Online Transfer Learning for Differential Diagnosis Determination

Jie Xu[*†], **Daby Sow**[†], **Deepak Turaga**[†], **Mihaela van der Schaar**[*]
[*]University of California at Los Angeles, Los Angeles, California 90095
[†]IBM T. J. Watson Research Center, Yorktown Heights, New York 10598

Abstract

In this paper we present a novel online transfer learning approach to determine the set of tests to perform, and the sequence in which they need to be performed, in order to develop an accurate diagnosis while minimizing the cost of performing the tests. Our learning approach can be incorporated as part of a clinical decision support system (CDSS) with which clinicians can interact. The approach builds on a contextual bandit framework and uses online transfer learning to overcome limitations with the availability of rich training data sets that capture different conditions, context, test results as well as outcomes. We exploit context information to incrementally learn the performance of different policies (sequences of tests) on different clusters of the target population, and use that to develop context-specific policies to reduce the diagnosis cost. We provide confidence bounds for our recommended policies, which is essential in order to build the trust of clinicians. We evaluate the algorithm against different transfer learning approaches using both simulated datasets and on real-world patient alarm datasets collected from Neurological Intensive Care Units (with reduced costs by 20%). We also discuss the implications of these improved results on patient diagnosis, and the applicability of the approach in other domains.

Introduction

Recent advances in sensing and measurement technologies are enabling us to monitor complex human, engineered, physical, biological and chemical systems and processes in many sophisticated ways. This enables improved ability to understand the state of health of these systems, diagnose problems, and use this to design interventions to maximize health at varying timescales. However, while several such measurements can be made (e.g. by performing different tests on a patient), the decision on which test to perform and when to perform it remains a very challenging problem.

Challenges stem from multiple factors: i) There are complex relationships between different attributes that are being measured. ii) Tests have varying degrees of costs associated with them (e.g. some tests are very expensive). iii) Tests are significantly impacted by *context*, i.e. the *best* set of tests, measurements and interventions may be different depending on the context in which it takes place. For instance in a medical diagnosis scenario, depending on their age and gender, two different patients that exhibit the same apparent set of symptoms may require a very different set of tests (with different costs and accuracies). iv) The determination of tests is often challenged by the limited access to relevant data. For instance, existing patients datasets often have distributions that do not necessarily capture the information needed for the accurate diagnosis in a novel problem domain. Thus, the resulting diagnosis policies may perform poorly. This prompts the need for a system that can effectively perform context-specific diagnosis that maximizes diagnosis accuracy and minimizes test costs even when highly relevant data pertaining to the diagnosis decision is missing.

A large body of research in machine learning and planning has been dedicated to solving a variety of diagnosis problems as well as providing decision support systems for professionals (see Related Work section). However, many of the existing techniques do not address many of the aforementioned challenges. In this paper, we present a novel decision support system that addresses these challenges by transferring knowledge from multiple related problem domains and incrementally learning the best policies (i.e. sequences of test) to adopt depending on the context of the diagnosis problem. These contexts can be exogenous facts or metadata about the problem. In the medical setting, they can be patient's age, gender and weight. Note that the contexts are different than the endogenous testing results. The use of multiple related problem domains enables transferring knowledge from the most relevant domains for different diagnosis contexts; it also creates a way to measure the semantic similarity between contexts: contexts are similar if their most related existing domains are the same. The learned semantic similarity is then used to develop context-specific solutions in the novel problem domain. The proposed approach is able to provide diagnosis confidence bounds which are important to ensure the trust of domain professionals.

Related Work

Support systems for decision making have been extensively studied and we do not aimed to provide here a comprehensive overview. Instead, we will focus on the most related research. A first strand of related research focuses on cost-sensitive learning and can be roughly divided into various categories. The first category formalizes the diagnosis problem as a classification problem and aims to minimize misclassification errors while ignoring the test costs (Turney 2000)(Greiner, Grove, and Roth 2002)(Ling et al. 2004). The second category jointly considers the misclassification errors and test costs. A majority of works in this category solve the diagnosis problem using decision tree induction algorithms (Núñez 1991)(Tan 1993)(Norton 1989), while others use Markov Decision Processes (MDP) (Zubek, Dietterich, and others 2004). Several heuristics are developed in (Zubek, Dietterich, and others 2004) to reduce overall problem complexity and over-fitting when the datasets are small. A disadvantage of these approaches is that they rely on training datasets to learn the appropriate model. In this paper, we also model the diagnosis problem as an MDP but our focus is on how to overcome the lack of initial training data by using transfer of knowledge from relevant datasets.

Transfer learning is used to improve learning in a target task (i.e. the task for which the solution/policy needs to be learned) by leveraging existing knowledge from different source tasks (i.e. tasks for which the dataset is sufficiently large and the solution/policy has already been derived). The majority of this literature assumes a single source task from which knowledge can be transferred. (Marx et al. 2005)(Raina, Ng, and Koller 2006)(Singleton et al. 2014). Transfer learning from multiple source tasks is much more challenging; most works aiming to address this problem focus on classification problems (Duan et al. 2009)(Gao et al. 2008)(Ge et al. 2014)(Luo et al. 2008)(Yao and Doretto 2010). Moreover, the target classifier is built offline using a training dataset for the target problem. In contrast, in our considered setting, the target data arrives sequentially and the features are not given but need to be discovered (by performing various tests). In the scenario when the task data becomes available sequentially online, imitation-type transfer learning techniques are often adopted where source policies are applied to the target task initially, while the target task solution is learned gradually (Madden and Howley 2004)(Fernández and Veloso 2006)(Torrey et al. 2008b)(Torrey et al. 2008a) However, such works only consider the availability of a single source task, while our work focuses on multiple source tasks.

Our solution builds on the contextual bandit framework (Slivkins 2009)(Li et al. 2010)(Tekin, Zhang, and van der Schaar 2014) to boost learning using policies from source tasks (e.g. similar diseases or datasets of patients with a similar demography) but provides numerous innovations. For instance, while conventional works on multi-armed bandits focus on learning the best policy (or policies) among an fixed set, our algorithm uses the learned semantic similarity between contexts to produce new context-specific target policies (which may be distinct from existing policies) using the data accumulated so far for the target domain.

Finally, there has also been work on AI planning based diagnosis (Sohrabi, Baier, and McIlraith 2010) that models the process being monitored as a discrete dynamical system with an associated underlying state transition representation. These approaches use planning based algorithms to develop

	Decision Tree/MDP	Transfer Learning	Contextual Bandits	This Paper
Training dataset not required	*No*	*No*	*Yes*	*Yes*
Context Info	*No*	*No*	*Yes*	*Yes*
Multiple source knowledge	*No*	*Some*	*No*	*Yes*
Target policy produced using new data	*Yes*	*Yes*	*No*	*Yes*
Confidence bound	*No*	*No*	*Yes*	*Yes*

Table 1: Comparison with existing works.

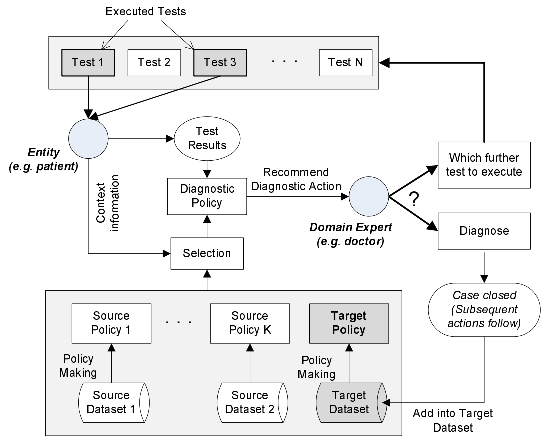

Figure 1: Computer-aided diagnosis system

diagnoses in the form of posited events that explain a given set of observations. These approaches do not focus on developing the schedule of tests and measurements to perform; instead they observe the results of these tests and aim to determine the most appropriate diagnosis. In our paper, we focus on designing the diagnosis policy that determines the sequence of tests that maximize the diagnosis accuracy and minimize the test cost.

Table 1 summarizes the comparison with existing works.

Computer-aided Diagnosis System

Figure 1 depicts a generic computer-aided diagnosis system designed to help domain experts provide accurate diagnosis in a cost effective way. In a cyber-security setting, the domain expert is a cyber-analyst attempting to diagnose problems in a computer network in a cost effective manner. In a healthcare setting, the domain expert is a physician. The entity being analyzed for diagnosis is a patient. We use $\theta \in \Theta$ to denote the initial exogenous meta information about the entity being analyzed, such as the patient's basic symptoms and personal medical profile (e.g. gender, age, weight, medical history etc.). We call θ the *context* information about the entity. In our problem formulation, we assume that the set of possible tests that can be ordered by the domain expert is finite. Let $\mathcal{Q} = \{1, 2, ..., N\}$ denote this space, $N < \infty$. We further assume that each test $q \in \mathcal{Q}$ has a finite set of possible results, denoted by $\mathcal{O}_q$. We also define an "unknown" test result to be assigned to tests that have not been performed and that do not provide any information that can be used for diagnosis.

At any point in time, an entity is assigned with a state

s that represents the known test results that have been performed on that entity. For instance, at any point in time, a patient may be associated with a state $\boldsymbol{s}$ reflecting the knowledge gained by performing a series of medical tests on him/her. In this case, this state does not reflect the patient's medical condition but rather the knowledge about the patient with respect to the medical tests. This state evolves as more medical tests are executed. Specifically, the cardinality of a state $\boldsymbol{s}$ is N and each entry s_q is the test result to test q. Let $\mathcal{S}$ denote the state space. The initial state of an entity appearing in our system is $s_q^{\text{init}} = \text{unknown}, \forall q \in \mathcal{Q}$. Depending on the current entity state $\boldsymbol{s}$, the computer-aided diagnostic system either recommends new tests to be performed to extract more knowledge needed for an accurate diagnosis or recommends a diagnosis decision if it has enough information about the entity. Let the action space be $\mathcal{A} = \{\mathcal{Q}, \mathcal{D}\}$ where $a \in \mathcal{Q}$ represents the recommendation to execute a new test and $a \in \mathcal{D}$ represents a recommended diagnosis; they are kept fixed. We assume that if the expert follows $a \in \mathcal{D}$, then the diagnosis for the current entity case is closed and subsequent intervention actions follow. Let $\emptyset$ be a special terminal state which denotes that the case is closed. For an entity k, let $\{q_1^k, ..., q_{n^k}^k\}$ be the sequence of tests that are executed and d^k be the final diagnosis decision. The diagnosis cost c^k for this entity is defined as $c^k = \sum\limits_{i \in \{1,...,n^k\}} c^k(q_i^k) + \beta c^k(d^k)$ where $c^k(q_i^k), \forall i = 1, ..., n^k$ are the costs incurred by executing the tests, $c^k(d^k)$ is the costs due to incorrect diagnosis and $\beta \in [0, 1]$ is a trade-off factor. Our goal is to develop diagnosis policies that minimize the diagnosis cost.

A diagnostic policy is defined as a set of actions that are recommended to the domain expert in the various states. Specifically, a policy is denoted by $\pi = \{a(\boldsymbol{s})\}_{\boldsymbol{s} \in \mathcal{S}}$. Hence, given a diagnostic policy, after observing the entity state, the diagnostic system can recommend an action to the domain expert. Note that during a medical diagnosis process, these actions are merely recommendations to the experts who may follow the recommendation or choose different actions based on their own domain expertise and experience. However, we will make the assumption in the remaining part of this paper that the domain expert always follows the recommendation to facilitate our analysis.

At any point in time, the execution of tests on entities provides additional information on the entities. Such state transitions are probabilistic and specific to the domain. Let $p(\boldsymbol{s}'|\boldsymbol{s}, q)$ denote the transition probability from state $\boldsymbol{s}$ to $\boldsymbol{s}'$ when test q is executed. Note that $\boldsymbol{s}$ and $\boldsymbol{s}'$ should differ only in the q-th entry; otherwise $p(\boldsymbol{s}'|\boldsymbol{s}, q) = 0$. Since taking an action $d \in \mathcal{D}$ always leads to a diagnosis and closes the current case, we have $p(\phi|\boldsymbol{s}, d) = 1, \forall d \in \mathcal{D}$. Let $c(q|\boldsymbol{s})$ denote the *expected* cost of performing test q on entities in state $\boldsymbol{s}$. Let $c(d|\boldsymbol{s})$ denote the *expected* cost of making a diagnose d on entities in state $\boldsymbol{s}$. We unify these two types of costs in a cost function $c : \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ as a mapping from the action space and the state space to a real value. In sum, we call the set of transition probabilities p and the diagnosis cost function c the *problem parameters*. These parameters are Markovian; they depend only on the last state. This is a reasonable approximation since a state represents all the knowledge revealed about the entity so far.

Diagnostic policy construction

The optimal diagnostic policy that minimizes the expected diagnostic cost in each state is defined using the Bellman equation: $J(a|\boldsymbol{s}) = c(a|\boldsymbol{s}) + \sum\limits_{\boldsymbol{s}'} p(\boldsymbol{s}'|\boldsymbol{s}, a) V(\boldsymbol{s}')$ where $V(\boldsymbol{s}') = \min_{a'} J(\boldsymbol{s}'|a')$. Thus $\pi^{opt} = \{a^{opt}(\boldsymbol{s})\}_{\boldsymbol{s} \in \mathcal{S}}$ such that $\forall \boldsymbol{s}$, $a^{opt}(\boldsymbol{s}) = \arg\min_a J(a|\boldsymbol{s})$. Since the entity comes with the initial state $\boldsymbol{s}^{\text{init}} = \{\text{unknown}\}_{q \in \mathcal{Q}}$, the expected diagnostic reward is $V(\boldsymbol{s}^{\text{init}})$. With abuse of notation, we let $V(\pi) = V(\boldsymbol{s}^{\text{init}}|\pi)$ denote the diagnostic cost by using π.

The optimal diagnostic policy making problem can be solved by backward induction using the estimated problem parameters from an existing dataset. This is because when the entity is in a level-n state (i.e. a state in which n test results are revealed), executing any not yet executed tests makes the state transition into a level-$(n+1)$ state. Since state space size is exponential in the test set size, the complexity grows as the number of tests increases. Reducing the solution complexity of this problem is not the main focus of the present paper; we refer readers interested in this topic to existing work that provides efficient heuristics algorithms such as (Zubek, Dietterich, and others 2004)(Bonet and Geffner 2012).

Transfer Learning in Diagnosis

One of the key challenges for many diagnosis systems is that access to relevant data is limited. In a medical setting, existing patient datasets often have distributions that do not necessarily capture the information needed for the accurate diagnosis in a novel problem domain. The resulting diagnosis policies constructed may perform poorly. To address this issue, we propose to efficiently reuse and transfer knowledge from other older domains to minimize as much as possible the diagnosis cost in the new domain. In what follows, we call the diagnosis problem in the new domain the *target* problem and the diagnosis problem in the old domain the *source* problem.

Algorithm Overview

We consider an online setting where data on entities in the target domain are received in sequence, indexed by $\{1, 2, ..., k, ...\}$. Due to the lack of a training dataset in the new domain, it is initially impossible to construct a good policy for the target problem. Instead, we have a set of K source policies Π constructed for K related source problems (e.g. similar diseases or datasets of patients with a similar demography). However, the exact relationship and the effectiveness of these source policies on the target problem are unknown a priori. Our algorithm begins by exploring the source policies for entities in the target domain. After accumulating sufficient data on entities for the target problem, it builds the target policy using the information extracted from applying the source policies. The algorithm is provided next. The parameter $\rho^k \in [0, 1]$ is used to control when to adopt source policies and when to use the newly built target policies; it is decreasing in k and $\lim\limits_{k \to \infty} \rho^k = 0$.

Algorithm 1 Transfer Learning with Multiple Sources

1: **for** each entity k **do**
2: With probability ρ^k
3: Select a source policy to apply
4: With probability $1 - \rho^k$
5: Apply the target policy
6: After the current case is closed
7: Build the target policy using received data
8: **end for**

In Algorithm 1, there are two major questions that remain to be addressed: which source policy to apply (line 3) and how to build the target policy (line 7). Ideally, we would like to always apply the most effective source policy. However, this is impossible since we cannot evaluate the effectiveness of source policies before the system starts as data is not available initially. Thus, we learn the effectiveness of different source policies online. Moreover, since the source policies may lead to a different performance on different entities, we adaptively cluster entities according to the effectiveness of different source policies in order to further improve the diagnosis performance. In a medical setting, the algorithm first learns the best source policy for patients within a larger range of ages, e.g. patients aged between 10 and 80, and then based on the learned results, gradually partitions this age range into smaller intervals. This adaptive clustering of entities has two functions. First, learning is refined for each cluster so that diagnosis performance is improved when the source policy is applied. Second, contexts that are similar to each other can be learned through this clustering and hence, context-specific target policies can be constructed using the data accumulated so far for the target domain.

Source policy selection and adaptive clustering

Let $(k_1, k_2, ..., k_t, ...)$ be the subsequence of received entity cases where a source policy is adopted according to Algorithm 1. Without loss of generality, we normalize the entity context space to be $\Theta \in [0,1]^W$ where W is the context space dimension. We introduce some concepts of the algorithm as follows: 1) **Entity cluster**. An entity cluster is represented by the range of context information that is associated with entities in the cluster. In this paper, we will consider clusters with the form $[i_w 2^{-(l-1)w}, (i_w+1)2^{-(l-1)w}]$ where $i_w \in \{0, 1, ..., 2^{(l-1)w} - 1\}$ for each context dimension $w = 1, ..., W$ for some positive integer l. Such a cluster is called a level-l cluster. At each time k_t when source policies are applied, the algorithm keeps a set of mutually exclusive clusters that cover the entire context space. We call these clusters the *active* clusters, and denote this set by $\mathcal{H}^t$. Clearly, we have $\cup_{C \in \mathcal{H}^t} = \Theta, \forall t$. 2) **Counters**. For each active cluster C, the algorithm maintains several counters: for each source policy $\pi \in \Pi$, $M_C(\pi)$ records the number of entity cases so far in which π is applied. 3) **Diagnosis cost estimates**. For each active cluster C, the algorithm also maintains the sample mean diagnosis cost estimate $\bar{r}_C(\pi)$ for each source policy $\pi \in \Pi$, using the observed diagnosis costs of cases that belong to C so far.

Algorithm 2 Policy Selection and Adaptive Clustering

Initialize $\mathcal{H} = \Theta$, $\bar{r}_\Theta(\pi) = 0, M_\Theta(\pi) = 0, \forall \pi \in \Pi$.
for each entity k_t **do**
Determine active cluster $C \in \mathcal{H}^t$ such that $\theta^t \in C$
Case 1: $\exists \pi \in \Pi$ such that $M_C(\pi) < \gamma(t)$
Randomly select among such policies $\sigma^t = \pi$
Case 2: $\forall \pi \in \Pi$, $M_C(\pi) \geq \gamma(t)$
Select $\sigma^t = \arg\min_{\pi \in \Pi} \bar{r}_C(\pi)$.
Set $M_C(\sigma^t) \leftarrow M_C(\sigma^t) + 1$
(The diagnosis reward r^t is observed.)
Update $\bar{r}_C(\sigma^t)$
Update $\sum_{\pi \in \Pi} M_C(\pi)$ using all past cases.
if $\sum_{\pi \in \Pi} M_C(\pi) \geq \zeta(l)$ **then**
Uniformly partition C into 2^W level-$(l+1)$ clusters.
Update the set of active clusters $\mathcal{H}^t$.
Update the counters and cost estimates for all new clusters using the entity cases received so far.
end if
end for

The algorithm (Algorithm 2) works as follows. When an entity case k_t is received, the algorithm first checks which active cluster $C \in \mathcal{H}^t$ it belongs to. Then it investigates counter $M_C(\pi)$ for all $\pi \in \Pi$ to see if there exists any under-explored source policy π such that $M_C(\pi) \leq \gamma(t, l)$ where $\gamma(t, l)$ is a time- and level-dependent control function. If there exists such an under-explored policy π, then the algorithm selects this policy for the current entity case. This is called an exploration step. If there does not exist any under-explored policy, then the algorithm selects the policy with the lowest cost estimate $\arg\min_{\pi \in \Pi} \bar{r}_C(\pi)$. This is called an exploitation step. After the diagnosis cost of the current entity case is observed, the cost estimate of the selected policy is updated. Moreover, if $\sum_{\pi \in \Pi} M_C(\pi) \geq \zeta(l)$, where $\zeta(l)$ is a level-dependent control function, the current cluster C is partitioned in to 2^W level-$(l+1)$ clusters. From the next entity case on, C is deactivated and the new level-$(l+1)$ clusters are activated. We will show how to select the control functions $\gamma(t, l)$ and $\zeta(l)$ in the next section.

Cluster-specific target policy construction

Entity clusters for which the estimated best source policies are the same are considered to be similar and hence, they are grouped together to form a dataset from which the problem parameters can be estimated. Using these K set of parameters, we can produce K context-specific target policies.

Prior knowledge

1) The current algorithm assumes that, before the system starts, there is no data about entities for the target task. Our algorithm can easily incorporate this information if it becomes available. Specifically, the reward estimates and counters are initialized using the existing entity cases instead

of zeros. 2) The current algorithm also assumes the performance of the source policies are unknown for all entity cases. Therefore, for each entity cluster, all source policies have to be explored to learn their performance. However, if we have prior knowledge about the source policies in different clusters, then only a subset of the source policies needs to be learned. For instance, if we know that the source policy π_1 does not perform well for an entity in cluster C, then there is no need to explore it for all entities in C.

Domain expert overriding

As mentioned, the computer-aided diagnostic system only makes diagnostic recommendations to the domain expert who has the option to override the recommended diagnostic policies when he/she has a high confidence of the best way to execute tests and diagnose based on his/her own expertise. The proposed system allows such overrides and the algorithm can be easily modified to incorporate such changes. Specifically in a medical setting, the counter and reward update steps can be skipped when the doctor decides to override the recommendation and uses a different diagnostic policy.

Theoretical Performance Evaluation

In this section, we evaluate the performance of the policy selection and entity clustering algorithm, which is the core of our transfer learning approach.

Learning regret bound

Let $\pi^*_\theta \in \Pi$ be the best source policy that minimizes the diagnosis cost for entity cases with context θ. Then the expected diagnosis cost for the beginning T entity cases where source policies are adopted is $R^*(T) = \sum_{t=1}^{T} \mathbb{E} r^t(\pi^*_{\theta^t}) = \sum_{t=1}^{T} V_{\theta^t}(\pi^*_{\theta^t})$. By following a learning strategy σ, the expected diagnosis cost is $R^\sigma(T) = \mathbb{E} \sum_{t=1}^{T} r^t(\pi^\sigma_{\theta^t}) = \sum_{t=1}^{T} \mathbb{E} r^t(\pi^\sigma_{\theta^t})$. We use learning regret as the performance metric, which is the marginal diagnosis cost of the learning algorithm with respect to the optimal diagnosis cost, i.e. $\mathrm{Reg}(T) = R^\sigma(T) - R^*(T)$.

To enable rigorous regret analysis, we make the following widely adopted technical assumption below; however, this is not needed for running the algorithm.

Assumption. *(Lipschitz) For each $\pi \in \Pi$, there exists $L > 0, \alpha > 0$ such that for all $\theta, \theta' \in \Theta$, we have $|V_\theta(\pi) - V_{\theta'}(\pi')| \leq L\|\theta, \theta'\|^\alpha$.*

The above assumption states that if the entity context information is similar, then the expected diagnosis cost by selecting the same diagnostic policy is also similar.

The regret bound depends on the entity context arrival process. Let $Y_l(T)$ be the number of level-l clusters that are activated at time T. Let $Z_l(T)$ be the number of level-l active clusters at time T and $l_{\max}(T)$ be the maximum level of clusters for entity k_T.

Theorem. *Let $\gamma(t,l) = 2^{2\alpha l}\ln t$ and $\zeta(l) = A2^{pl}$, then $Reg(T) \leq \sum_{l=1}^{l_{\max}(T)}[Z_l(T)K2^{2\alpha l}\ln T + Y_l(T)(K\sum_{t=1}^{\infty} t^{-2} + A(2LW^{\alpha/2}+2)2^{(p-\alpha)l})]$.*

Proof sketch. We break down the learning regret into three parts $\mathrm{Reg}(T) = \mathrm{Reg}_1(T) + \mathrm{Reg}_2(T) + \mathrm{Reg}_3(T)$ that are respective regrets due to exploration, selection of suboptimal policies in exploitation and selection of near-optimal policies in exploitation. Then we provide a bound for each of these three parts. $\mathrm{Reg}_1(T)$ can be bounded by $\sum_{l=1}^{l_{\max}(T)}[Z_l(T)K2^{2\alpha l}\ln T]$ since the number of exploration steps increases sublinearly in T. $\mathrm{Reg}_2(T)$ can be bounded by $\sum_{l=1}^{l_{\max}(T)} Y_l(T)K\sum_{t=1}^{\infty} t^{-2}$ since the probability of choosing a suboptimal policy in exploitation steps decreases sufficiently rapidly using a Chernoff-Hoeffding bound. $\mathrm{Reg}_3(T)$ can be bounded by $\sum_{l=1}^{l_{\max}(T)} Y_l(T)A(2LW^{\alpha/2}+2)2^{(p-\alpha)l}$ since marginal cost of selecting a near-optimal policy decreases sufficiently rapidly because of our definition of near-optimal policies. □

Corollary. *If the entity context arrivals by time k_T is uniformly distributed over Θ, we have $Reg(T) < T^{\frac{2\alpha+W}{3\alpha+W}}[K(\ln T + \sum_{t=1}^{\infty} t^{-2}) + (LW^{\alpha/2}+2)2^{2\alpha+W}]$*

As we can see, the regret bound is sublinear in T and hence, if T is sufficiently large, then the average regret will be close to 0, which means that the minimal average diagnosis cost is achieved for entity cases in which source policies are applied.

Diagnosis confidence

Initially, the effectiveness of source policies on the target problem is unknown. As more entity cases are received, the effectiveness is gradually learned. We can derive a confidence level of the learned effectiveness of diagnosis policies as follows:

Proposition. *For any active level-l context cluster C, at any time when policy π has been adopted for M_C times on entities which belong to C, then for any individual context $\theta \in C$, the following confidence relation between the estimated diagnosis reward and the true diagnosis reward holds $P(|\bar{r}_C - V_\theta| > L(\sqrt{W}/2^{-l})^\alpha + \epsilon) < e^{-2\epsilon^2 M_C}$.*

Experiments

In this section, we report experimental results using both synthetic data and real-world patient data. Experiments on synthetic data are performed to verify the theoretical results presented above while experiments on real-world data are performed to assess the efficacy of the proposed algorithm.

Synthetic data

Our algorithm works online and adapts to the accumulating target cases over time. We have proven the convergence of its performance to the optimal performance in the last section. Now, we use a synthetic dataset to illustrate its fast convergence to the optimal solution. In this simulation, we use 4 tests with binary results. We create two datasets with different state transition probabilities and cost functions. We

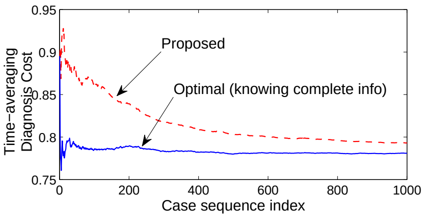

Figure 2: Convergence

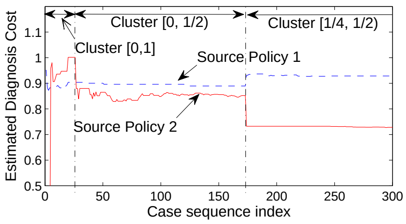

Figure 3: Cost estimates

then create a target dataset in which the cases are generated using these two sets of problem parameters. In particular, cases with context $\theta \in \{[0, 1/4], (1/2, 3/4]\}$ are generated using one set of parameters while cases with context $\theta \in \{(1/4, 1/2], (3/4, 1]\}$ are generated using the other set. Figure 2 shows the time-averaging diagnosis cost achieved by the proposed algorithm. We observe that it converges quickly to the optimal solution that knows the complete distribution and similarity information. Figure 3 illustrates the cost estimates of different source policies during the clustering process. The estimates gradually become more accurate and can provide experts with higher confidence in the different policies' performance.

Real-world patient dataset

We test our proposed algorithm using an alarm data set obtained from the Columbia Medical Center neurological Intensive Care Unit (ICU). This dataset contains over a million alarm events produced by patient monitoring systems for 581 patients.

Methodology

We artificially treat the patient alarm dataset on each day as a separate dataset. K such datasets are picked as the source datasets and another one is picked as the target dataset. Moreover, the target data is made available to the system in sequence. We treat each alarm as a medical test. Hence, only when the test is performed, the corresponding alarm status is revealed. In the experiments, we focus on predicting whether the patient will have at least one of the two secondary complications: Pneumonia and Respiratory failure. The set of tests (alarms) that we consider includes Bradycardia, Tachycardia, High Blood Pressure, Low Blood Pressure, High Respiratory Rate. We assign different costs to differen types of prediction errors. Specifically, we normalized the cost of a miss detection to be 1 and the cost of a false alarm to be c_1. A uniform cost c_2 is assigned for the execution of any test. In the experiments, we use a single patient context: the APACHE II ("Acute Physiology and Chronic Health Evaluation II") score that evaluates the severity of illness of our patients upon admission in the ICU.

Baseline approaches

1) **Empirical Diagnosis (EM)**: All medical tests are executed and all alarms are revealed. It predicts the complication if any of the alarms are positive. 2) **Average Transfer (AT)**: In this approach, we combine all source datasets to estimate the average problem parameters and construct a new diagnostic policy. This average source policy is applied to the target cases while the target policy is gradually learnt. 3) **MultiSourceTriAdaBoost (MSTAB)**: This is a modified version of the state-of-the-art MultiSourceTriAdaBoost algorithm in (Yao and Doretto 2010) for transfer learning with multiple sources. We modified the weight update structure to incorporate the diagnosis cost instead of a plain diagnosis accuracy. Since the original algorithm is an offline algorithm that assumes a training set for the target problem, we also extended it to produce an online version using batch updates as more target cases are received.

Results

We report the diagnosis cost results for three sets of experiments in Table 2 for various parameters. The diagnosis cost is computed by averaging the diagnosis cost of patient cases from the target patient case set. In all experiments, the proposed transfer learning algorithm significantly outperforms the baseline approaches by reducing the diagnosis cost up to 20% against the best baseline approach. We also investigate the impact of different choices of contexts on the diagnosis performance. Table 3 shows the achieved diagnosis cost by using different contexts for $K = 2, c_1 = 0.3$ and $c_2 = 0.005$. The experiment results indicate that the ApacheII score is the best context in our problem. Nevertheless, the diagnosis cost by using any context is lower than those achieved by the baseline approaches.

	c_2	EM	AT	MSTAB	**Proposed**
K = 2, c_1 = 0.3	0.01	0.299	0.259	0.246	**0.212**
	0.005	0.268	0.241	0.231	**0.193**
	0.001	0.245	0.233	0.224	**0.187**
K = 2, c_1 = 0.5	0.01	0.357	0.309	0.312	**0.292**
	0.005	0.332	0.290	0.282	**0.267**
	0.001	0.302	0.277	0.262	**0.260**
K = 3, c_1 = 0.3	0.01	0.274	0.260	0.265	**0.218**
	0.005	0.258	0.239	0.232	**0.191**
	0.001	0.250	0.226	0.224	**0.192**

Table 2: Diagnosis cost comparison

Context	Age	GCS	ApacheII	ApachePhys
Diag. cost	0.222	0.225	0.193	0.205

Table 3: Impact of contexts

Conclusion

In this paper, we proposed an online transfer learning approach for differential diagnosis determination and showed how it can be incorporated as part of a clinical decision support system to improve the diagnosis performance. We envision that the proposed methodology can also be used in other complex diagnosis systems besides clinical diagnosis, such as cyber-security, biological and mechanical diagnosis.

References

Bonet, B., and Geffner, H. 2012. Action selection for mdps: Anytime ao* versus uct. In *AAAI*.

Chow, S.-C.; Chang, M.; et al. 2008. Adaptive design methods in clinical trials-a review. *Orphanet J Rare Dis* 3(11).

Duan, L.; Tsang, I. W.; Xu, D.; and Chua, T.-S. 2009. Domain adaptation from multiple sources via auxiliary classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 289–296. ACM.

Fernández, F., and Veloso, M. 2006. Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 720–727. ACM.

Gao, J.; Fan, W.; Jiang, J.; and Han, J. 2008. Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 283–291. ACM.

Ge, L.; Gao, J.; Ngo, H.; Li, K.; and Zhang, A. 2014. On handling negative transfer and imbalanced distributions in multiple source transfer learning. *Statistical Analysis and Data Mining: The ASA Data Science Journal*.

Greiner, R.; Grove, A. J.; and Roth, D. 2002. Learning cost-sensitive active classifiers. *Artificial Intelligence* 139(2):137–174.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670. ACM.

Ling, C. X.; Yang, Q.; Wang, J.; and Zhang, S. 2004. Decision trees with minimal costs. In *Proceedings of the twenty-first international conference on Machine learning*, 69. ACM.

Luo, P.; Zhuang, F.; Xiong, H.; Xiong, Y.; and He, Q. 2008. Transfer learning from multiple source domains via consensus regularization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, 103–112. ACM.

Madden, M. G., and Howley, T. 2004. Transfer of experience between reinforcement learning environments with progressive difficulty. *Artificial Intelligence Review* 21(3-4):375–398.

Marx, Z.; Rosenstein, M. T.; Kaelbling, L. P.; and Dietterich, T. G. 2005. Transfer learning with an ensemble of background tasks. *Inductive Transfer* 10.

Norton, S. W. 1989. Generating better decision trees. In *IJCAI*, volume 89, 800–805.

Núñez, M. 1991. The use of background knowledge in decision tree induction. *Machine learning* 6(3):231–250.

Pearl, J., and Bareinboim, E. 2011. Transportability of causal and statistical relations: A formal approach. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, 540–547. IEEE.

Raina, R.; Ng, A. Y.; and Koller, D. 2006. Constructing informative priors using transfer learning. In *Proceedings of the 23rd international conference on Machine learning*, 713–720. ACM.

Singleton, K.; Speier, W.; Bui, A.; and Hsu, W. 2014. Motivating the additional use of external validity: Examining transportability in a model of glioblastoma multiforme. In *Proceedings of AMIA*.

Slivkins, A. 2009. Contextual bandits with similarity information. *arXiv preprint arXiv:0907.3986*.

Sohrabi, S.; Baier, J.; and McIlraith, S. 2010. Diagnosis as planning revisited. In *Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning (KR-10)*, 26–36.

Tan, M. 1993. Cost-sensitive learning of classification knowledge and its applications in robotics. *Machine Learning* 13(1):7–33.

Tekin, C.; Zhang, S.; and van der Schaar, M. 2014. Distributed online learning in social recommender systems. *IEEE Journal of Selected Topics in Signal Processing* 8(4):638–652.

Torrey, L.; Shavlik, J.; Natarajan, S.; Kuppili, P.; and Walker, T. 2008a. Transfer in reinforcement learning via markov logic networks. In *AAAI Workshop on Transfer Learning for Complex Tasks*.

Torrey, L.; Shavlik, J.; Walker, T.; and Maclin, R. 2008b. Relational macros for transfer in reinforcement learning. In *Inductive Logic Programming*. Springer. 254–268.

Turney, P. 2000. Types of cost in inductive concept learning. In *Workshop on Cost-Sensitive Learning at 17th International Conference on Machine learning*.

Yao, Y., and Doretto, G. 2010. Boosting for transfer learning with multiple sources. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 1855–1862. IEEE.

Zubek, V. B.; Dietterich, T. G.; et al. 2004. Pruning improves heuristic search for cost-sensitive learning. Technical report, Corvallis, OR: Oregon State University, Dept. of Computer Science.