>
<tr>
<td>Hospitalization</td>
<td>from-domain</td>
<td>Procedure</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>to-domain</td>
<td>inpatient</td>
<td>[1, +]</td>
</tr>
</tbody>
</table>

The definition of from-domains and to-domain sets constraints on the classes on which a relation can be applied, so it serves to guarantee the legitimacy of the ontology. Domain constraint has the transitive property: if a relation $A$ has to-domain $X$, and from-domain $Y$, and a relation $B$ has to-domain $Y$ and from-domain $Z$, then relation $A$ can also directs from domain $Z$ and relation $B$ can direct to domain $X$ (as shown in Figure 3-1). For example, relation PrescribeDrug has from-domain of Procedure, and the relation MakeProcedure directs from Diagnosis and Procedure, so relation PrescribeDrug can also directs from Diagnosis without Procedure as intermediary. This property significantly improves ontology flexibility, and facilitates the ontology reconstruction which will be introduced in the next section.
Under the framework of ontology constitution, an initial ontology can be defined based on general knowledge. This is essentially designed by domain experts for a specific objective. The initial ontology in Figure 3-2 is describing medical services that a patient received for a period of time, which includes the providers he/she visited, the diagnoses made, the drugs provided, and so forth. The objective for this example is assessing the relapse risk of depression for a patient.
3.2 3-level ontology-based features

Feature construction is realized by evolving ontology, which is initiated by domain hypotheses and evaluated in data-driven fashion. Ontology evolution has the following 3 levels in descending order of granularity:

Property level: ontology is modified by adding new properties of a class. The constructed properties are considered as new features for the class. Examples include: the length of days for an inpatient: days=to_date-from_date; quantity of drugs per day: quant_day=quantity/days. Property features are the most granular features, and are mostly driven by domain knowledge since normally no complex logic is applied.

Cohort level: Properties are created by applying simple logic or statistics on an instance cohort. For example, the drug a and drug b comprise the drug cohort for the class procedure 1-1. Same statistics (e.g. max, min, and mean) can be applied on
properties (e.g. days, and quant_day) of drug cohort. These constructed cohort features are considered as new properties of the class of from-domain. For example, max(quant_day_1, ..., quant_day_n) is a new property for class Procedure. Note that quant_day is a property feature constructed for class drug.

Relation level: features are constructed across classes linked by relations. Features constructed on relation level have much higher complexity and normally convey more discriminative information, so it is the primary focus of this study. The construction is typically driven by hypotheses made by domain experts, and it integrates domain knowledge and data-embedded information in an interactive way.

The following hypothesis is used to demonstrate the construction of relation level features: “if a female has taken antidepressants belonging to SSRIs or SNRIs within the last 1 year, the older she is, the higher relapse risk of depression she suffers”. The hypothesis can be completely represented under the ontology in Figure 3-2, and further formulized as:

Patient.age | {Patient.sex IN [female], ∃i, Drug_i, class IN [SSRIs, SNRIs], Drug_i.time > now – 1 year}

The formulas within the braces stand for the conditions that the hypothesis is based on, and the formula outside the braces is the evaluation function that is hypothesized to be related to the objective. In this example, the objective is assessing the relapse risk of depression; and the evaluation function is a patient’s age. A new feature can be constructed by assigning evaluation function (Patient.age in this example) and missing value to, respectively, samples satisfying the hypothesis condition and the ones not satisfying. The usefulness of the hypothesis can be gauged by evaluating the new feature based on two criteria: support and confidence. Support and confidence are two concepts widely employed in association rules application. In this framework, support is
defined as the proportion of samples satisfying the hypothesis condition, which assesses the generality of the hypothesis. Confidence is defined as the fitness of the hypothesis against the data. It is represented as, on the samples satisfying hypothesis condition, the discriminative power of the new feature in term of supporting the objective. Thus, confidence can be measured as the correlation between the new feature and a response variable built based on the objective. A wide array of univariate feature evaluation algorithms (e.g. AUC, information gain, and t-test) can be employed based on the data types of the feature and response variable. In this example, a binary response variable can be built indicating if a patient has depression relapse. The new feature is the age of a patient, which is discrete numeric value. T-test can be applied to evaluate the correlation between the response variable and the new feature, and p-value is adopted as confidence score.

Normally generality and fitness have contradictory effect to each other: lower generality more likely leads to higher fitness against data, and vice versa. To evaluate the overall performance of a hypothesis, two weights can be assigned to the support score and confidence score by domain experts based on the objective and preference. The criterion evaluating the overall usefulness of a hypothesis is given as:

$$C = w_1 \times Score_{support} + w_2 \times Score_{confidence}$$  \hspace{1cm} (3-1)$$

It is used as the fitness function for heuristic optimization algorithms that will be introduced in the next section.

Under the ontology framework, the features of 3 levels can be readily constructed and evaluated based on domain knowledge. Moreover, this framework has two important functionalities: new hypotheses exploration and ontology reconstruction.
3.3 New hypotheses exploration

Under the ontology framework, not only a hypothesis proposed by domain experts can be evaluated in data-driven fashion, but also new hypotheses can be derived, assessed, and compared in pursuit of exploring better hypotheses. A new hypothesis can be derived by changing the slots of the hypothesis condition, and this process is named as hypothesis evolution. A slot in the hypothesis can be a property value (e.g., Patient.sex), a group of property values (e.g., Drug.class), or an operator (e.g. the operator in Drug.time constraint). The search space that defines the bounds of hypothesis evolution is set by domain experts. In this example, the drug class group [SSRIs, SNRIs] can be expanded to include more antidepressant class, or reduced to only one or no antidepressant class. Also, more time constraints of taking antidepressant can be explored. The defined search space encompasses a wide variety of candidate hypotheses, and in most cases the enormous amount of hypotheses cannot be examined in an exhaustive way. This necessitates the use of heuristic search algorithms that substantially reduces computation burden at expense of comprising the possibility of find optimal candidate. Genetic algorithm (GA) and particle swarm optimization (PSO) are two of most widely used heuristic optimization methods. They share the following common points:

1. random initialization on population
2. fitness function based candidate evaluation
3. population reproduction
4. iterative process until stop criterion is met
3.3.1 Genetic algorithm introduction

In light of the similarity of these algorithms, genetic algorithm is taken as an example for illustration, and the procedure can be readily extended to other similar heuristic optimization algorithms. The workflow of genetic algorithm is shown in Figure 3-3.

![Figure 3-3: Genetic algorithm (GA) workflow](image)

Suppose a targeted problem is to identify the optimal combination of a string of binary bits according to a defined fitness function. In this illustration example, a chromosome is comprised of 10 binary bits, so the search space is $2^{10}=1024$.

1) Initial population creation.
As shown in Figure 3-4, an initial population comprised of 5 chromosomes are created, and each one is assigned a corresponding fitness score. In this example, higher score is indicative of better performance.

2) Parent selection

Two parents are selected based on the fitness score, and the chromosome with higher score has higher probability of being selected. In Figure 3-4, the top two with fitness score of 0.85 and 0.8 are selected and are fed to the subsequent processes.

3) Crossover

As shown in Figure 3-5, a crossover site is randomly selected which separates each parent into left group and right group of strings. Two parts of string of the two parents are exchanged, and it results in two new offspring whose strings inherit from both two parents. These two new offspring have fitness score of 0.88 and 0.78. This mechanism is aimed to further explore better combinations from good performers.
4) Mutation

As shown in Figure 3-6, with some low probability, a portion of the new chromosomes will have some of their bits flipped. Its purpose is to maintain diversity within the population and inhibit premature convergence. In the search space, one can see that mutation alone induces a random walk. In this example, after mutation, the resulting two new chromosomes have increased and decreased fitness score.

All of new offspring resulting from crossover and mutation are added to the whole population, where another iteration including parent selection, crossover, and mutation is applied until the defined stopping criterion is met.
3.3.2 Genetic algorithm application on hypothesis evolution

To apply genetic algorithm on hypothesis evolution, a specification on chromosome is made to accommodate the constrains of the hypothesis evolution. As shown in Figure 3-7, a chromosome encompasses a series of gene groups (highlighted by different colors), and each group is comprised of multiple binary encoded genes. A gene group stands for a slot in the hypothesis that can take on different values. Each binary encoded gene represents a value that the corresponding slot can be assigned. Each gene group is characterized by cardinality, which is the bound on the number of values it can be assigned. For example, gene group Patient.sex can take 1 (female or male) or 2 values (female and male); and Drug.time can only take on 1 value (one of 3 months, 6 months, and 12 months). Under this chromosome structure, each candidate hypothesis under the ontology framework can be represented as a chromosome.

<table>
<thead>
<tr>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>......</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>female</td>
<td>male</td>
<td>SSRIs</td>
<td>SNRIs</td>
<td>TCAs</td>
<td>MAOIs</td>
<td>......</td>
<td>&gt;</td>
<td>&lt;</td>
<td>12 months</td>
<td>6 months</td>
<td>3 months</td>
</tr>
<tr>
<td>Patient.sex</td>
<td>Drug.class</td>
<td>......</td>
<td>Drug.time (operator)</td>
<td>Drug.time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3-7: A chromosome representing a candidate hypothesis

With the chromosome specification, the standard genetic algorithm steps are implemented.

First, an initial population of chromosomes is generated randomly. Each chromosome needs to meet the cardinality requirement, and the overall usefulness is evaluated based on the criterion $C$. Second, two chromosomes are chosen as the parents in such a way that higher value of criterion $C$ leads to higher probability of being selected...
as parents. Next, a new generation is reproduced by applying operators of crossover and mutation. In crossover, the swapping locus can only be chosen on the boundaries of gene groups. Put differently, only gene groups, rather than individual genes, can be swapped across two parents. On the mutation step, the requirement of cardinality is involved: if a gene is mutated, other genes in the same gene group also need to be mutated if cardinality requirement is violated. For example, the gene group Drug.time has cardinality of 1, if a gene is mutated to 0, then another gene in this group needs to be selected and mutated to 1. Stop criterion is defined by domain experts, which can be the maximum iteration number or a threshold on criterion \( C \). In addition, domain experts can set a threshold on criterion \( C \), and the chromosomes with criterion \( C \) higher than the threshold are recorded. The recorded candidate hypotheses can be compared and investigated based on domain knowledge, and the final hypotheses are selected.

### 3.4 Ontology reconstruction

In some cases, the ontology structure needs to be modified to accommodate the hypotheses. On the new ontology structure, the hypotheses can be evaluated in the same way based on the criterion \( C \). Ontology reconstruction can be normally realized by removing classes or relations irrelevant to hypothesis or adding new classes and relations.

For example, a new antidepressant hypothesis is: if a female has taken antidepressants belonging to SSRIs or SNRIs when she was hospitalized, the older she is, the higher relapse risk of depression she suffers. This new hypothesis does not completely fit the initial ontology because the initial ontology has no hospitalization information. On one hand, to accommodate this new hypothesis, a new class Inpatient and corresponding relation Hospitalization are added (as shown in Figure 3-8).
Corresponding new feature for this hypothesis can be evaluated in the same fashion above. On the other hand, the hypothesis can be reduced to: if a female has taken antidepressants belonging to SSRIs or SNRIs, the older she is, the higher relapse risk of depression she suffers. The reduced hypothesis can also be evaluated by the same criterion in data-driven fashion. A comparison between the original hypothesis (based on new ontology) and reduced hypothesis (based on old ontology) can be made, and the domain expert can make a decision if his/her hypothesis is significantly more useful than reduce hypothesis. If his/her hypothesis is proved to be better than reduced hypothesis, the new ontology is chosen.

Ontology can also be simplified by removing classes and relations irrelevant to the hypothesis. The diagnosis information is not included in the above hypothesis, so we can simplify ontology by removing class Diagnosis and relation MakeDiagnosis. The simplified ontology is shown in Figure 3-9. The relation MakeProcedure directs from classes of Patient and Provider, and the legitimacy is guaranteed by the transitive property of relation. On the other hand, the hypothesis can be elevated based on the old ontology. For example, an elevated hypothesis can be: if a female was diagnosed as depression with insomnia, and has taken antidepressants belonging to SSRIs or SNRIs, the older she is, the higher relapse risk of depression she suffers. This elevated hypothesis can be evaluated on the initial ontology. In the same way, these two hypotheses can be compared. If a decision is made that the elevated hypothesis is better, the old ontology is kept and corresponding new features can be constructed.
Figure 3-8: Elevated ontology

Figure 3-9: Simplified ontology
4 Case Studies

4.1 Hospitalization prediction on miscellaneous medical claim data

Hospitalization accounts for a large portion of medical spending, and the spending on unnecessary hospital admission is estimated to be more than $30 billion in 2006. Given the limited medical resources and increasing demands, prospective hospitalization planning will go a long way to optimize the allocation of hospitalization resources and reduce unnecessary admission. In this sense, a great deal of efforts has been devoted to develop models to predict the hospital readmission. This case study focuses on employing the proposed feature construction methodology on the medical claim data to predict the length of stay in hospital for the patients who have received medical services in the previous years.

4.1.1 Data overview

The medical claim data is provided by Kaggle and Heritage Provider Network (HPN) in a public data science competition which lasted from April 2011 to April 2013 and rewarded a $3 million prize to the winning teams. The competition aims to inspire the development of innovative and breakthrough techniques of predicting hospitalization based on non-structured medical claim data.

After preliminary data cleaning and preprocessing, the final modeling data includes information of various medical services that 39,459 patients received in 2 years. The objective is to accurately predict the length of stay in hospital for those patients in the 3rd year. The data contains miscellaneous health care information including 45 diagnosis categories (e.g., gynecology, pneumonia, liver disorders, and stroke) and 16 procedure categories (e.g., evaluation and management, surgery-nervous system,
anesthesia, and radiology). All of the variables used for modeling and the corresponding explanations are shown in Table 4-1.

**Table 4-1: Variables of medical claim data**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member</td>
<td>Patient ID</td>
</tr>
<tr>
<td>Year</td>
<td>The year of the claim, Y1, Y2</td>
</tr>
<tr>
<td>DSFS</td>
<td>Months since first service that year</td>
</tr>
<tr>
<td>Age</td>
<td>Age of the patient</td>
</tr>
<tr>
<td>Sex</td>
<td>Sex of the patient</td>
</tr>
<tr>
<td>Hospital_y2</td>
<td>Total length of stay in hospital in year 2</td>
</tr>
<tr>
<td>Specialty</td>
<td>Provider’s specialty</td>
</tr>
<tr>
<td>POS</td>
<td>Generalized place of service</td>
</tr>
<tr>
<td>Length</td>
<td>Length of stay</td>
</tr>
<tr>
<td>Charlson</td>
<td>A measure of the effect diseases have on overall illness, grouped by significance, that generalizes additional diagnoses.</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>Broad diagnostic categories</td>
</tr>
<tr>
<td>Procedure</td>
<td>Broad categories of procedures</td>
</tr>
<tr>
<td>Drug_count</td>
<td>Count of unique prescription drugs filled by DSFS</td>
</tr>
<tr>
<td>Lab_count</td>
<td>Count of unique laboratory and pathology tests by DSFS</td>
</tr>
</tbody>
</table>

Every patient is assigned a unique member ID and is characterized by age and sex. The time when a medical service is received is specified by the year (year 1 and year 2) and the months since the first service of that year (month 1 – month 12). The total length of stay in hospital for each patient in year 2 is also available which is represented by the variable Hospital_y2. For each month, the total number of unique prescription drugs and unique laboratory tests are recorded, and they are represented by Drug_count and Lab_count respectively. The diagnosed conditions are categorized into 45 broad diagnostic categories, and each diagnosis is associated with Charlson Index which
indicates the effect of the diseases have on overall illness. Charlson Index is binned into 5 buckets: 0, 1-2, 3-4, 4-5, and 5+. A diagnosis leads to multiple medical procedures which fall into 16 broad categories. The place where a procedure takes place is represented by the variable POS which includes 9 categorical values. The response variable (Hospital_y3) is the days of stay in hospital in the 3<sup>rd</sup> year, and its value is an integer starting from 0.

4.1.2 Evaluation criterion

The accuracy of prediction is evaluated by:

$$\varepsilon = \sqrt{\frac{1}{n} \sum_{i=1}^{n} [\log(p_i + 1) - \log(a_i + 1)]^2}$$  \hspace{1cm} (4-1)

where:

1. \(a_i\) is actual length of stay in hospital in year 3 (Hospital_y3) for patient \(i\);
2. \(p_i\) is predicted length of stay in hospital in year 3 (Hospital_y3) for patient \(i\);
3. \(n\) is the total number of patients.

Table 4-2 lists the benchmarks and the corresponding evaluation criterion scores. The low bound of the evaluation criterion is 0, in which case all of predicted values fit exactly the actual values. If all predicted values are assigned 0, the score is 0.522226. Another constant value benchmark is 0.209179, which is inferred analytically with the objective function of minimizing the evaluation criterion with constant value, and it decrease score from 0.522226 in the all-zero scenarios to 0.486459. In the public leaderboard of the competition, 1353 final predictions are submitted by teams after 2-year competition period and the corresponding scores are ranked and compared to the above benchmarks. The 1<sup>st</sup>, 100<sup>th</sup>, 200<sup>th</sup>, and 300<sup>th</sup> ranked scores are listed, and one can see that the difference between the top scores are small. The resulting scores of the proposed
Table 4-2: Evaluation criterion benchmarks

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>0</td>
</tr>
<tr>
<td>1\textsuperscript{st} / 1353</td>
<td>0.443793</td>
</tr>
<tr>
<td>100\textsuperscript{th} / 1353</td>
<td>0.461527</td>
</tr>
<tr>
<td>200\textsuperscript{th} / 1353</td>
<td>0.462774</td>
</tr>
<tr>
<td>300\textsuperscript{th} / 1353</td>
<td>0.463787</td>
</tr>
<tr>
<td>400\textsuperscript{th} / 1353</td>
<td>0.466397</td>
</tr>
<tr>
<td>Optimized constant value (0.209179)</td>
<td>0.486459</td>
</tr>
<tr>
<td>All zeros</td>
<td>0.522226</td>
</tr>
</tbody>
</table>

4.1.3 Ontology construction

An initial ontology in Figure 4-1 is built based on the medical claim domain knowledge and data information. Table 4-3 lists the relations and the linked domains which correspond to different classes. The properties associated with all classes are shown in Table 4-4. The initial ontology is constructed by 4 classes (i.e., Patient, Medical Service, Diagnosis, and Procedure) which are linked by 4 relations (i.e., Receive Service, Categorize, Diagnose, and Make Procedure). Each patient receives medical service of year 1 and year 2. Since the length of stay in hospital in year 2 is available for each patient, the variable Hospital_y2 can be regarded as a property of class Patient. Each yearly medical service is further categorized into monthly medical services, and the
corresponding properties include Drug_count and Lab_count for each monthly medical service. The monthly medical service leads to diagnoses which are comprised of multiple diagnosis code and Charlson index. Each diagnosis leads to multiple procedures characterized by POS, length of stay in hospital, and specialty.

**Figure 4-1: Initial ontology**

**Table 4-3: Relations and domains**

<table>
<thead>
<tr>
<th>Relation</th>
<th>Domain</th>
<th>Class</th>
<th>Cardinality limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receive Service</td>
<td>From-domain</td>
<td>Patient</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>To-domain 1</td>
<td>Service_Y</td>
<td>[1, 2]</td>
</tr>
<tr>
<td></td>
<td>To-domain 2</td>
<td>Service_M</td>
<td>[1, 12]</td>
</tr>
<tr>
<td>Categorize</td>
<td>From-domain</td>
<td>Service_Y</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>To-domain</td>
<td>Service_M</td>
<td>[1, 12]</td>
</tr>
<tr>
<td>Diagnose</td>
<td>From-domain</td>
<td>Service_Y</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>To-domain</td>
<td>Service_M</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Diagnosis</td>
<td>[1, +]</td>
</tr>
<tr>
<td>Make Procedure</td>
<td>From-domain</td>
<td>Diagnosis</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>To-domain</td>
<td>Procedure</td>
<td>[1, +]</td>
</tr>
</tbody>
</table>
### Table 4-4: Classes and properties

<table>
<thead>
<tr>
<th>Class</th>
<th>Property</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient</td>
<td>Age</td>
<td>Ratio</td>
</tr>
<tr>
<td></td>
<td>Sex</td>
<td>Nominal</td>
</tr>
<tr>
<td></td>
<td>Hospital_y2</td>
<td>Ratio</td>
</tr>
<tr>
<td>Medical service Year</td>
<td>Year</td>
<td>Nominal</td>
</tr>
<tr>
<td>Medical service Month</td>
<td>DSFS</td>
<td>Interval</td>
</tr>
<tr>
<td></td>
<td>Drug_count</td>
<td>Ratio</td>
</tr>
<tr>
<td></td>
<td>Lab_count</td>
<td>Ratio</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>Diagnosis code</td>
<td>Nominal</td>
</tr>
<tr>
<td></td>
<td>Charlson</td>
<td>Ordinal</td>
</tr>
<tr>
<td>Procedure</td>
<td>Procedure code</td>
<td>Nominal</td>
</tr>
<tr>
<td></td>
<td>Specialty</td>
<td>Nominal</td>
</tr>
<tr>
<td></td>
<td>POS</td>
<td>Nominal</td>
</tr>
<tr>
<td></td>
<td>Length</td>
<td>Ratio</td>
</tr>
</tbody>
</table>

The case study is implemented in the statistical computing platform R. The ontology structure can be specified in the data type named `list` in R. Figure 4-2 shows the code for the ontology specification in R. For a list representing a class, the `id` serves as the identification and points the class to the corresponding variable in relational data in R. The `features` contains the list of associated properties; and the `offspring` represents the descending class it is linked to. This ontology specification helps to readily translate the relational data to the semi-structured ontology, which significantly facilitates the subsequent computations and ontology evolution.

```r
procedure_onto <- list(id='procedure', features=c("pos", "length", "delay"), offspring=NULL)
diagnosis_onto <- list(id='condition', features="charlson", offspring = procedure_onto)
month_onto <- list(id='dsfs', features=c("drug_count", "lab_count"), offspring = diagnosis_onto)
year_onto <- list(id='year', features=NULL, offspring=month_onto)
patient_onto <- list(id="member", features=c("sex", "age", "hospital_y2"), offspring = year_onto)
```

**Figure 4-2: Ontology specification in R**
4.1.4 Predictive model

The modeling objective is to predict the length of stay in hospital for each patient, so the response variable is the count of days which is an integer ranging from 0 to 365. Poisson regression is a statistical modeling approach specifically for making a prediction in the form of count. It is a generalized linear model with the logarithm as the link function, so the inference process is represented in a simple, neat, and interpretable function. Due to its statistical soundness and simplicity, Poisson regression is selected as the predictive model in this case study.

The form of Poisson regression model is given as

\[
\log(E(Y)) = \alpha + \beta'X
\]  

(4-2)

Where:

1. \(Y\) is the response variable in form of count;
2. \(X\) is a vector of independent variables;
3. \(\alpha\) and \(\beta\) are coefficients to be estimated.

One can see that the model is a generalized linear regression with logarithm link function. The predicted value \(E(Y)\) is applied on the evaluation criterion function

\[
\varepsilon = \frac{1}{n} \sum_{i=1}^{n} \left[ \frac{\log(p_i + 1) - \log(a_i + 1)}{2} \right]^2
\]

(4-1), and the resulting value \(\varepsilon\) is indicative of the model performance in terms of accuracy.

4.1.5 Feature construction

As the prediction is the length of stay for each patient, only the features associated with the class Patient can be directly fed to predictive model, so new features should be
constructed to the class Patient. As shown in Table 4-4, the raw data has 3 features (i.e., Age, Sex, and Hospital_y2) available on the patient level.

4.1.5.1 Simple cohort level feature

The feature construction starts from constructing the simplest cohort level features from 2 properties of the class Service_M, namely, Drug_count and Lab_count. The operator applied on these 2 properties is summation. It leads to 4 cohort level features at patient level:

\[
\begin{align*}
\text{Drug\_year1} &= \text{sum (Drug\_count (year\_1))} \\
\text{Drug\_year2} &= \text{sum (Drug\_count (year\_2))} \\
\text{Lab\_year1} &= \text{sum (Lab\_count (year\_1))} \\
\text{Lab\_year2} &= \text{sum (Lab\_count (year\_2))}
\end{align*}
\]

The group of constructed features represents the total number of unique drugs (lab tests) a patient receives in year 1 and year2. Higher values normally reflect higher severity of the diseases in medical history, so higher probability of requiring more time in hospital in the future.

Other operators such as Maximum can also be applied which leads to a new feature stands for the maximum number of unique drugs (lab tests) a patient receives in an individual month during year 1 and year2. Similarly, these features are expected to have positive relation with response variable.

4.1.5.2 Relation level feature evolution – place of service (POS)

In this case study, 2 relation level features are constructed and evolved based on initial hypotheses suggested by domain experts. The evolution process is driven by heuristic algorithm introduced in Section 3.3, and corresponding new hypotheses are
explored in pursuit of searching for optimal hypothesis-based features with high discriminative power for hospitalization prediction.

In the medical service, a procedure is delivered to a patient in one of the 9 places of service: $Popu_{POS} = \{\text{Missing value, Ambulance, Home, Independent lab, Inpatient hospital, Office, Outpatient hospital, Urgent care, Other}\}$. For the property POS, a new hypothesis is made: more unique POSs belonging to a subset of POS population $(Sub\_popu\_POS)$ where a patient receives procedures in medical history is indicative of larger likelihood of getting hospitalization in the future. This hypothesis is rooted in two intuitive assumptions: 1) normally more POS in medical history implies higher complexity of the diseases; and 2) only a subset of POS population reflects the severity of the diseases. The abstracted form of the new feature based on the above hypothesis is represented as:

$$Count_{POS} = \text{count}(\text{unique}(POS)), \quad POS \in Sub\_popu\_POS; \quad Sub\_popu\_POS \subseteq Popu\_POS$$

Based on the above formula, a $Sub\_popu\_POS$ corresponds to a feature $Count\_POS$, so the search for optimal feature $Count\_POS$ is realized by the exploration of the optimal $Sub\_popu\_POS$ implemented in the data-driven approach described in Section 3.3. The proposed initial $Sub\_popu\_POS = \{\text{Missing value, Home, Independent lab, Inpatient hospital, Outpatient hospital, Urgent care, Other}\}$. Genetic algorithm is selected as the heuristic searching method. As shown in Figure 4-3, an initial chromosome is built as the starting point for heuristic search where the binary value at each gene site indicates the presence of the corresponding POS in $Sub\_popu\_POS$. 


In the searching process, each candidate feature is fed to the predictive model Poisson regression along with the 3 initial patient level features: Sex, Age, and Hospital_y2. The Equation

$$C = w_1 \times Score_{support} + w_2 \times Score_{confidence}$$

(3-1)

is used as fitness function to guide the heuristic search in genetic algorithm. In this case, $Score_{confidence} = 1 - \varepsilon$, where $\varepsilon$ is deviance in Equation

$$\varepsilon = \sqrt{\frac{1}{n} \sum_i^n [\log(p_i + 1) - \log(a_i + 1)]^2}$$

(4-1)

which represents the modeling accuracy. Higher value of $Score_{confidence}$ implies more discriminative power the new feature contributes to hospitalization prediction. Because no instance is excluded for any derived hypothesis, $Score_{support}$ is constant value 1 in the searching process, and $w_1$ and $w_2$ have no impact on the fitness function.

An initial chromosome population of size 10 is randomly created, and the genetic-based searching scheme comprised of parent selection, crossover, and mutation is iterated 100 times. The chromosome with the highest criterion $C$ (lowest deviance $\varepsilon$) is selected as the optimal $Sub_{popu \_POS}$. The comparison between optimal feature and one of initial candidates are presented in Table 4-5.
The optimal sub_popu_POS selected in genetic algorithm only contains 2 POSs, namely, inpatient hospital and urgent care, which is consistent with the domain knowledge that inpatient hospital and urgent care are the two places most reflecting the high complexity and severity of diseases, and higher likelihood of more hospitalizations will occur in the future.

In the perspective of modeling accuracy, Table 4-5 shows that adding optimized feature Count_POS in Possion regression decreases deviance from 0.46332 to 0.46286, whereas the initial feature contributes no discriminative power evidenced by the slight increase of deviance. In the Possion regression model with optimized constructed feature, the 3 numeric features Count_POS, Age, and Hospital_y2 have positive coefficients, which indicates that they have positive relation to the response variable. For the feature Sex, categorical value of Female is associated with bigger value than Male, so it implies that female patients have more expected hospitalization stays than male patients.

**Table 4-5: Comparison of the optimal and initial constructed feature Count_POS**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Chromosome</th>
<th>Sub_popu_POS</th>
<th>Deviance</th>
<th>Poisson regression coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Count_POS</td>
</tr>
<tr>
<td>No feature</td>
<td>NA</td>
<td>NA</td>
<td>0.46332</td>
<td>NA</td>
</tr>
<tr>
<td>Initial feature</td>
<td>1 0 1 1 0 0 1</td>
<td>Missing value Home Independent lab Inpatient hospital Outpatient Hospital Urgent care Other</td>
<td>0.46335</td>
<td>0.01</td>
</tr>
<tr>
<td>Optimal feature</td>
<td>0 0 0 1 0 0 0 0 1</td>
<td>Inpatient hospital Urgent care</td>
<td>0.46286</td>
<td>0.22</td>
</tr>
</tbody>
</table>
4.1.5.3 Relation level feature evolution - Procedure & Diagnosis

In the same rational as POS, normally a certain group of diagnoses and procedures are regarded as more related to hospitalization, and more unique diagnoses and procedures a patient received in medical history indicates more severe illnesses and larger likelihood of hospitalization in future. Accordingly, a hypothesis can be formulized as:

\[
\text{Diag}_\text{proc} = \text{IF}(\text{Service}_Y2.\text{DSFS} > \text{Month AND Diag.code IN Sub.popu.diag AND Proc.code IN Sub.popu.proc})
\]

where:

1. \text{Month} is a integer ranging from 0 to 11;
2. \text{Sub.popu.diag} is a subset of population of diagnosis;
3. \text{Sub.popu.proc} is a subset of population of procedure.

If the DSFS in year 2 is greater than \text{Month} and the diagnoses and delivered procedures are belonging to, respectively, \text{Sub.popu.diag} and \text{Sub.popu.proc}, the binary feature \text{Diag.proc} is assigned 1, otherwise 0. Conceptually, value 1 of the feature \text{Diag.proc} indicate that the patient has suffered serious illness at the most recent time, so more stays in hospital in future is expected, and it has positive relation to the response variable.

As illustrated in the formula, each combination of \text{Month}, \text{Sub.popu.diag}, \text{Sub.popu.proc} determines a constructed feature \text{Diag.pro}, which leads to an exhaustive search space of size 8,832 (12*45*16). The optimal combination of \text{Month}, \text{Sub.popu.diag}, \text{Sub.popu.proc} are determined by heuristic search of genetic algorithm. In each searching step, the corresponding \text{Diag.proc} is constructed and its discriminative power is evaluated by the prediction performance of Possion regression including Age, Sex, Hospital_\text{y2}, Count_POS, and Diag.proc. The optimal feature is obtained after 500 iterations of the heuristic search. Table 4-6 compares the optimal \text{Diag_pros}, an initial \text{Diag.proc} and no \text{Diag.proc} in terms of chromosome, deviance and
coefficient in Poisson regression. The details of month threshold, subset of population of diagnosis and procedure corresponding to the initial feature and optimal feature can be found in Table 4-7. Including the optimal Diag_proc decrease the deviance from 0.46268 to 0.46204, whereas the initial feature contributes little discriminative power evidenced by the slight increase of deviance. The coefficient in the resulting model for the optimal Diag_proc is 0.46, which is consistent with the assumption of positive relationship between Diag_proc and response variable.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Sub_popu_diag</th>
<th>Sub_popu_proc</th>
<th>Month</th>
<th>Deviance $\epsilon$</th>
<th>Coefficient of Diag_proc</th>
</tr>
</thead>
<tbody>
<tr>
<td>No feature</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>0.46286</td>
<td>NA</td>
</tr>
<tr>
<td>Initial feature</td>
<td>1011000001010001</td>
<td>11110001110001</td>
<td>001000000000</td>
<td>0.46305</td>
<td>0.26</td>
</tr>
<tr>
<td>Optimal feature</td>
<td>11110000010011110</td>
<td>1111111101110000</td>
<td>00000000010</td>
<td>0.46204</td>
<td>0.46</td>
</tr>
</tbody>
</table>

In summary, these two construction processes demonstrates that the feature built by the proposed methodology has two most important and desirable traits: 1) contributes significant discriminative power for prediction; and 2) is interpretable in terms of the logic behind the building process, so the feature can be validated by domain experts and, more importantly, facilitate exploration and understanding on the relevant domain. Specifically, the domain hypothesis of a certain group of POS, diagnosis, and procedures are more relevant to the likelihood of hospitalization is validated by data. On the other hand, the resulting group of POS, diagnosis, and procedure proved by data to be relevant.
to future hospitalization can be further evaluated and validated by domain experts. For example, the POSs selected to be most relevant to future hospitalization in the data-driven fashion include inpatient hospital and urgent care, which is consistent with general medical knowledge.

Table 4-7: Subset of population of diagnosis, procedure and month

<table>
<thead>
<tr>
<th>Feature</th>
<th>Sub_popu_diag</th>
<th>Sub_popu_proc</th>
<th>Month</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initial feature</strong></td>
<td>Gastrointestinal, inflammatory bowel disease, and obstruction</td>
<td>Surgery-Digestive System Evaluation and Management Medicine</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Miscellaneous 2</td>
<td>Pathology and Laboratory</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pregnancy</td>
<td>Surgery-Nervous System</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Urinary tract infections</td>
<td>Surgery-Respiratory System</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arthropathies</td>
<td>Surgery-Integumentary System</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Other metabolic</td>
<td>Surgery-Urinary System</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fractures and dislocations</td>
<td>Surgery-Genital System</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chronic obstructive pulmonary disorder</td>
<td>Surgery-Eye and Ocular Adnexa</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Miscellaneous 3</td>
<td>Surgery-Maternity Care and Delivery</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Non-malignant hematologic</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Skin and autoimmune disorders</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Atherosclerosis and peripheral vascular disease</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chronic renal failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Other cardiac conditions</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Congestive heart failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gynecologic cancers</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Liver disorders</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Diabetic ketoacidosis and related metabolic</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cancer A&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pancreatic disorders</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stroke</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acute renal failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sepsis</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Perinatal period</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gastrointestinal, inflammatory bowel disease, and obstruction</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gynecology</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Miscellaneous 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pregnancy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Urinary tract infections</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cancer B</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>All other infections</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acute myocardial infarction</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fractures and dislocations</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chest pain</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chronic obstructive pulmonary disorder</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Non-malignant hematologic</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chronic renal failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Other cardiac conditions</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Congestive heart failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Appendicitis</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Liver disorders</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Seizure</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Miscellaneous 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hip fracture</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Catastrophic conditions</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Perinatal period</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ovarian and metastatic cancer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

73
4.1.6 Ontology reconstruction

After the construction of features of cohort level and relation level, new properties are associated with classes as shown in Table 4-8. The newly constructed feature at patient level are highlighted in bold. Because the properties Drug_count and Lab_count have been used for constructing new features, their most discriminative information have been included in the new feature. DSFS is involved in deriving the feature Diag_proc, but if it is demonstrated that DSFS does not play a big role so it can be excluded from the feature construction process, all of the 3 properties associated with Service_M can be excluded from ontology.

<table>
<thead>
<tr>
<th>Class</th>
<th>Property</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient</td>
<td>Age</td>
<td>Ratio</td>
</tr>
<tr>
<td></td>
<td>Sex</td>
<td>Nominal</td>
</tr>
<tr>
<td></td>
<td>Hospital_y2</td>
<td>Ratio</td>
</tr>
<tr>
<td></td>
<td>Drug_year1</td>
<td>Ratio</td>
</tr>
<tr>
<td></td>
<td>Drug_year2</td>
<td>Ratio</td>
</tr>
<tr>
<td></td>
<td>Lab_year1</td>
<td>Ratio</td>
</tr>
<tr>
<td></td>
<td>Lab_year2</td>
<td>Ratio</td>
</tr>
<tr>
<td></td>
<td>Count_POS</td>
<td>Ratio</td>
</tr>
<tr>
<td></td>
<td>Diag_proc</td>
<td>Nominal</td>
</tr>
<tr>
<td>Service_Y</td>
<td>Year</td>
<td>Nominal</td>
</tr>
<tr>
<td>Service_M</td>
<td>DSFS</td>
<td>Interval</td>
</tr>
<tr>
<td></td>
<td>Drug_count</td>
<td>Ratio</td>
</tr>
<tr>
<td></td>
<td>Lab_count</td>
<td>Ratio</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>Diagnosis code</td>
<td>Nominal</td>
</tr>
<tr>
<td></td>
<td>Charlson</td>
<td>Ordinal</td>
</tr>
<tr>
<td>Procedure</td>
<td>Procedure code</td>
<td>Nominal</td>
</tr>
<tr>
<td></td>
<td>Specialty</td>
<td>Nominal</td>
</tr>
<tr>
<td></td>
<td>POS</td>
<td>Nominal</td>
</tr>
<tr>
<td></td>
<td>Length</td>
<td>Ratio</td>
</tr>
</tbody>
</table>

A chromosome derived from the optimal one with changing the month threshold from 10 (0 0 0 0 0 0 0 0 0 0 1 0) to 0 (1 0 0 0 0 0 0 0 0 0 0) is assessed for its discriminative
power in Poisson regression. As shown in Table 4-8, the resulting deviance and coefficient have no negligible difference between the optimal Diag_proc and the one with modified month threshold. So if the modified Diag_proc with month threshold as 0 is selected, the corresponding formula is:

\[
\text{Diag} \_ \text{proc} = \text{IF}(\text{Service}_Y.\text{DSFS} > 0 \text{ AND } \text{Diag.code IN Service.diag AND Proc.code IN Service.proc})
\]

which can be simplified to:

\[
\text{Diag} \_ \text{proc} = \text{IF}(\text{Diag.code IN Service.diag AND Proc.code IN Service.proc})
\]

One can see that the property DSFS is not included in this feature construction.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Month</th>
<th>Deviance $\epsilon$</th>
<th>Coefficient of Diag_proc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal feature</td>
<td>0 0 0 0 0 0 0 0 1 0</td>
<td>0.46204</td>
<td>0.46</td>
</tr>
<tr>
<td>Optimal feature with modified Month</td>
<td>1 0 0 0 0 0 0 0 0 0</td>
<td>0.46194</td>
<td>0.43</td>
</tr>
</tbody>
</table>

Figure 4-4: Comparison of optimal Diag_proc and derived Diag

As all of the 3 properties associated with Service_M do not significantly contribute to hospitalization prediction any more, it is reasonable to remove the class Service_M from ontology without losing substantial overall information. As shown in Figure 4-5, the relation Diagnose directly links from class Service_Y to class Diagnosis, which is in line with the pre-defined constrains on the domains a relation directs from and to as shown in Table 4-3.
This step is aimed to elaborate the ontology reconstruction process (simplification in this case) with justification of data-driven derivation and domain knowledge, rather than to demonstrate that the class Service_M is not useful in the ontology. The determination of ontology reconstruction is a trade-off between 1) general use and task-specific use; 2) and retaining maximal information and keeping neatest form.
5 Conclusions

In a sense, “knowledge discovery” has not been best practiced because not only is modeling on non-structured data an under-researched filed, but also is the importance of generating meaningful and interpretable outcomes constantly neglected. This study strives to resolve these issues to some extent by constructing discriminative and interpretable features from non-structured data to realize both knowledge discovery and prediction.

The extensive body of literature review reveals the lack of unified and effective feature construction framework for non-structured data. The domain-specific structural information in non-structured data requires the involvement of domain knowledge in the data-driven procedure. So the study proposes to represent the non-structured data in an intuitive form of ontology, and it is demonstrated to notably facilitate subsequent feature construction and modeling process. A suitable hypothesis made by domain experts can be applied on the ontology, and a corresponding feature is built. The most important functionality of this framework is that the domain opinions and data jointly drive the evolution of hypothesis, optimizing corresponding features in terms of prediction ability and meaningfulness. One byproduct of feature construction is that the ontology can also be evolved and restructured, in response to the results of hypothesis evolution and domain opinions, to best represent the domain-specific structure and knowledge in the data. Therefore, as opposed to merely building features for modeling, the proposed framework serves the ultimate and general goal of discovering and representing knowledge from data for a specific domain.
The implementation in the medical claim data verifies the feasibility and effectiveness of every functionality embedded in the framework. The high discriminative power of the constructed feature is demonstrated by the comparison to benchmarks. So the features serve well for the prediction purpose. More importantly, the evolved hypothesis and features convey information that can be easily understood by domain experts, which helps verify existing hypotheses and discover new knowledge that can be assessed and accepted.

In summary, the proposed approach is the first, to the best of knowledge, unified feature construction framework for non-structured data with experimental validation of effectiveness and robustness. It serves as the a bridge between a specific domain and data analytics, so a domain expert can easily input domain insights, evaluate existing hypotheses, learn new knowledge provided by data, and make predictions of competitive accuracy. Essentially, it is a successful practice on the goal of seamlessly integrating data and domain knowledge for knowledge learning.

Future efforts should be focused in the following three aspects. First, the framework’s compatibility to the data in different fields, other than medical claim data, needs to be further validated. Second, the flexibility can be improved by exploring more heuristic search algorithms, evaluation criteria, and new ontology specifications. Finally, a software with a GUI based on the framework can be developed, which will go a long way toward facilitating the implementation and improving efficiency.
Reference


Ghani, R., et al. (2000). Data mining on symbolic knowledge extracted from the web. Workshop on Text Mining at the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.


West, K. and S. Cox (2004). Features and classifiers for the automatic classification of musical audio signals. ISMIR.


