LEAP: Learning to Prescribe Effective and Safe Treatment Combinations for Multimorbidity

Yutao Zhang
Tsinghua University
yt-zhang13@mails.tsinghua.edu.cn

Robert Chen
Georgia Institute of Technology
rchen87@gatech.edu

Jie Tang
Tsinghua University
jietang@tsinghua.edu.cn

Walter F. Stewart
Sutter Health
stewarwf@sutterhealth.org,

Jimeng Sun
Georgia Institute of Technology
jsun@cc.gatech.edu

ABSTRACT

Managing patients with complex multimorbidity has long been recognized as a difficult problem due to complex disease and medication dependencies and the potential risk of adverse drug interactions. Existing work either uses complicated rule-based protocols which are hard to implement and maintain, or simple statistical models that treat each disease independently, which may lead to sub-optimal or even harmful drug combinations. In this work, we propose the LEAP (LEArn to Prescribe) algorithm to decompose the treatment recommendation into a sequential decision making process while automatically determining the appropriate number of medications. A recurrent decoder is used to model label dependencies and content-based attention is used to capture label instance mapping. We further leverage reinforcement learning to fine tune the model parameters to ensure accuracy and completeness. We incorporate external clinical knowledge into the design of the reinforcement reward to effectively prevent generating unfavorable drug combinations. Both quantitative experiments and qualitative case studies are conducted on two real world electronic health record datasets to verify the effectiveness of our solution. On both datasets, LEAP significantly outperforms baselines by up to 10-30% in terms of mean Jaccard coefficient and removes 99.8% adverse drug interactions in the recommended treatment sets.

KEYWORDS

Treatment Recommendation, Multimorbidity, Multi-Instance Multi-Label Learning

1 INTRODUCTION

Multimorbidity, i.e., the co-occurrence of two or more chronic or acute medical conditions in an individual patient, is increasingly prevalent and represents a major challenge in healthcare [26, 32]. The goal of treatment recommendation for such patients is to decide the most appropriate combination of treatments for their disease

conditions. Such decisions involve the assessment of patient's medical condition, the efficacy of the treatments and potential adverse effects of the treatments. Doctors typically prescribe medications based on their intuition and experience. However, due to knowledge gaps or unintended biases, often times these clinical decisions can be sub-optimal. Studies have shown that 6.7% of hospital patients in the United States suffer from serious adverse drug reactions and 0.32% of such adverse drug events are fatal, leading to a yearly cost of over $136 billion [20]. Broad adoption and usage of electronic health records (EHRs) in the last decade has opened up a great opportunity to leverage healthcare data for improved clinical decision making. In this context, computer-assisted treatment recommendation systems based on EHR data could provide tremendous value.

Prior research on treatment recommendation based on EHR data is limited. Existing treatment recommendation systems are mainly implemented using rule-based protocols defined by doctors based on clinical guidelines or personal experience [1, 8, 14]. These hard-coded protocols are time-intensive to curate and difficult to maintain or implement [16]. Moreover, due to the high prevalence of complex clinical conditions, which often require personalized drug combinations to treat, hard-coded protocols are not able to provide optimized personalized treatment regimens, especially for those complex patients who present with multiple diseases concurrently. We seek to address this issue by discovering hidden clinical knowledge from EHR data and leveraging it to form effective and safe treatment recommendations.

EHRs are clinical records that capture comprehensive medical histories of patients, including diagnoses, medications, treatment plans, imaging and laboratory test results. In this work, our objective is to learn a prediction model from an EHR that takes a set of disease conditions as input and gives a treatment recommendation in the form of a set of medications and their mapping to those disease conditions. Disease conditions and treatment plans are represented as discrete sets of diseases and drugs, respectively. Other available clinical evidence such as demographics, test results or allergies can be also included as additional input along with disease conditions.

We formulate the treatment recommendation as a Multi-Instance Multi-Label (MIML) learning problem [42], where training samples are provided as a set of instances (i.e., disease conditions) and the associated label set (i.e., medications). The direct mapping between instances (diseases) and labels (drugs) are absent. There exists a large body of literature on both Multi-Instance and Multi-Label problems [6, 7, 27, 41, 42]. Most prior solutions either assume

independence among the items in the label set, or equal contribution of all of the instances towards the label set. These assumptions are invalid in our problem due to the inherent, complex higher order relationships among diseases and drugs. For example, the drugs paroxetine and quietepine are typically used for depression, and traditional models for treatment recommendation would assign the patient to these drugs if depression is in the patient's disease diagnosis set. However, since these drugs may lead to weight gain, they should be avoided in depression patients that also present with obesity. Dependencies like this are not able to be captured by models following the independence assumptions made in traditional models.

To address the aforementioned challenges, we propose LEArn to Prescribe model (LEAP) to jointly consider label dependency and label instance mapping in an end-to-end learning framework. Our contributions can be summarized as follows:

- We formulate treatment recommendation as a sequential decision making problem and propose LEAP, which uses a recurrent decoder to model label dependency and uses content-based attention to model the underlying label-to-instance mapping.
- LEAP also incorporates a policy gradient based reinforcement learning method to fine-tune the model which can effectively improve the accuracy and completeness of treatment recommendation.
- We show that external domain knowledge can also be integrated into LEAP to avoid unfavorable drug interactions in treatment recommendation.
- We conduct both quantitative experiments and qualitative case studies on real world EHR datasets to demonstrate the effectiveness of LEAP.

2 RELATED WORK

Treatment Recommendation systems are mainly implemented based on hard-coded recommendation protocols [1, 8, 14] which are typically defined by doctors or their institution's guidelines. For example, Lakkaraju and Rudin [19] propose to use a Markov Decision Process (MDP) to provide cost-effective recommendations based on a healthcare institution's financial restrictions. There are existing works on estimating treatment effect [10, 24] using clinical data, but this line of work is limited to the estimation of the effectiveness of some specific drug combinations for target disease conditions.

Wei et al. [36] construct a medication-indication database by integrating resources from medical knowledge bases such as RxNorm[1], SIDER[2], and Wikipedia[3]. However, despite the accuracy and completeness of the mapping, the indication database only specifies the potential correlation between diseases and drugs and cannot directly support personalized treatment recommendation. Bajor and Lasko [3] proposed to use recurrent neural networks to predict medications given a patient's clinical history. The drawback to their strategy is that it predicts whether a drug is being used by a patient but does not recommend an effective combination of drugs for managing multiple disease conditions.
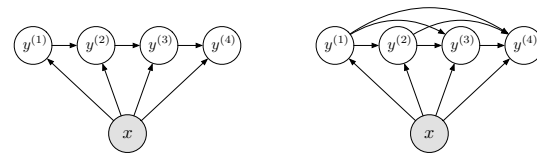
(a) Binary Relevance Method **(b) Classifier Chains Method**

Figure 1: Graphical representation of BR (1a) and CC (1b) for single instance multi-label learning with $|\mathcal{Y}| = 4$.

Multi-Instance Learning was first described by Dietterich et al. [9], where training examples are bags composed of instances, and the task is to predict the labels of unseen bags through analyzing the training bags with known labels. A bag is positive if it contains at least one positive instance, otherwise negative. However, the labels of the instances in each bag are unknown. Multi-instance learning has been used for biomedical applications such as prediction of binding sites in mRNA molecules [4], and classification of mammogram images [43]. To date, there have been no implementations of multi-instance learning for treatment recommendation. More recently, deep learning methods have been developed for multi-instance learning [37, 38]. Wu et al. [37] proposed a multi-instance convolutional neural network method for learning deep representations from image data. Xu et al. [38] demonstrated the use of subnets in multi-instance neural networks which are used to classify individual concepts. Mapping to our problem, one may consider a bag to be a set of disease diagnoses for a patient and the label to be a drug. However, we wish to impose an extra constraint such that the drug should cover all the diagnoses in the bag, which is very different from traditional multi-instance learning. Our problem differs from the traditional multi-label learning setting since an effective drug for any given disease a patient has is not necessarily an appropriate treatment due to the complex dependency among drugs and diseases.

Multi-Label Learning studies the problem where each instance is associated with a set of labels simultaneously [40]. Prior works have tackled the multi-label learning problem by transforming it into other well-established learning scenarios. The simple solution is Binary Relevance (BR) [5] which assumes independence among labels and decomposes multi-label learning into a set of binary classification tasks. Fürnkranz et al. [11] propose Calibrated Label Ranking to distinguish relevant and non-relevant labels by adding a "neutral element". However, the above methods ignore the correlations among labels. Chen et al. [28] propose Classifier Chains (CC) to model label dependency by using previous predictions as extra input for future classifiers. Figure 1 illustrates the graphical representation of BR and CC methods. Similar to CC, we also decompose the problem as a chain of correlated sub-problems, but instead of learning a set of binary classifiers with separate parameters for each label, we interpret this problem as a sequential decision making process. Liu et al. [22] discuss the optimality of Classifier Chains by considering the order of labels. The label dependency

can also be modeled by graphical models, such as Conditional Random Fields [12], Dependency Networks [15], and co-occurrence matrices [39]. However, most of these models only capture simple pairwise co-occurrences of labels and exhibit a high computation cost when there are a large number of labels (e.g., if there are over 1000 drugs, and each drug corresponds to a label).

Multi-Instance Multi-Label Learning Multi-instance multi-label (MIML) learning combines the settings of multi-instance learning and multi-label learning where a set of multiple labels is associated with each instance bag. Zhou et al. [41] first address this problem in the context of scene classification and proposes to transform the MIML problem to an ensemble of single instance single label (SISL) sub-problems. Li et al. [21] modeled which instances trigger which labels by considering the shared patterns across relevant labels. Briggs et al. [6] proposed rank-loss support instance machines for MIML instance annotation. Briggs et al. [7] considered the problem of predicting instance labels while learning from data labeled only at the bag level by using a new regularized rank-loss objective. Huang et al. [17] proposed a fast MIML algorithm by exploiting label relations with shared space and discovering sub-concepts for complicated labels. Pham et al. [27] used a discriminative probabilistic model to discover novel class instances in a MIML setting. Most existing approaches assume equal contribution of all the instances towards the label set which is different from our setting due to the complex higher order relationship among diseases and drugs.

Recurrent Neural Networks and Attention Mechanism Recently, recurrent neural networks (RNNs) have been successfully applied in various sequence modeling tasks including language modeling [29], machine translation [30], speech recognition [13], and image captioning [23, 34]. Wang et al. [35] propose to leverage an RNN decoder to model label dependency in a multi-label learning problem, which is closely related to our work. An attention mechanism [2, 25, 33] can be augmented to RNNs in order to guide the network to focus on one particular part of the input. This strategy has been adopted for translation of text between languages [2], and for classification of images [25]. Vinyals et al. [33] modeled unordered sets by decomposing output random variables using a chain rule, and used an attention RNN to model long-term correlation. To date, there is no existing work that leverages RNN with attention to address a MIML learning problem.

3 CHALLENGES OF TREATMENT RECOMMENDATION

We seek to build a treatment recommendation system which mimics the clinical decision making process of doctors by learning from diagnosis and medication relations of existing EHR data. Based on their domain knowledge, doctors prescribe treatments for patients by carefully considering all the dependencies among diseases and medications to find the optimal combination of treatments for a given patient. Such knowledge of specific combinations of drugs used for specific combinations of diagnoses, is encoded in the large amount of EHR data used in this study. Figure 2 summarizes common dependencies among medical conditions and drugs.
Drug to Disease Mapping: Generally, there is an underlying mapping between medications and disease conditions that indicates the effectiveness of medications towards diseases. One disease
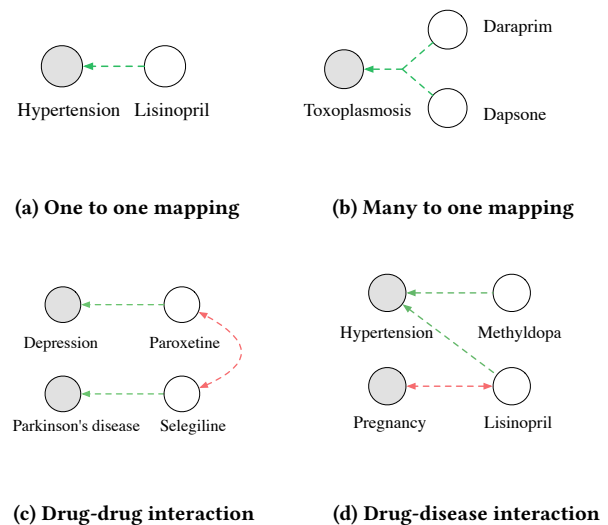
Figure 2: Example high order dependencies among diagnoses and drugs. Gray nodes indicates diseases, white nodes indicates drugs. Green arrows between drugs and diseases indicates treatment relationship. Red arrows indicates adverse drug interactions.

may require a combination of multiple medications as a treatment (Figure 2b), while another medication may be suitable for treating multiple related diseases. To select appropriate treatment plans for patients, however, requires the expertise and experience of doctors. Doctors typically formulate a mapping or hierarchy of drug and disease relationships based on their medical knowledge and experiences. A comprehensive medication to disease condition mapping that accounts for the complexity of patients with multimorbidity is difficult to explicitly construct but can be helpful in assisting with patient care. There have been previous efforts to create a medication-indication database by integrating resources from medical knowledge bases such as RxNorm, SIDER, and Wikipedia [36]. However, besides the concern of its accuracy and the incompleteness of the mapping, the main drawback of such a database is that it does not dictate how drugs should be prescribed when a patient presents with a complex constellation of multiple diseases and existing medications. This is concerning because oftentimes certain drug combinations should be avoided because they may either exacerbate a co-occurring disease in the patient, or because the drug combination is antagonistic and results in a non-efficacious or harmful outcome for a patient.
Adverse Drug Interaction: When a drug interacts with another drug the patient is taking (drug-drug interaction), or interacts with another medical condition the patient exhibits (drug-disease interaction), the outcome may be unfavorable (i.e. may result in severe allergic reaction, death). Drug-drug interactions can be caused by duplication (if two drugs with the same effect are taken, their side effects may be intensified), antagonism (two drugs with opposing actions can interact, thereby reducing the effectiveness of one or both), or alteration (one drug may alter how the body absorbs, distributes, metabolizes, or excretes another drug). Figure 2c shows an

example of a drug-drug interaction. In this example, both paroxetine and selegiline increase serotonin levels in the blood. However, an excessive serotonin level is a potential life-threatening situation. Thus, the combination of these two drugs should be avoided in the treatment. Sometimes, drugs that are helpful in one disease or medical condition are harmful for another. For instance, exposure to ACE inhibitors such as lisinopril has been associated with fetal abnormalities. Thus, lisinopril should be avoid as a treatment for pregnant patient with hypertension (Figure 2d).

4 LEAP MODEL

We first define the notations and formulate treatment recommendation problem (Section 4.1), followed by the description of basic LEAP method (Section 4.2), and how to leverage reinforcement learning to improve recommendation performance and avoid adverse drug interactions (Section 4.3).

4.1 Problem Formulation

Let $\mathcal{X}$ denote the diagnosis space and $\mathcal{Y}$ denote the medication space. $\mathbf{R} = \{(X_1, Y_1), (X_2, Y_2), ...\}$ is a set of medical records where $X_k \subseteq \mathcal{X}$ is a diagnosis set $X_k = \{x_1^k, x_2^k, ..., x_{|X_k|}^k\}$ and $Y_k \subseteq \mathcal{Y}$ is a medication set $Y_k = \{y_1^k, y_2^k, ..., y_{|Y_k|}^k\}$, where $|X_k|$ and $|Y_k|$ are the cardinalities of X_k and Y_k, respectively. The explicit mapping between a diagnosis in X_k and a medication in Y_k is not given. To avoid clutter, we omit k in the notation if there is no ambiguity.

The treatment recommendation objective is to select an optimal set of medications Y from all medications $\mathcal{Y}$ based on a diagnosis set X. Thus, we want to model the conditional probability $p(Y|X)$ and find the maximum likelihood solution Y^*. We formulate treatment recommendation as a Multi-Instance Multi-Label Learning (MIML) problem [42]. The task is to learn a function $f : 2^{\mathcal{X}} \to 2^{\mathcal{Y}}$ that maps an arbitrary set of diagnoses to a corresponding set of medications. The challenge here is to model: (1) label instance mapping, i.e. the relationship between drugs and diseases. (2) label dependency, e.g. drug drug interactions.

4.2 Basic LEAP

We propose to incorporate label dependency and instance-label mapping in an end-to-end learning framework. We decompose the problem as a chain of correlated sub-problems.In particular, we interpret this problem as a sequential decision making process, which means we add the medication one at time. In particular, We select the t-th medication y_t from $\mathcal{Y}$ based on the input X and the already selected medications $y_1, ..., y_{t-1}$. Intuitively, early selected drug y_t is more important than later selected drug $y_{t'}$ when $t < t'$.

Figure 3 gives an overview of LEAP. The input on the left hand side are the set of diseases x_1 to x_4, which are embedded into a 4-dimensional space $\mathbf{x_1} ... \mathbf{x_4}$ (indicated by rectangles with 4 circles). Then via an attention mechanism, we compute the disease-drug mapping and their association to each selected drug y_i and its latent state $\mathbf{s_i}$. Here $\mathbf{s_0}$ is a vector represents the initial state of drug selection process, y_0 is a <START> label. To enable the model to generate variable length output, we append an <END> label after each Y to indicate the termination of recommendation. In this way,

LEAP is able to automatically determine the number of drugs recommended in Y as well as the sequence of drug recommendations based on their importance.

Formally, the conditional probability of Y given X is decomposed as

$$p(Y|X) = \prod_{t=1}^{|Y|} p(y_t|X, y_1, y_2, ..., y_{t-1})$$
$$= \prod_{t=1}^{|Y|} p(y_t|\{x_1, x_2, ..., x_{|X|}\}, y_1, y_2, ..., y_{t-1}), \quad (1)$$

where x_i is the ith diagnosis code in X and y_t is the t-th medication being selected, $\{...\}$ indicates a set of unordered and non-repetitive discrete instances. Each diagnosis x_i in X is $|\mathcal{X}|$-dimensional one hot vector and each medication y_t in Y is a $|\mathcal{Y}|$-dimensional one hot vector. We use distributed representation to encode both diagnosis and medications, that is, we leverage embedding matrix $\mathbf{W}_{\mathcal{X}}$ and $\mathbf{W}_{\mathcal{Y}}$ to project diagnosis x_i and medication y_j into a unified d-dimensional space. We use vector $\mathbf{x}_i, \mathbf{y_j} \in \mathbb{R}^d$ to denote the embedding of x_i and y_j respectively, where

$$\mathbf{x}_i = \mathbf{W}_{\mathcal{X}} x_i, \quad \mathbf{y_j} = \mathbf{W}_{\mathcal{Y}} y_i. \quad (2)$$

To model different degrees of contribution of each $x_i \in X$ to medication y_t at step t we leverage a content-based attention mechanism [33], which is widely used in sequence modeling. Let $\mathbf{s}_t \in \mathbb{R}^d$ be a variable summarizing the state at step t.

$$\mathbf{s}_t = g(\mathbf{s}_{t-1}, y_{t-1}, \Psi_t(X)) \quad (3)$$

where $\Psi_t(\cdot)$ is an attention function that encode the compatibility between each $\mathbf{x}_i$ and the current state variable.

$$\Psi_t(X) = \sum_{i=1}^{|X|} \mathbf{M}_{ti} \mathbf{x}_i, \quad (4)$$

where $\mathbf{M} \in \mathbb{R}^{|X| \times |Y|}$ is an mapping matrix, in which each element $\mathbf{M}_{ti}$ indicates the contribution of the ith diagnosis code x_i to generating the tth medication y_t. We formulate mapping matrix $\mathbf{M}$ as

$$\mathbf{M}_{ti} = \frac{\exp(\alpha(\mathbf{x}_i, \mathbf{s}_{t-1}))}{\sum_{k=1}^{|X|} \exp(\alpha(\mathbf{x}_k, \mathbf{s}_{t-1}))} \quad (5)$$

where $\alpha(\mathbf{x}_i, \mathbf{s}_{t-1})$ is a function determines the weight for the ith diagnosis code. $\alpha(\mathbf{x}, \mathbf{y})$ is implemented as a MLP that takes the concatenation of $\mathbf{x}$ and $\mathbf{y}$ as input. Now we can rewrite Equation (3) as

$$\mathbf{s}_t = g([\Psi_t(X); \mathbf{y}_{t-1}], \mathbf{s}_{t-1}), \quad (6)$$

where $[\cdot; \cdot]$ denotes the concatenation of two vectors, g can be defined as a RNN unit (e.g. we used GRU in our implementation). The prediction at step t is given by

$$y_t = \arg\max_{y \in \mathcal{Y}} \text{softmax}(\mathbf{s}_t). \quad (7)$$

Training Model learning is to find an optimal parameter θ that maximize the conditional log-probability on training samples $\mathbf{R}$.
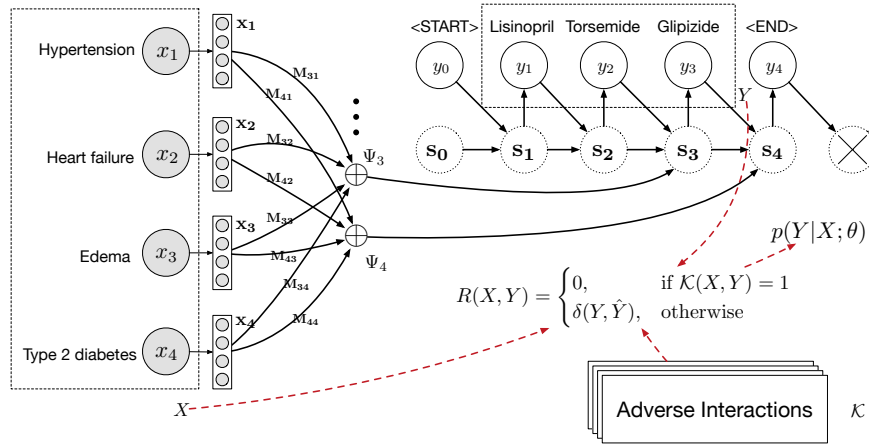
Figure 3: An overview of our solution for treatment recommendation. Gray nodes indicate input diagnoses, white nodes denote output treatments, dashed nodes are state variables.

$$\theta^* = \arg \max_{\theta} \sum_{(X,Y) \in \mathbf{R}} \log p(Y|X; \theta)$$

$$= \arg \max_{\theta} \sum_{(X,Y) \in \mathbf{R}} \sum_{t=1}^{|Y|} \log p(y_t|X, y_1, ..., y_{t-1}; \theta). \quad (8)$$

This can be achieved by leveraging the cross-entropy loss on Equation (7) and back-propagation. Recall that in Equation (1) we decompose $p(Y|X)$ using chain rule, hence, we condition random variables $y_1, ..., y_t$ in a particular order. In principle the order should not matter, but due to the non-convex nature of the optimization there can be certain orders better than others empirically [33]. We explore four different heuristics to order the training data:

- *Frequent first* determines the order of drugs by their occurrence frequencies in the training data where drugs prescribed more frequently will be put in front of the less frequent ones.
- *Rare first* put rare drugs before more frequent drugs.
- *Vocabulary*, we sorts drugs in an order according to fixed vocabulary. In the experiment, we used the alphabetical order.
- *Random*, we shuffles the labels of each sample randomly at training time.

Inference With the learned model parameter θ, the inference step is to predict the optimal treatment set $Y^* = \{y_1^*, y_2^*, ..., y_{|Y|}^*\}$. The probability of each medication y_t^* is computed with the information of the diagnosis and the previously predicted medications $y_1^*, ..., y_{t-1}^*$. The model automatically determine the sequence length of labels by finding a prediction path that maximizes a priori probability

$$y_1^*, ..., y_t^* = \arg \max_{y_1, ..., y_t} \log p(y_1, ..., y_t|X; \theta)$$

$$= \arg \max_{y_1, ..., y_t} \sum_{k=1}^{t} \log p(y_k|X, y_1, ..., y_k; \theta). \quad (9)$$

Finding the global optimal of Equation (9) is intractable, thus, we employ beam search to estimate the best prediction path. Let $\mathcal{B}_t$ be a beam of size K, $Y_{t':t}$ is a search path from t' to t, C is a candidate
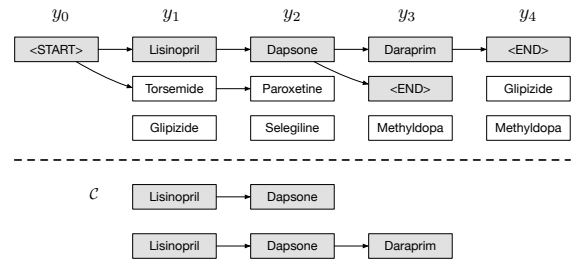

Figure 4: Top: beam search tree with a beam of size $K = 2$. Gray rectangle indicates active search path at $t = 4$. Bottom: resulting candidate prediction paths.

set of completed prediction paths. Figure 4 illustrate the process of beam search. Starting from a <START> label, we iteratively extend the search path. At step $t + 1$, we extend each active search path $Y_{0:t} \in \mathcal{B}_t$ with top K most probable y_{t+1} and add all the resulting $Y_{0:t+1}$ into $\mathcal{B}_{t+1}$. $\mathcal{B}_{t+1}$ is then filtered to keep only the top K active paths. We move all paths in $\mathcal{B}_{t+1}$ end with an <END> label to candidate set C and remove all $Y_{0:t+1}$ from $\mathcal{B}_{t+1}$ such that

$$p(Y_{0:t+1}|X; \theta) < p(\bar{Y}|X; \theta), \quad \forall \bar{Y} \in C. \quad (10)$$

The algorithm terminates when the probability of active paths in $\mathcal{B}_t$ is smaller than that of all the completed candidate paths in C. The inference algorithm is summarized in Algorithm 1.

ALGORITHM 1: Inference LEAP with beam search.

Input: Input X, model $p(X|Y;\theta)$, beam size K;
Output: Estimated best output Y^*;
Initialize candidate set C;
Initialize beam $\mathcal{B}[0]$ with $Y_{0:0} = (\text{<START>})$;
$t \leftarrow 0$;
while $\mathcal{B}[t] \neq \emptyset$ **do**
$\quad$ **foreach** $Y_{0:t} \in \mathcal{B}[t]$ **do**
$\quad\quad$ $\{Y_{0:t+1}\} \leftarrow$ extend $Y_{0:t}$ with $y_{t+1} \in \mathcal{Y}$;
$\quad\quad$ Append topK($\{Y_{0:t+1}\}$) into $\mathcal{B}[t+1]$;
$\quad$ **end**
$\quad$ $\mathcal{B}[t+1] \leftarrow$ topK($\mathcal{B}[t+1]$);
$\quad$ Move all paths in $\mathcal{B}[t+1]$ end with <END> to C;
$\quad$ Filter $\mathcal{B}[t+1]$ by Eq. (10) ;
$\quad$ $t \leftarrow t+1$;
end
$Y^* \leftarrow \arg\max_{Y \in C} p(Y|X;\theta)$;
return Y^*;

4.3 Reinforcement Fine-Tuning

The basic LEAP has the capability to capture the label dependency and instance-label correlation. However, there are two limitations: (1) due to the lack of a negative training sample, it is still hard to avoid all of the adverse drug interactions in the recommended treatment plans. (2) according to our empirical results, the model tends to generate incomplete medication sequences. To address these limitations, we propose fine-tuning the model via reinforcement learning by leveraging clinical evidences from external knowledge bases. Assume we have a list of known adverse drug interaction pairs $\mathcal{K}$ (both drug to drug interaction and drug to disease interaction). To avoid adverse drug interaction, one simple heuristic is to perform post-processing by removing conflicted drugs from the recommendation. However, this will often hurt the coverage of the treatment recommendations towards all the diseases the patient has. Thus, instead, we try to directly fine-tune the model parameter θ to prevent the model from generating conflict drugs. We address this problem using model-free policy-based Reinforcement Learning [31]. We regard the above basic LEAP model $p(Y|X;\theta)$ as a pre-trained parametrized stochastic policy. The objective is to maximize the expected reward of the treatment set Y recommended by the policy:

$$J(\theta|X) = \mathbb{E}_{Y \sim p(Y|X;\theta)}[R(X, Y, \hat{Y})], \quad (11)$$

where $R(X, Y, \hat{Y})$ is a scalar value reward function that evaluates the quality of Y, $\hat{Y}$ is the treatment set for X prescribed by doctors (i.e. from the EHR data). We design the reward so that it encourage the model to avoid adverse interactions. We define an evaluator $\mathcal{K}(X, Y)$ such that

$$\mathcal{K}(X, Y) = \begin{cases} 0, & \text{if } \exists(x_i, y_j) \in \mathcal{K}, \ \forall x_i \in X, \ \forall y_j \in Y \\ 0, & \text{if } \exists(y_i, y_j) \in \mathcal{K}, \ \forall y_i, y_j \in Y \\ 1, & \text{otherwise} \end{cases} \quad (12)$$

where X is a input diagnosis, Y is the corresponding treatment. The reward is formulated as follows:

$$R(X, Y, \hat{Y}) = \delta(Y, \hat{Y}) \times \mathcal{K}(X, Y), \quad (13)$$

where $\delta(\cdot, \cdot)$ is a similarity measure. We choose Jaccard coefficient as the similarity measure because it balances the accuracy and completeness. In this way, a recommended treatment gets zero reward if it contains adverse interaction, otherwise it gets a positive reward based on its similarity to the corresponding prescription in the data. We use policy gradient [31] to optimize the parameters. The gradient of Equation (11) is given by

$$\nabla_\theta J(\theta|X) = \mathbb{E}_{Y \sim p(Y|X;\theta)}[R(X, Y)\nabla_\theta \log p(Y|X;\theta)] \quad (14)$$

We then update parameters as: $\theta \leftarrow \theta + \sigma \nabla_\theta J(\theta|X)$, where $\sigma \in \mathbb{R}^+$ denotes the learning rate. Other advanced optimization algorithm such as ADAM can be also adopted here.

In summary, Algorithm 2 shows the overview of our solution.

ALGORITHM 2: Fine-tune LEAP with reinforcement

Input: Training set **R**, number of training steps N;
Output: Fine tuned model parameters θ;
Initialize model parameter θ;
Pre-train basic LEAP $p(Y|X;\theta)$ on **R** by Eq. (8) using SGD;
for $i = 1$ to N **do**
$\quad$ Sample pair $(X, \hat{Y})$ from **R**;
$\quad$ $Y^* \sim p(Y|X;\theta)$;
$\quad$ Calculate $R(X, Y^*, \hat{Y})$ by Eq. (13);
$\quad$ Calculate $\nabla_\theta J(\theta|X)$ by Eq. (14);
$\quad$ $\theta \leftarrow \theta + \sigma \nabla_\theta J(\theta|X)$;
end
return θ;

5 EXPERIMENTS

In this section, we demonstrate the effectiveness of our model. Our experiments are conducted real EHR datasets, some are public available. First we describe the datasets and then present the results. We make the source code of LEAP publicly available at https://github.com/neozhangthe1/AutoPrescribe.

5.1 Data

We carry out experiments on two datasets, namely *MIMIC-3* and *Sutter*.

MIMIC-3. The MIMIC-3 dataset [18] is a publicly available dataset consisting of medical records of 40K intensive care unit (ICU) patients over 11 years. It consists of 50,206 medical encounter records that associated with 6,695 distinct diseases and 4,127 drugs.

Sutter. This dataset from Sutter Palo Alto Medical Foundation (PAMF) consists of 18-years longitudinal medical records of 258K patients between age 50 and 90. It contains 2,415,414 medical encounters associated with 8,359 distinct diseases and 7,516 drugs. Average number of diseases and drugs per record are 2.97 and 1.75 respectively.

Drugs in Sutter and MIMIC-3 are encoded using GPI[4] and NDC[5] codes, respectively. The GPI coding is inherently a multi-level ontology that identifies drugs from their primary therapeutic use down to unique interchangeable product regardless of manufacturer or package size. To conduct experiments at different granularity, we use the first and third level of GPI codes, resulting 93 and 982

---

[4]http://www.wolterskluwercdi.com/drug-data/medi-span-electronic-drug-file/
[5]http://www.fda.gov/Drugs/InformationOnDrugs/ucm142438.htm

distinct drug groups, respectively. We convert drugs in MIMIC-3 to GPI code using an open-source software from OHDSI[6]. The diagnoses in both datasets are represented using ICD-9 code[7]. In order to learn robust and nontrivial mapping between diseases and medications, We extract the records from each dataset with more than two diagnosis codes and filtered the records to include only top 2,000 most common diagnosis codes, which covers 95.3% of all records.

5.2 Baseline Comparison Methods

We compare our method with the following baselines:

- **Rule-based:** This method recommends drugs based on an existing drug to disease mapping from the MEDI database [36]. For each disease, one of the drugs mentioned in the mapping is assigned.

- **K-Most frequent:** This is a simple baseline that retrieves the top K medications that most frequently co-occur with each disease as their treatment. We set $K = 1$ on Sutter dataset and $K = 3$ on MIMIC-3 dataset according to the performance on validate set[8].

- **Softmax MLP:** We learn a multi-label classifier using a multi-layer perceptron with softmax output layer. Our implementation uses a 3-layer MLP. A global threshold is used to select positive medications. The value of the threshold and hyper parameters are tuned on a validation set using grid search.

- **Classifier Chains:** Classifier Chains [28] is a popular multi-label learning method that models the correlation between labels by a feeding both input and previous classification results into the latter classifiers. We use a multi-hot vector to encode input diagnosis set X and leverage SVM as binary classifiers for each label.

- **LEAP:** This is our proposed methods: Basic LEAP models label-instance mapping and label dependency as described in Section 4.2. We then evaluate full LEAP with reinforcement learning fine-tuning as described in Section 4.3. We implemented LEAP with Theano and used ADAM for model training,

5.3 Experimental Results

To evaluate the performance of treatment recommendations, we measure quantitative as well as qualitative performance metrics. Quantitative measures include prediction accuracy as well as measures related to drug-drug interactions. Qualitative measures include assessments of clinical meaningfulness, completeness, and avoidance of drug-drug interactions. For all experiments, the diagnosis and medications was divided into training, validation and test sets in a 0.7:0.1:0.2 ratio. The validation set was used to determine hyper-parameters.

5.3.1 Quantitative Performance: We first evaluate how closely the generated prescription compares against the corresponding prescription of medications (for a disease set) written by doctors. We

---

	Sutter		MIMIC-3	
Granularity	1	3	1	3
Rule-based	0.3207	0.2770	0.2753	0.2354
K-Most frequent	0.4283	0.3181	0.2609	0.2616
Softmax MLP	0.4908	0.3739	0.4897	0.3342
Classifier Chains	0.4839	0.3620	0.4621	0.3204
Basic LEAP	0.5270	0.3936	0.5107	0.3865
LEAP	**0.5341**	**0.4073**	**0.5582**	**0.4342**

Table 1: Treatment recommendation performance on Sutter and MIMIC3 dataset. We evaluate the experimental results in terms of Jaccard Coefficient. Granularity indicates the level of GPI medication code we are using.

utilize Mean Jaccard Coefficient to measure the performance. Assume X_i is the ith input diagnosis, Y_i is the treatment set generated by the algorithm, and $\hat{Y}^i$ is the doctor prescription in the data. The Jaccard coefficient is defined as the size of the intersection divided by the size of the union of ground truth label set and predicted label set.

$$\text{Jaccard} = \frac{1}{K} \sum_i^K \frac{|Y_i \cap \hat{Y}_i|}{|Y_i \cup \hat{Y}_i|},$$

where K is the number of samples in test set.

Table 1 shows the performance of the aforementioned performance metrics on Sutter and MIMIC-3 dataset. The rule-based method is not effective because the drug to disease mapping simply aggregates all the related drugs for a disease, which cannot be accurately tailored to individual patients. The K-Most frequent method works poorly on the MIMIC-3 dataset because the number of drugs and diseases associated with each encounter is large due to high severity of patients at ICU, and thus the drug co-occurring most frequently with a disease may not be an effective treatment. Moreover, both Rule-based and K-most frequent greedily select a constant number of drugs for each disease in the diagnosis without considering any context information which can lead to unwanted redundancy and adverse drug interaction.

Both Softmax MLP and Classifier Chains assume equal contribution of all the diseases in one encounter record. The representation of diagnoses is simply a reduction of linear embeddings of different instances which will cause information loss. Classifier Chains considers the label dependency by incorporating former predictions into the input of latter classifiers, but the performance is constrained by the expressiveness of the model.

LEAP consistently outperforms other methods by 10+% on both datasets with respect to Mean Jaccard Coefficient. The reason is that LEAP effectively captures the label dependency and label to instance mapping. The reinforcement fine-tuning provides significantly additional improvement of 3.4% on the Sutter dataset and an extra 12.3% on the MIMIC-3 dataset, compared to basic LEAP. Figure 5 illustrates the average reward obtained by each sample with respect to training epochs, which confirmed that LEAP is able to progressively find better treatment recommendation policy over training iterations.

Prior work [3] used micro-averaged area under ROC curve (AUC) as primary measure to evaluate the performance of drug prediction.
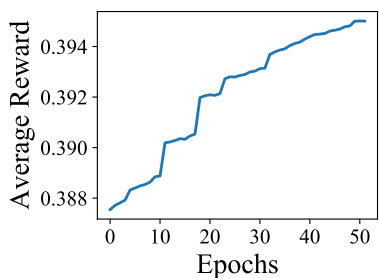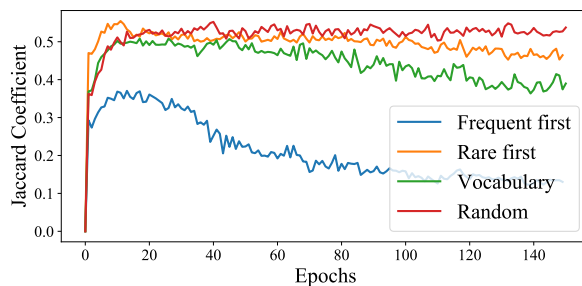
(a) Sutter Dataset
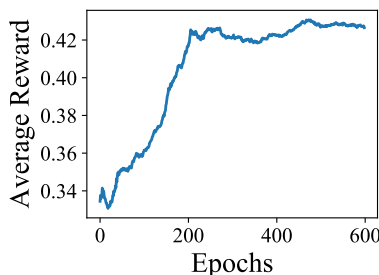
(b) MIMIC-3 Dataset

Figure 5: Average reward obtained by each recommendation w.r.t training epochs on Sutter (5a) and MIMIC-3 (5b) dataset.
.

(a) Sutter Dataset

(b) MIMIC-3 Dataset

Figure 6: Performance w.r.t training epochs on Sutter (6a) and MIMIC-3 (6b) dataset with different label order.

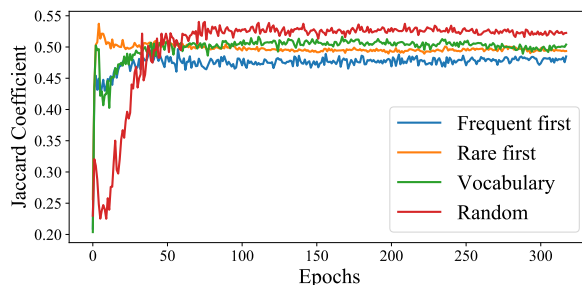| Method	Drug Interaction Rate
K-Most frequent	12.06%
Softmax MLP	3.51%
Basic LEAP	2.41%
LEAP	0.23%

Table 2: Evaluation of the ability of avoiding adverse drug interactions. The table shows the proportion of treatment plans recommended by different methods that contains drug to drug interaction mentioned in the database.

We want to argue that AUC is not an appropriate measure for the treatment recommendation problem. The reason is that instead of outputting a ranked list of drugs, we need to generate a combination of drugs that is both effective and avoid of redundancy and adverse interactions. Empirically we observe that Softmax MLP can achieve a near perfect AUC, but this does not necessarily indicate that the recommended treatment is appropriate since selecting a proper threshold for each record is non-trivial. LEAP can accurately predict the <END> label to determine the completeness of a treatment set.

5.3.2 Different Order of Labels. To address the label ordering issue described in Section 4.2, we test the robustness of LEAP to different label ordering, Figure 6 illustrates the performance of LEAP with different ordering heuristics over training epochs. We can see that *Frequent first* performs poorly on both datasets. The reason is that by constraining the model to predict more frequent drugs

first, the generated treatment plans are likely to be predominated by non-informative popular drugs. An interesting observation is that random shuffled order converges slowly at first, but eventually outperforms the alternative labeling strategies. Note that rare first actually performed quite well especially regarding to the initial convergence speed, because it essentially upsampled the rare medications so that a robust representation can be learned for them.

5.3.3 Avoiding Drug-Drug Interactions. To evaluate the ability of avoiding adverse drug effect of our model, we compare the prevalence of drug-drug interactions in the treatment recommendations given by the baselines in against LEAP, which considers drug to drug interaction information using reinforcement fine-tuning described in Section 4.3. We collect known drug to drug interaction pairs by crawling an online Drug interaction Database[9]. In table 2 we show the percentage of recommendations containing unfavorable drug-drug interactions (i.e., one drug impedes the efficacy of another drug, two drugs interacting together lead to adverse effects) on the Sutter dataset, comparing LEAP against baselines. The result shows that reinforcement fine-tuning effectively removes 90+% of adverse drug interactions in the recommended treatment plan without hurting the effectiveness.

5.3.4 Qualitative Evaluation. We also invited a clinical expert to manually score the results. We randomly chose 100 diagnosis sets and corresponding medication treatment sets identified with our algorithms and baseline methods. For each diagnosis set, a

---

[9]https://www.drugs.com/drug_interactions.php

Diagnosis	Methods	Recommended Treatments
Type 2 diabetes Hyperlipidemia Depressive disorder Hypertension	*K*-Most frequent	PEG KCl Bicarb, Quinapril, Pravastatin, Metformin, Paroxetine
	Softmax MLP	Metformin
	Basic LEAP	Metformin, Quinapril, Pravastatin, Paroxetine
	LEAP	Metformin, Amiloride/HCTZ, Fenofibrate, Paroxetine
Depressive Disorder Acute bronchitis Imbalance (gait)	*K*-Most frequent	Azithromycine, Privastatin, Paroxetine
	Softmax MLP	Paroxetine, Azithromycin
	Basic LEAP	Azithromycin
	LEAP	Azithromycin, Dextromethorphan-K Guaiacolsulfonate, Paroxetine

Table 3: Example Recommended Treatments by Different Methods
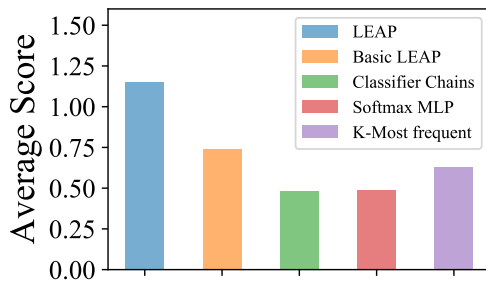

Figure 7: Average meaningfulness score given to recommendations from each method. Scores range from 0 to 2, where higher scores are better (n=100 recommendations evaluated).

clinical domain expert (RC, MD/PhD in training) scored the recommended medications in the following way: a score of 2 is given if the medication set is complete (addresses all diagnoses) and does not yield unfavorable drug-drug interactions; a score of 1 is given if the medication set is partially complete (addresses at least 50% of the diagnoses) and does not yield unfavorable drug-drug interactions; a score of 0 is given if the medication either addresses less than 50% of the diagnoses or possesses negative drug-drug interactions. Figure 7 shows the average rating of meaningfulness of recommended treatments given for each of the 5 different methods. LEAP yielded the highest mean score of meaningfulness, and exhibits a 130+% score improvement over the Classifier chains and Softmax MLP. LEAP further improves over the basic model by 50+%, hence validate the effectiveness of reinforcement fine-tuning.

5.3.5 Case Study. In table 3 we show two examples of treatment recommendations made for patients multiple diseases. LEAP performs favorably in two situations when comparing it against the other baselines. For the first patient, LEAP recommended a set of treatments with 100% coverage, with Metformin for Type 2 diabetes, Amiloride/HCTZ for Hypertension, Fenofibrate for Hyperlipidemia, and Paroxetine for depressive disorder. In contrast, the Classifier chains only picked up Metformin, thus only targeting diabetes. The Softmax MLP only picked up Maroxetine, thus only targeting depressive disorder. In the second patient in the case study, we observe that the drugs recommended by the *K*-Most frequent method yielded unfavorable drug-drug interactions. Privastatin and Azithromycin can lead to harmful, potentially fatal

effects when taken together including rhabdomyolysis and renal failure. The combination of drugs recommended by LEAP was able to target the diseases present in the patient, while simultaneously avoiding the unfavorable drug-drug interaction.

6 CONCLUSION

In this paper, we propose LEAP, an end to end learning algorithm for treatment recommendation that jointly models drug disease mapping and drug drug interaction. LEAP decompose treatment recommendation task into a sequential decision making process. A recurrent decoder is used to model label dependencies and content-based attention is used to capture is used to learn the disease-drug mapping. We further leverage reinforcement learning to fine-tune the model to ensure accuracy and completeness. External clinical knowledge is incorporated into the design of reinforcement reward and effectively prevent adverse drug combinations. Throughout experiments, we successfully demonstrated the superior performance of LEAP by 10+% over all baselines.

REFERENCES

[1] Daniel Almirall, Scott N Compton, Meredith Gunlicks-Stoessel, Naihua Duan, and Susan A Murphy. 2012. Designing a pilot sequential multiple assignment randomized trial for developing an adaptive treatment strategy. *Statistics in medicine* 31, 17 (2012), 1887–1902.
[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
[3] Jacek Bajor and Thomas Lasko. 2017. Predicting medication from diagnosis codes with recurrent neural networks. *ICLR* (2017).
[4] Sanghamitra Bandyopadhyay, Dip Ghosh, Ramkrishna Mitra, and Zhongming Zhao. 2014. MBSTAR: multiple instance learning for predicting specific functional binding sites in microRNA targets. *Scientific reports* 5 (2014), 8004–8004.
[5] Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. 2004. Learning multi-label scene classification. *Pattern recognition* 37, 9 (2004), 1757–1771.
[6] Forrest Briggs, Xiaoli Z Fern, and Raviv Raich. 2012. Rank-loss support instance machines for MIML instance annotation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 534–542.
[7] Forrest Briggs, Xiaoli Z Fern, Raviv Raich, and Qi Lou. 2013. Instance annotation for multi-instance multi-label learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 7, 3 (2013), 14.

[8] Zhuo Chen, Kyle Marple, Elmer Salazar, Gopal Gupta, and Lakshman Tamil. 2016. A Physician Advisory System for Chronic Heart Failure Management Based on Knowledge Patterns. *CoRR* abs/1610.08115 (2016). http://arxiv.org/abs/1610.08115

[9] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence* 89, 1 (1997), 31–71.

[10] Ralph B DâĂŹAgostino. 2007. Estimating treatment effects using observational data. *Jama* 297, 3 (2007), 314–316.

[11] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. 2008. Multilabel classification via calibrated label ranking. *Machine learning* 73, 2 (2008), 133–153.

[12] Nadia Ghamrawi and Andrew McCallum. 2005. Collective multi-label classification. In *Proceedings of the 14th ACM international conference on Information and knowledge management.* ACM, 195–200.

[13] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on.* IEEE, 6645–6649.

[14] Meredith Gunlicks-Stoessel, Laura Mufson, Ana Westervelt, Daniel Almirall, and Susan Murphy. 2016. A pilot SMART for developing an adaptive treatment strategy for adolescent depression. *Journal of Clinical Child & Adolescent Psychology* 45, 4 (2016), 480–494.

[15] Yuhong Guo and Suicheng Gu. 2011. Multi-Label Classification Using Conditional Dependency Networks. In *Twenty-Second International Joint Conference on Artificial Intelligence.*

[16] Karin Hannes, Marcus Leys, Etienne Vermeire, Bert Aertgeerts, Frank Buntinx, and Anne-Marie Depoorter. 2005. Implementing evidence-based medicine in general practice: a focus group based study. *BMC family practice* 6, 1 (2005), 1.

[17] Sheng-Jun Huang, Wei Gao, and Zhi-Hua Zhou. 2014. Fast Multi-Instance Multi-Label Learning. In *Twenty-Eighth AAAI Conference on Artificial Intelligence.*

[18] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3 (2016).

[19] Himabindu Lakkaraju and Cynthia Rudin. 2016. Learning Cost-Effective and Interpretable Regimes for Treatment Recommendation. *arXiv preprint arXiv:1611.07663* (2016).

[20] Jason Lazarou, Bruce H Pomeranz, and Paul N Corey. 1998. Incidence of adverse drug reactions in hospitalized patients: a meta-analysis of prospective studies. *Jama* 279, 15 (1998), 1200–1205.

[21] Yu-Feng Li, Ju-Hua Hu, Yuan Jiang, and Zhi-Hua Zhou. 2012. Towards discovering what patterns trigger what labels. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence.* AAAI Press, 1012–1018.

[22] Weiwei Liu and Ivor Tsang. 2015. On the optimality of classifier chain for multi-label classification. In *Advances in Neural Information Processing Systems.* 712–720.

[23] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. 2015. Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). *ICLR* (2015).

[24] James J McGough and Stephen V Faraone. 2009. Estimating the size of treatment effects: moving beyond p values. *Psychiatry (1550-5952)* 6, 10 (2009).

[25] Volodymyr Mnih, Nicolas Heess, Alex Graves, and others. 2014. Recurrent models of visual attention. In *Advances in neural information processing systems.* 2204–2212.

[26] Maria Panagioti, Jonathan Stokes, Aneez Esmail, Peter Coventry, Sudeh Cheraghi-Sohi, Rahul Alam, and Peter Bower. 2015. Multimorbidity and Patient Safety Incidents in Primary Care: A Systematic Review and Meta-Analysis. *PLOS ONE* 10, 8 (08 2015), 1–30. DOI:http://dx.doi.org/10.1371/journal.pone.0135947

[27] Anh Pham, Raviv Raich, Xiaoli Fern, and Jesús P Arriaga. 2015. Multi-instance multi-label learning in the presence of novel class instances. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15).* 2427–2435.

[28] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2009. Classifier chains for multi-label classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer, 254–269.

[29] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM Neural Networks for Language Modeling. In *Thirteenth Annual Conference of the International Speech Communication Association.*

[30] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems.* 3104–3112.

[31] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Advances in Neural Information Processing Systems.* 1057–1063.

[32] Marjan van den Akker, Frank Buntinx, and J André Knottnerus. 1996. Comorbidity or multimorbidity: what's in a name? A review of literature. *The European Journal of General Practice* 2, 2 (1996), 65–70.

[33] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2015. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391* (2015).

[34] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 3156–3164.

[35] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. 2016. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2285–2294.

[36] Wei-Qi Wei, Robert M Cronin, Hua Xu, Thomas A Lasko, Lisa Bastarache, and Joshua C Denny. 2013. Development and evaluation of an ensemble resource linking medications to their indications. *Journal of the American Medical Informatics Association* 20, 5 (2013), 954–961.

[37] Jiajun Wu, Yinan Yu, Chang Huang, and Kai Yu. 2015. Deep multiple instance learning for image classification and auto-annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 3460–3469.

[38] Yeong-Yuh Xu. 2016. Multiple-instance learning based decision neural networks for image retrieval and classification. *Neurocomputing* 171 (2016), 826–836.

[39] Xiangyang Xue, Wei Zhang, Jie Zhang, Bin Wu, Jianping Fan, and Yao Lu. 2011. Correlative multi-label multi-instance image annotation. In *Computer Vision (ICCV), 2011 IEEE International Conference on.* IEEE, 651–658.

[40] Min-Ling Zhang and Zhi-Hua Zhou. 2014. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering* 26, 8 (2014), 1819–1837.

[41] Zhi-Hua Zhou and Min-Ling Zhang. 2007. Multi-instance multi-label learning with application to scene classification. (2007).

[42] Zhi-Hua Zhou, Min-Ling Zhang, Sheng-Jun Huang, and Yu-Feng Li. 2012. Multi-instance Multi-label Learning. *Artif. Intell.* 176, 1 (Jan. 2012), 2291–2320. DOI:http://dx.doi.org/10.1016/j.artint.2011.10.002

[43] Wentao Zhu, Qi Lou, Yeeleng Scott Vang, and Xiaohui Xie. 2016. Deep Multi-instance Networks with Sparse Label Assignment for Whole Mammogram Classification. *arXiv preprint arXiv:1612.05968* (2016).