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 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 on real-world patient alarm datasets collected from Neurological Intensive Care Units (with reduced costs by 20%).

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. 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.

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 meta-data 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. A first strand of related research focuses on cost-sensitive learning (Turney 2000)(Greiner, Grove, and Roth 2002)(Zubek, Dietterich, and others 2004). A disadvantage of these approaches is that they rely on training datasets to learn the appropriate model. Our focus in this paper is on how to overcome the lack of initial training data by using transfer of knowledge from relevant datasets. The majority of the transfer learning literature assumes a single source from which knowledge can be transferred (Marx et al. 2005). Transfer learning from multiple sources is much more challenging; most works aiming to address this problem focus on classification problems (Duan et al. 2009)(Yao and Doretto 2010). In our considered setting, the target data arrives sequentially and the features are not given but need to be discovered (by performing various tests). 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 (Fernández and Veloso 2006). However, such works only consider the availability of a single source, while our work focuses on multiple sources. Our solution builds on the contextual bandit framework (Slinkins 2009). 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.

**Computer-aided Diagnosis System**

We use \( \theta \in \Theta \) to denote the initial exogenous context 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.). Let \( Q = \{1, 2, \ldots, N\} \) denote the set of possible tests, \( N < \infty \). We assume that each test \( q \in Q \) has a finite set of possible results, denoted by \( O_q \). We also define an “unknown” test result to be assigned to tests that have not been performed.

At any point in time, an entity is assigned with a state \( s \) that represents the known test results that have been performed. 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. Let \( S \) denote the state space. The initial state of an entity is \( s_0 = \text{unknown} \), \( \forall q \in Q \). Depending on the current entity state \( s \), the computer-aided diagnostic system either recommends new tests to be performed to extract more knowledge or recommends a diagnosis decision if it has enough information about the entity to perform. Let the action space be \( A = \{Q, D\} \) where \( D \) represents the diagnosis space; they are kept fixed. We assume that if the expert follows an action \( a \in D \), 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 \( \{q_1^k, \ldots, q_n^k\} \) 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:

\[
    c^k = \sum_{i \in \{1, \ldots, n^k\}} c^k(q_i^k) + \beta d^k
\]

where \( c^k(q_i^k), \forall i = 1, \ldots, n^k \) 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. 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 \( \pi = \{a(s)\}_{s \in S} \). Hence, given a diagnostic policy, after observing the entity state, the diagnostic system can recommend an action to the domain expert. Our goal is to develop diagnostic policies that minimize the diagnosis cost.

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 \( p(s'|s, q) \) denote the transition probability from state \( s \) to \( s' \) when test \( q \) is executed. Since taking an action \( d \in D \) always leads to a diagnosis and closes the current case, we have \( p(\emptyset|s, d) = 1, \forall d \in D \). Let \( c(q|s) \) denote the expected cost of performing test \( q \) on entities in state \( s \). Let \( c(d|s) \) denote the expected cost of making a diagnose \( d \) on entities in state \( s \). We unify these two types of costs in a cost function \( c : A \times S \rightarrow \mathbb{R} \) 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. The optimal diagnostic policy that minimizes the expected diagnostic cost in each state is defined using the Bellman equation:

\[
    J(a|s) = c(a|s) + \sum_{s' \in S} p(s'|s, a)V(s')
\]

where \( V(s') = \min_{a'} J(a'|s') \). Thus \( \pi_{opt} = \{a_{opt}(s)\}_{s \in S} \) such that \( \forall s, a_{opt}(s) = \arg \min_{a'} J(a|s) \). Since the entity comes with the initial state \( s^\text{init} \), the expected diagnostic reward is \( V(s^\text{init}) \). With abuse of notation, we let \( V(\pi) = V(s^\text{init} | \pi) \) denote the diagnostic cost by using \( \pi \). If the prob-

![Diagram of Computer-aided diagnosis system](image)
lem parameters were known, then the optimal diagnostic policy making problem can be solved by backward induction using the estimated problem parameters from an existing dataset. 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).

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 constructed in 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

We consider an online setting where data on entities in the target domain are received in sequence, indexed by \( \{1, 2, \ldots, k, \ldots\} \). 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 \( \Pi \) 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 known 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 in Algorithm 1. 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 \( \lim_{k \to \infty} \rho^k = 0 \).

Algorithm 1 Transfer Learning with Multiple Sources

1: for each entity \( k \) do
2: \quad With probability \( \rho^k \), select a source policy to apply
3: \quad With probability \( 1 - \rho^k \), apply the target policy
4: \quad After the current case is closed
5: \quad Build the target policy using received data
6: 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). We discuss them next. Let \( (k_1, k_2, \ldots, k_t, \ldots) \) 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 \( [\frac{i_w}{w}, \frac{(i_w + 1)2^{-(l-1)w} - 1}{w}] \) for each context dimension \( w = 1, \ldots, W \) for some positive integer \( l \). Such a cluster is called a level-\( l \) cluster. At each time \( k \), 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 \( \mathcal{H}^l \). Clearly, we have \( \bigcup_{C \in \mathcal{H}^l} = \Theta, \forall l \).

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 \( \pi \) is applied. 3) Diagnosis costs. 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

1: Initialize \( \mathcal{H} = \Theta, \bar{r}_C(\pi) = 0, M_C(\pi) = 0, \forall \pi \in \Pi \).
2: for each entity \( k \) do
3: \quad Determine active cluster \( C \in \mathcal{H}^l \) such that \( \theta^l \in C \)
4: \quad Case 1: \( \exists \pi \in \Pi \) such that \( M_C(\pi) < \gamma(t) \)
5: \quad \quad Randomly select among such policies \( \sigma^l = \pi \)
6: \quad Case 2: \( \forall \pi \in \Pi \), \( M_C(\pi) \geq \gamma(t) \)
7: \quad \quad Select \( \sigma^l = \arg \min_{\pi \in \Pi} r_C(\pi) \).
8: \quad Set \( M_C(\sigma^l) \leftarrow M_C(\sigma^l) + 1 \)
9: \quad \quad (The diagnosis reward \( r^l \) is observed.)
10: \quad Update \( \bar{r}_C(\sigma^l) \)
11: \quad Update \( \sum_{\pi \in \Pi} M_C(\pi) \) using all past cases.
12: \quad if \( \sum_{\pi \in \Pi} M_C(\pi) \geq \zeta(l) \) then
13: \quad \quad Uniformly partition \( C \) into \( 2^W \) level-\( l + 1 \) clusters.
14: \quad \quad Update the set of active clusters \( \mathcal{H}^l \).
15: \quad \quad Update the counters and cost estimates for all new clusters using the entity cases received so far.
16: \quad end if
17: end if
18: end for

The algorithm is described in Algorithm 2. When an entity case \( k \) is received, the algorithm first checks which active cluster \( C \in \mathcal{H}^l \) 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 \( \pi \) 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, 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 pol-
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 \( \gamma(l,1) \) and \( \zeta(l) \) in the next section.

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.

Confidence Bound

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. 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 \( \pi \) 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

\[
P(|\bar{r}_C - V_\theta| > L(\sqrt{W}/2^{-l})^\alpha + \epsilon) < e^{-2e^2M_C}.
\]

**Experiments**

**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 different 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 1 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 2 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 ApachII 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.

<table>
<thead>
<tr>
<th>( K = 2, c_1 = 0.3 )</th>
<th>( c_2 )</th>
<th>EM</th>
<th>AT</th>
<th>MSTAB</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.259</td>
<td>0.259</td>
<td>0.259</td>
<td>0.246</td>
<td>0.212</td>
</tr>
<tr>
<td>0.005</td>
<td>0.268</td>
<td>0.241</td>
<td>0.241</td>
<td>0.231</td>
<td>0.193</td>
</tr>
<tr>
<td>0.001</td>
<td>0.245</td>
<td>0.233</td>
<td>0.233</td>
<td>0.224</td>
<td>0.187</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( K = 2, c_1 = 0.5 )</th>
<th>( c_2 )</th>
<th>EM</th>
<th>AT</th>
<th>MSTAB</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.357</td>
<td>0.309</td>
<td>0.309</td>
<td>0.312</td>
<td>0.292</td>
</tr>
<tr>
<td>0.005</td>
<td>0.332</td>
<td>0.290</td>
<td>0.290</td>
<td>0.282</td>
<td>0.267</td>
</tr>
<tr>
<td>0.001</td>
<td>0.302</td>
<td>0.277</td>
<td>0.277</td>
<td>0.262</td>
<td>0.260</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( K = 3, c_1 = 0.3 )</th>
<th>( c_2 )</th>
<th>EM</th>
<th>AT</th>
<th>MSTAB</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.274</td>
<td>0.260</td>
<td>0.260</td>
<td>0.265</td>
<td>0.218</td>
</tr>
<tr>
<td>0.005</td>
<td>0.258</td>
<td>0.239</td>
<td>0.239</td>
<td>0.232</td>
<td>0.191</td>
</tr>
<tr>
<td>0.001</td>
<td>0.250</td>
<td>0.226</td>
<td>0.226</td>
<td>0.224</td>
<td>0.192</td>
</tr>
</tbody>
</table>

**Table 1: Diagnosis cost comparison**

<table>
<thead>
<tr>
<th>Context</th>
<th>Age</th>
<th>GCS</th>
<th>ApacheII</th>
<th>ApachePhys</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diag. cost</td>
<td>0.222</td>
<td>0.225</td>
<td>0.193</td>
<td>0.205</td>
</tr>
</tbody>
</table>

**Table 2: 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


