Using Anchors to Estimate Clinical State without Labeled Data

Yoni Halpern[1], Youngduck Choi[1], Steven Horng MD MMSc[2], David Sontag PhD[1]
[1]New York University, New York, NY
[2]Beth Israel Deaconess Medical Center, Boston, MA

Abstract

We present a novel framework for learning to estimate and predict clinical state variables without labeled data. The resulting models can used for electronic phenotyping, triggering clinical decision support, and cohort selection. The framework relies on key observations which we characterize and term "anchor variables". By specifying anchor variables, an expert encodes a certain amount of domain knowledge about the problem while the rest of learning proceeds in an unsupervised manner. The ability to build anchors upon standardized ontologies and the framework's ability to learn from unlabeled data promote generalizability across institutions. We additionally develop a user interface to enable experts to choose anchor variables in an informed manner. The framework is applied to electronic medical record-based phenotyping to enable real-time decision support in the emergency department. We validate the learned models using a prospectively gathered set of gold-standard responses from emergency physicians for nine clinically relevant variables.

1 Introduction

Health information technology is an essential part of modern health care, providing health care professionals with critical information about patients and allowing them to make maximally informed decisions about a patient's care.

In order to accelerate the development of advanced clinical decision support tools, we seek to create a new middleware application layer consisting of hundreds of clinical state variables that summarize a patient's past and current state. These clinical state variables collectively form a patient phenotype that is continuously estimated throughout a patient's stay and can be used by decision support applications to better inform, guide, and expedite the workflows of clinicians. We define clinical decision support for this paper very broadly to include any functionality that helps a clinician to be better, whether this means more complete or efficient documentation, adherence to clinical guidelines, disease specific order sets, alerts and reminders, visualizations that summarize patient information, or contextual information retrieval, extraction, and summarization.

For example, nursing home patients have different clinical care needs and treatments than other patients in the emergency department. They are more likely to have resistant organisms to standard antibiotic therapy, and therefore must be empirically treated with broad spectrum antibiotics. They are also more likely to fall and therefore require special handling to minimize fall risk. Decision support tools can be used to remind clinicians to order broad spectrum antibiotics and warn them when they do not. They can also be used to alert ancillary staff such as patient transporters and radiology technicians to take appropriate fall precautions, a precaution that may not always be obvious. Although whether a patient is from a nursing home could be collected manually as structured data, there are hundreds of clinical state variables that would be valuable for decision support. Collecting all of these variables for all patients is not feasible. Previous systems that use this type of approach often fail to get the support of clinical users and are systematically not used.

Standard ontologies and knowledge representations are necessary but not sufficient to catalyze the development of advanced decision support and enable transferability of models across institutions. In particular, the patient's state may not be directly observable. Machine learning allows us to reason about patients in these settings. Previous approaches such as logistic regression, support vector machines, decision trees, and neural networks require domain experts to label a fairly large number of positive and negative examples which is time consuming (e.g. [1, 2, 3]). Even after this labeling work has been done, these learned classifiers often do not generalize well across institutions since the learned classifiers are highly dependent on the representation they are trained on. Retraining for each site can require repeating the labeling process, modifying the representation and adjusting rules (e.g. [4, 5]). A review of recent approaches to automated patient phenotyping from electronic records can be found in [6].

In this paper, we describe a methodology for learning to estimate a patient phenotype, consisting of hundreds of different clinical state variables, based on information available in the Electronic Medical Record (EMR). We

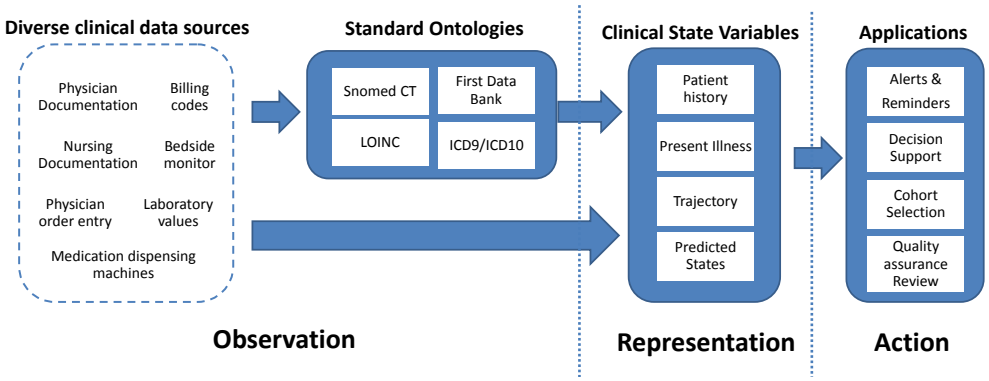

Figure 1: Schematic diagram of a middleware layer, reading inputs from the EMR and acting as an intermediate representation for applications. To improve generalization across institutions, we make no assumptions about the raw clinical data sources. Rather, we make use of existing standardized ontologies to specify anchors for each clinical state variable, and then use the anchors within each institution to learn classifiers to estimate the variables using previously collected, unlabeled, raw clinical data.

use a combination of domain expertise and vast amounts of unlabeled data, rather than relying solely on domain expertise or requiring labor-intensive manual labeling. To learn the models, we require only that certain highly informative variables that we call *anchor variables* be identified by an expert, and the rest is learned from large amounts of unlabeled data in an unsupervised manner. To promote transferability between institutions, we assume only that these anchor variables remain stable between institutions while the rest of the underlying observation model can change. Figure 1 presents a schematic view of such a system.

The main contributions of this paper are as follows:

1. We introduce the concept of anchor variables.
2. We show how to use anchors within an unsupervised machine learning algorithm to estimate each clinical state variable without any human labeling.
3. We describe a novel user interface developed to help with choosing a good set of anchors for each clinical state variable and for performing interactive cohort selection.
4. We evaluate our algorithm's performance using a prospectively gathered set of gold-standard responses from emergency physicians on nine different clinically relevant patient phenotyping tasks.

More broadly, this paper presents a novel approach to harnessing unstructured and structured data found in electronic medical records to estimate clinical state variables that can be used in a wide variety of settings, including both retrospective and prospective clinical care, research, administration, and quality improvement.

2 Anchor variable framework

2.1 Observed and latent variables in the EMR

We formalize anchor variables within the context of latent variable models. In our framework, variables are categorized as either *latent* or *observed*. In a model with m latent variables and n observed variables, each patient is described with a collection of latent variables $\mathcal{Y} = \{Y_1, Y_2, ...Y_m\}$ and a collection of observations $\mathcal{X} = \{X_1, X_2, .., X_n\}$. Here we focus on binary variables but the work can be readily extended to consider categorical variables as well.

Observed variables represent quantities that can be observed directly from the EMR. These include structured variables like age and sex, but also queries that can be computed on semi-structured or free-text data, such as *Does any of the free text contain the phrase "chest pain"?* or *Does the medication list include "GSN_004380" (aspirin)?* Here we assume a very general form of the EMR without making assumptions of which fields are present and what structure they take. These observations are represented in Figure 1 as the leftmost section of the schematic.

Observations from the EMR are inherently noisy and may miss important information. For example, if the note contains the short-form "cp" instead of chest pain, the chest pain observation would be negative even though

the patient may actually have the symptom. Our model addresses this by explicitly treating observations as noisy evidence about the clinical state variables of interest. Additionally, as we will see later on in more detail, we allow certain key observations to be trusted as true when they are *positive* but not when they are negative.

Latent variables represent quantities that **cannot** be observed directly from the EMR or computed as a simple query, but for which the EMR can provide information that would be useful to answer the question. These variables could be the answers to higher level questions such as *Does the patient have* $\{$*an infection, altered mental status, a history of alcoholism*$\}$*?* The answers to these higher level clinical state questions form a useful representation of the patient, shown in the middle section of Figure 1. In many cases, the answers can be found in the EMR, though they are usually found in free-text sections of the record. Formulating queries to extract this information directly from the free-text is notoriously difficult due to the vast number of ways each fact can be expressed [7]. In other cases, the answers may simply not be documented at all, either because they are not known or due to a failure to document. In all of these cases, even though it is impossible to observe the answers directly, it may still be possible to infer them based on all of the other data observable in the record. In our framework, the goal will be to learn a classifier to predict the value of the latent variables $\mathcal{Y}$ with only access to the observations $\mathcal{X}$.

2.2 Anchor variables

It is important to recognize that many different EMR systems exist and the set of observable variables in one system may not map directly to the set of observable variables in another [8]. Promulgation of standards in the form of ontologies to normalize the representation of observations as well as knowledge representation standards such as the Continuity of Care Document will help to reduce some of this effect. However, different vendors will continue to innovate beyond these standardizations and there will always be a structural differences in data representations. In our framework we assume that the set of observable variables can change from system to system as long as a few key informative observations are conserved. These observations are *anchor* observations with the following property:

Definition 1. *An observation X_j is an* anchor *for a latent variable Y_i if X_j is conditionally independent of all other observations, X_k $\forall k \neq j$, conditioned on the value of Y_i.*

In other words, anchor observations provide a direct, albeit noisy, view of the underlying latent variable we wish to predict. The key characteristic of an anchor is the conditional independence property which states once the value of the latent variable is known, no other observables provide additional information about the anchor variable. While anchors tend to provide strong evidence towards the value of the latent variable, it would not make sense to answer the questions simply based on the values of the anchors alone. However, our insight is that we can explicitly treat anchors as noisy labels and use them within a learning algorithm. Learning with noisy labels is a subject that has been studied extensively in machine learning literature (e.g. [9, 10]) and we leverage that work here. By using domain knowledge to identify these noisy labels in the data itself, we no longer require manual labeling of the data before it can be fed to a machine learning algorithm. The resulting method is extremely portable. As long as anchors can be shared between institutions, no new labeling work is required to train the classifiers for a new institution's data.

In the next section we describe an unsupervised method to learn decision rules to predict the values of latent variables, $\mathcal{Y}$, using all of the the observed variables, $\mathcal{X}$, when anchors for each latent variable have been specified. When transferring to a new system with a potentially different set of observed variables we can perform the same unsupervised training on the data available in the new system with no additional human input as long as the anchors are preserved or can be mapped into the new data system. In order to ensure that anchors are not particular to an institution, we can use standard ontologies when specifying the anchors.

2.3 Learning decision rules with positive anchors

Let $A \in \mathcal{X}$ be an anchor variable for Y_i. The collection $\tilde{\mathcal{X}} = \mathcal{X} \setminus A$ represents the observations $\mathcal{X}$ with the anchor removed. We describe a procedure for learning a single decision rule, trying to predict the value of a single latent variable Y_i from observed values X using a special type of anchor that we call *positive* anchors.

Definition 2. *An observation A is a* positive anchor *for a latent variable Y_i if it is an anchor for Y_i and $P(Y_i = 1|A = 1) = 1$.*

Intuitively, observing the anchor to be positive unambiguously reveals the state of the latent variable to be positive, while observing it to be negative does not reveal the state of the latent variable. This setting has previously been studied under the name *positive-only labels* [9] and a procedure for learning with positive-only labels is as follows (see [9] for a detailed derivation):

1. Learn a calibrated classifier (e.g. logistic regression) to predict $P(A = 1|\tilde{\mathcal{X}})$.
2. Using a validate set, compute $C = \frac{1}{|\mathcal{P}|} \sum_{k \in \mathcal{P}} P(A = 1|\tilde{\mathcal{X}}^{(k)})$ where $\mathcal{P}$ is the set of data in the validate set where $A = 1$.
3. For a previously unseen patient t, predict $\begin{cases} P(A = 1|\tilde{\mathcal{X}}^{(t)})/C & \text{if } A^{(t)} = 0 \\ 1 & \text{if } A^{(t)} = 1 \end{cases}$

Positive anchors have some appealing properties. First, the algorithm to learn with them is extremely simple and requires only the capability of learning logistic regression models, which are standard in most statistical packages. Second, positive anchors have an intuitive interpretation for the human providing the anchors. For positive anchors, the anchor should be a quantity that *can only be caused* by the latent variable being "on". Finally, the positivity of an anchor is easy to verify. If an expert is presented with examples of patients with a positive anchor, they can confirm that in fact all (or almost all) of the presented patients are positive for the latent variable of interest. Conditional independence is more difficult to verify and requires that an expert verify that the proposed anchor is truly completely explained by the clinical state of interest.

Anchors like this exist in medicine. For example, a positive rapid antigen test for Group A streptococci is very specific for strep throat and can serve as a positive anchor. The absence of a positive test result can be uninformative either due to the low sensitivity of the test, or due to the possibility that the test was never performed because the diagnosis was obvious and the patient was treated without testing [11]. In addition to carefully choosing anchors, data preprocessing can be performed to increase the amount of conditional independence between anchors and other observations. For example, in the Methods section we describe how common bigrams are represented in order to avoid obvious violations of conditional independence. In real use cases, no anchors will ever perfectly meet the criteria in Definitions 1 and 2, and missing data will not be completely at random. Nonetheless, approximate anchors can perform well on real data, as we demonstrate with experimental results for a range of clinical variables. The above definitions are still useful as they give theoretical principles by which to choose good anchors.

If multiple anchors are specified, they can be combined in a number of different ways. The simplest way is to create a composite anchor out of the union of all of the individual anchors. For example, if the diagnosis code ICD9-288.00 (neutropenia, unspecified) and the word "immunocompromised" are both anchors for the latent variable `isImmunosuppressed`, then we can create a single composite anchor which is present if *either* of these two observations is present. If the original anchors are positive as in Definition 2, then the new composite anchor is also a positive anchor. Using the composite anchor is advantageous compared to choosing a single anchor because it occurs more frequently, providing more positive examples for training.

3 Specifying anchors with an interactive display

In practice, specifying anchors can be challenging. In order to specify anchors, one must have sufficient domain knowledge to evaluate whether a variable fits the definition of an anchor. To ease the process of eliciting anchors from domain experts, we built an interactive interface to allow domain experts to specify anchor variables and visualize the resulting model learned with those anchors. Figure 2 shows a screenshot of the tool being used to specify anchors to identify HIV positive patients.

The interface is a general tool for specifying anchors and viewing the learned classifier. A user can add latent variables and specify anchors for them. After adding an anchor, the user can, in real time, update the learned model and view a ranked list of patients at the bottom of the screen. The ranking is generated according to the predicted likelihood of the latent variable being positive according to the model built with the current set of anchors. For each patient, a short summary is presented for easy viewing, and selected patients can be viewed in more detail in the middle pane. Patients can be filtered according to three different criteria: view only patients with anchors (to judge whether the anchors are catching the correct subset of patients), view patients that have the most recently added anchor (to judge the incremental effect) and view patients without anchors (looking at a ranked list of these patients provides an idea of how well the learning algorithm has *generalized* beyond simply looking for patients that have the anchors).

After learning a model, the tool additionally *suggests* new anchors by showing the observations ranked by weights of a linear classifier learned with a penalty on the L1 norm of the weight vector. The L1 penalty encourages the learned classifier to use a minimal number of variables, effectively selecting highly informative observations. The user then uses clinical judgment to decide whether or not each suggestion would make a good anchor, e.g. by including the new observation and seeing the incremental effect on the ranking, or by viewing the newly anchored patients. The result is a simplified active learning workflow with a human-in-the-loop. In this work we focus on

the basic task of learning classifiers with anchors, leaving more advanced active learning techniques like asking questions about specific patients in a maximally informative manner [12] and providing detailed performance feedback for the user to future work.

The ranking and filtering mechanisms provide feedback to the user, giving information about whether the model is being built reasonably or not. Figure 3 in Section 5 shows a user trace of a clinician using our interface to specify anchors to identify patients with a cardiac etiology. The feedback coming from patient rankings, viewing recently anchored patients and looking at suggested anchors is sufficient to allow him to incrementally build better models by specifying new anchors.

Anchors can be specified as words or phrases, and they are interpreted as queries on the free text portions of the medical record. Additionally, the interface allows for incorporation of anchors according to standardized hierarchical ontologies. For example, medications are grouped by families according to First Databank's Enhanced Therapeutic Classification (ETC) hierarchy, and diagnosis codes are grouped according to the ICD9 hierarchy. For these hierarchical structures, including a parent as an anchor automatically adds all of its children as well.

In addition to specifying anchors, the tool is useful for performing fast interactive cohort selection, allowing the user to quickly learn classifiers to find members of a target population using the anchor approach. The learned classifiers can be exported as well for use in real-time decision applications. The tool is freely available for download at `http://sontaglab.cs.nyu.edu/`.

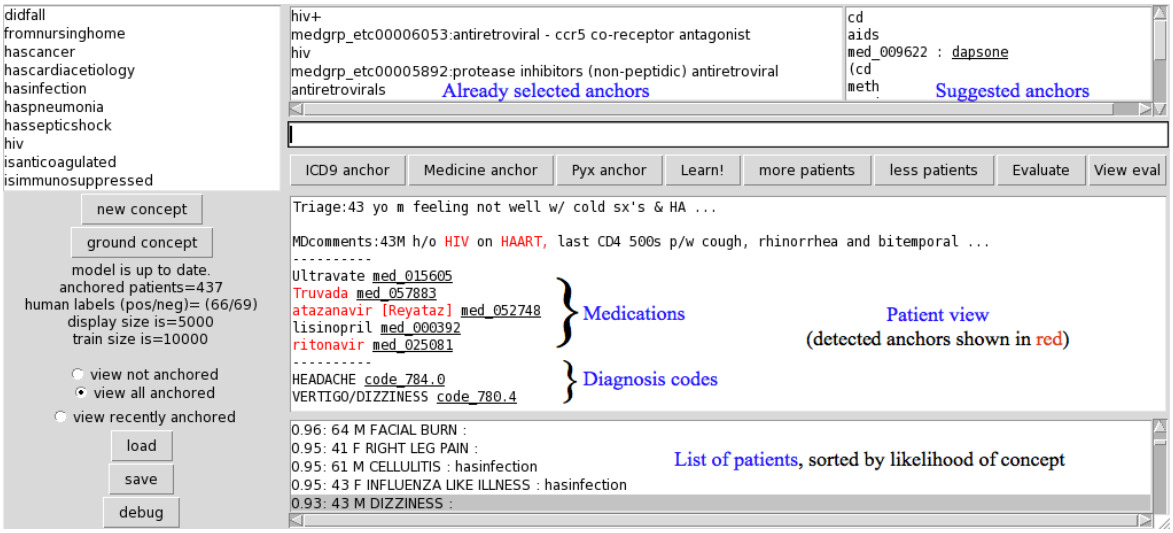

Figure 2: A screenshot of the anchor elicitation tool using deidentified patient information. Conditional highlighting emphasizes the presence of previously specified anchors in a note so that the physician can determine quickly whether its usage is as expected.

4 Methods

4.1 Data description

For training and evaluation we use a collection of 273,174 emergency department patient records collected from a 55,000 patient/year Level 1 trauma center and tertiary academic teaching hospital between 2008 and 2013. Each record represents a single patient visit. All consecutive ED patient visits were included in the data set. No visits were excluded. This study was approved by our institutional review board.

In order to evaluate the utility of the latent variable predictors learned, we collected gold standard labels from the primary clinical provider caring for a patient at the time of disposition from the emergency department, (admission, discharge, or transfer). As part of the routine clinical workflow of disposition, clinicians were asked a series of 2-3 questions chosen randomly from a rotating pool of questions. The answers serve as gold standard labels. These questions were entirely voluntary, but received an approximately 85% completion rate. As part of our quality assurance process, we routinely review random samples of cases to confirm the quality of data obtained using this technique and confirm that missing data occurs completely at random. Since only 2-3 questions were asked at any given time, it took approximately 2 years to collect the labels for this paper, in addition to labels for other unrelated research projects not presented here. Table 1 shows the clinical state variables and the associated disposition questions that we use for evaluation in this paper. Responses were collected for all consecutive patients

except for the `didFall` question which was only asked for patients who had a head CT scan. Clinicians reported responses using a five point scale with 1 being most negative and 5 being most positive. The `didFall` question used a three-point scale of "No", "Uncertain" or "Yes". When converting the labels to binary values we take 4 or above as positive. Responses labeled as "Uncertain" are treated as unlabeled.

Latent Variable	Disposition Question	Additional Information	Labels Collected	Fraction Positive
didFall	Did the patient fall from standing or lesser height?	None	9,831	0.118
hasCardiacEtiology	In the workup of this patient, was a cardiac etiology suspected?	None	17,258	0.068
hasInfection	Do you think this patient has an infection?	Suspected or proven viral, fungal, protozoal, or bacterial infection	62,589	0.213
fromNursingHome	Is the patient from a nursing home or similar facility?	Interpret as if you would be giving broad spectrum antibiotics	36,256	0.045
hasCancer	Does the patient have an active malignancy?	Malignancy not in remission; and recent enough to change clinical thinking	4,091	0.042
hasPneumonia	Do you think the patient has pneumonia?	None	9,934	0.073
isAnticoagulated	Prior to this visit, was the patient on anticoagulation?	Excluding antiplatelet agents like aspirin or plavix	1,082	0.047
isImmunosuppressed	Is the patient currently immunocompromised?	None	12,857	0.040
hasSepticShock	Is the patient in septic shock?	None	6,867	0.020

Table 1: Questions asked of physicians at disposition time to obtain a gold standard set of labels. Additional information was displayed in a clickthrough screen with a link from the main question page.

4.2 Representation and preprocessing

Patient records are represented as containing six distinct types of observable variables which come from semi-structured sections of the EMR: 1. ICD9 diagnosis codes (from billing information), 2. current medications recorded during medication reconciliation, 3. medications dispensed during the ED course as reported by medication dispensing machines (Pyxis), 4. free text sections formed by a concatenation of chief complaint, triage assessment and physician's comments, 5. Age, 6. Sex.

Deidentified free text was preprocessed using a modified version of NegEx [13, 14] and negated words were replaced by a new token (i.e. if the token "fever" was within the scope of a negation, it was transformed to a new token, "negfever"). A second step of preprocessing collected 1,500 significant bigrams and appended them to the text (i.e. the phrase "chest pain" was augmented to be "chest pain chest-pain" with an extra token representing the bigram). When learning with anchors, we remove the component words (i.e. "chest pain" is replaced by a single token "chest-pain"). We do this in order to increase the amount of conditional independence between anchors which are bigrams and the rest of the text. If the token "chest-pain" is chosen as an anchor, it will not be conditionally independent of the tokens "chest" and "pain" without the removal step. For training linear classifiers, the first representation is strictly more general, so it should not hurt the performance of our baseline algorithms.

Medications are represented by generic sequence number (GSN) and diagnosis codes by ICD9 codes. Age was discretized by decade with a binary indicator for each decade. Patients are represented as a binary feature vector representing the presence or absence of each distinct diagnosis code, current medication, dispensed medication, word, discretized age value and sex. Observations that occur in fewer than 50 patients in the entire dataset were discarded, leaving a final binary feature vector of size 20,334.

4.3 Anchor specification

A single emergency physician specified anchors for each clinical state variable using our custom anchor elicitation tool with access to a database of 20,000 unlabeled patients chosen at random from the full patient set. Figure 3 shows the evolution of the performance of the learned classifier as the physician specified anchors. Unless otherwise noted, results are reported using the final set of anchors specified by the physician. Note that the physician was not provided with explicit feedback about the performance of the model on the ground truth labels, but was able to use the interface to determine which anchors were useful and make progress. In order to accurately assess

the effort involved in specifying anchors for a new classification task and to avoid overfitting, each anchor-based predictive model was only built once with the exception of `hasCancer` which was used as an example for development purposes.

4.4 Machine learning

We compare the classifiers learned using anchors to a simple rule-based baseline and a supervised machine learning baseline which uses a subset of the collected gold standard labels for training. Evaluation is reported on nine separate estimation tasks, one for each clinical state variable in Table 1. We emphasize that unlike the supervised baseline, the anchor algorithm does not require ground truth labels for training.

The rule-based baseline simply predicts positively when at least one anchor is present and negatively otherwise. This approach also requires no training, and is evaluated on the entire labeled set.

Evaluation of the supervised baseline is reported using 4-fold cross validation in order to fully utilize the limited number of gold standard labels we have for some of our clinical state variables. In each experiment, the labeled patients are divided into four equal-sized test sets. For each test set, a classifier is trained using a portion of the 75% of patients which are not in the test set ("training patients") and then used to predict for the 25% of patients designated as test. The results are averaged across the four test sets, giving an estimate of the performance on the entire labeled dataset. Each classifier of the supervised baseline is learned using at most 3000 training patients, representing approximately 3 weeks-worth of patients at our institution.

The supervised baseline was learned with logistic regression using the scikit-learn package [15] in Python. We use 5-fold cross validation within the train set to choose parameters, trying all combinations of the regularization constant (options are $\{10^{-6}, 10^{-5}, ..., 10^{6}\}$) and the norm used in regularization. Choices for the norm are L1 (encourages the learned classifier to use a minimal number of features by penalizing the sum of absolute values of the regression weights) or L2 (avoids overly emphasizing any one feature by penalizing the sum of squares of the regression weights). For the supervised results for 100 and 200 labels, 5-fold cross validation is not viable, due to the low number of positively labeled samples for each disposition question. Hence, for these numbers of labels the parameters are chosen to be the default values in scikit-learn. Lastly, we reduce regularization of the bias parameter by setting the "intercept_scaling" parameter to 1000.

The anchor method is trained using the specified anchors and 200,000 examples chosen randomly from the unlabeled dataset, and tested on the entire labeled set. Scikit-learn is used to fit logistic regression models as in the supervised setting, but holding the regularization norm fixed as L2 and doing cross-validation over the regularization parameter. Since the logistic regression models learned in the anchor method are meant to predict the presence or absence of the *anchor* (as described in Section 2.3), the cross-validation technique to choose parameters also uses the presence or absence of the anchor to measure performance, requiring no ground truth labels.

We measure performance using area under the ROC curve (AUC), a measure of the overall quality of a ranking predictor. Estimating the constant C in step 2 of the anchor algorithm is not necessary to obtain a ranking, so we omit that step. In the rule-based approach, ties are broken by counting the number of distinct anchors present in the patient record. In the anchor approach, ties among patients with anchors are broken according to the predicted probability of the latent variable ignoring the presence of the anchors.

4.5 Real-time decision support evaluation

We evaluate a real-time decision support scenario where the estimation tasks are performed without access to diagnosis codes (i.e., diagnosis codes are excluded from the feature vector), as these would usually be assigned after the patient leaves the emergency department. The supervised baseline and rule-based approach simply ignore all ICD9 codes because they cannot incorporate extra information that is not available at test time. However, since the algorithms are meant to be trained on previous patients, it is reasonable to assume that diagnosis codes would be available at the time of training. Thus, the anchor algorithm uses ICD9 codes as *anchors* during training, even though the resulting prediction rules do not use them as features.

We also tested a retrospective setting where the algorithms have access to diagnosis codes, both at train and test time. This setting is meaningful since our algorithm can also be used in retrospective settings for tasks such as cohort selection, information retrieval and quality control. The results were qualitatively similar, so we only present the real-time setting here.

5 Results

In this section we present results comparing the performance of the anchor-based learning method to the baseline methods on the task of predicting the nine clinical state variables listed previously in Table 1.

First, we show the utility of the anchor-specification interface described in Section 3. Figure 3 shows the learning path for the `hasCardiacEtiology` clinical state variable, describing the changes to AUC as the clinician added and subtracted anchors from the model. It is noteworthy that using our interface, the quality of the model tends to increase as time progresses. We observed this trend for all of the clinical state variables. Table 2 shows some of the final anchors specified by the clinician who used the interface. Using our interactive tool, the total time to specify anchors for all nine models was approximately 5 hours.

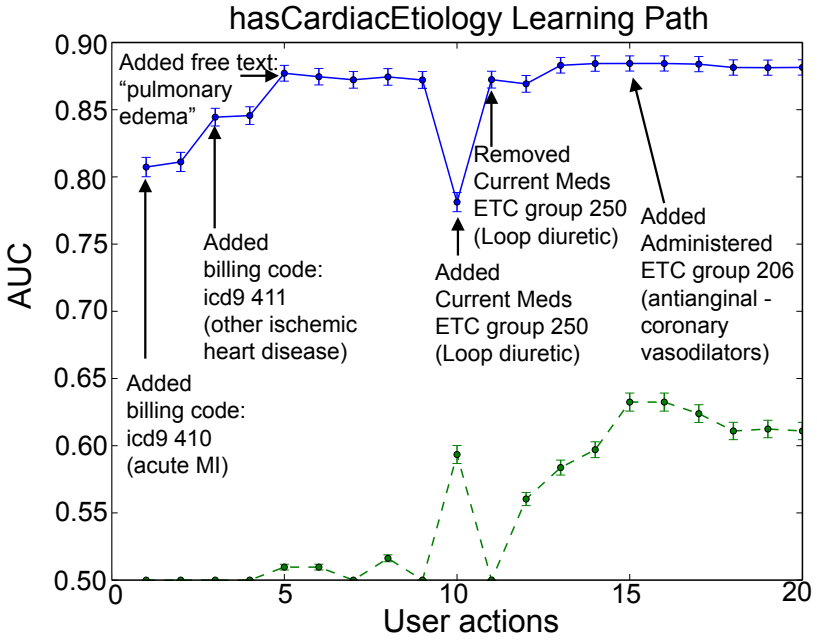

Figure 3: A learning path for one of the learned models (`hasCardiacEtiology`) as the interface is being used to build this model. On the x-axis are user actions, either additions to or deletions from the current set of anchors. On the y-axis is AUC evaluated on the gold standard collected labels. The dotted line shows the progress of the rules baseline which simply predicts 1 when an anchor is present and 0 otherwise. The rules baseline initially has an AUC of 0.5 since the earliest anchors are ICD9 codes and we consider the real-time setting where ICD9 codes are not available for the test set.

Clinical State Variable	Selected Anchors
didFall	Billing code [Accidental Falls] (ICD9 E880-E888), "slipped", "s/p-fall', "slip-fall", "fell down", "mechanical_fall", "witnessed-fall", …
hasSepticShock	Billing code [septic shock] (ICD9 785.52) , Administered meds from [cardiac sympathomimetics] (ETC:00003064)
isimmunosuppressed	Current meds from [antineoplastic - antimetabolite - folic acid analogs] (ETC:00002881), "immunocompromised", …
hascancer	Billing code [neoplasms] (ICD9 group 2), "breast-ca", "mets", "oncology", …

Table 2: A selection of anchors specified by a clinical collaborator. Anchors that utilize structured variables such as diagnosis codes or medications (current or administered) are mapped to a relevant structured ontology.

Table 3 shows a comparison to supervised learning for the real-time setting described in Section 4.5. For many of the clinical state variables, the anchor algorithm outperforms learning with 3K labels, suggesting that it would take a non-trivial amount of time and human effort to collect the data necessary to train in a supervised setting with comparable accuracy. The `didFall` task was evaluated on a biased population since the question was only asked for patients with a head CT scan. The supervised method was trained on a similarly biased population, so it makes sense that the performance of the anchor algorithm underperforms in this setting.

To provide a comparison to a semi-supervised algorithm, we experimented with training using SVMlight in a transductive setting [16], using up to 20,000 visit instances (500 labeled, 19500 unlabeled) for the `hasInfection`

variable. Samples were reweighted to account for the class imbalance. However, we found that this semi-supervised approach did not improve AUC over the supervised baseline with 500 labeled examples.

Variables	Rules	Supervised						Anchors
		100	200	500	1K	2K	3K (min, max)	
didFall	0.725 ± 0.008	0.814	0.852	0.900	0.914	0.920	**0.924** (0.917,0.934)	0.883 ± 0.006
hasCardiacEtiology	0.611 ± 0.006	0.772	0.827	0.824	0.875	0.900	**0.906** (0.891, 0.920)	0.881 ± 0.006
hasInfection	0.723 ± 0.002	0.728	0.767	0.804	0.830	0.861	0.883 (0.881, 0.886)	**0.903± 0.001**
fromNursingHome	0.620 ± 0.005	0.725	0.792	0.822	0.869	0.894	0.891 (0.873, 0.906)	**0.918 ± 0.004**
hasCancer	0.822 ± 0.018	0.635	0.673	0.693	0.810	0.882	0.902 (0.880, 0.930)	**0.945 ± 0.01**
hasPneumonia	-	0.856	0.907	0.933	0.947	0.956	0.963 (0.954, 0.972)	**0.971 ± 0.003**
isAnticoagulated	0.849 ± 0.03	-	-	-	-	-	-	**0.930 ± 0.02**
isImmunosuppressed	0.650 ± 0.01	0.584	0.659	0.740	0.814	0.842	**0.862** (0.840, 0.877)	0.840 ± 0.009
hasSepticShock	0.738 ± 0.02	-	0.760	0.773	0.863	0.920	0.952 (0.928, 0.967)	**0.967 ± 0.008**

Table 3: Comparing AUC in the real-time setting. The supervised method is trained using logistic regression with a small number of gold standard labels. When the anchors are composed entirely of diagnosis codes, the rules approach cannot be meaningfully evaluated on the test set (in the real-time setting, diagnosis codes are not available at test time). When we had insufficient data to train, the supervised approach could not be evaluated. Best methods in each row are bolded. The anchor approach uses 200K *unlabeled* examples in training. Standard errors of the AUC for Rules and Anchors are computed using 1000 bootstrap samples of the test set. Min and max values for the 3K supervised baseline are from the 4-fold cross validation.

6 Discussion

Across the nine clinical state variables considered in our evaluation, our anchor-based unsupervised learning algorithm obtains prediction accuracy comparable to and in many cases better than a supervised prediction algorithm. It is important to note that the clinical states we are interested in are precisely those for which ground truth labels cannot be easily derived from diagnosis codes or from natural language processing on the clinical notes. Labeling data would be expensive, time consuming, and in many cases institution-specific since the learned predictors may not generalize.

Surprisingly, despite the physician not receiving explicit feedback about the performance of the model on the ground truth label, using our user interface he was able to determine when adding an anchor helped or hurt overall performance (as seen by a generally monotonic increase in AUC with each additional anchor specified). Our initial user interface also allowed the physician to label individual patients as positive or negative for a given clinical state variable. However, despite trying various ways of integrating this feedback into our learning algorithm, we found negligible gains in accuracy compared to only using the anchors.

There are several interesting directions for future work. Our current approach predicts each clinical state variable independently, but it would also be interesting to jointly model the clinical states. Doing so may provide a solution for the vexing problem of how to efficiently provide the learning algorithm *negative* feedback, which could be addressed by introducing additional variables (and anchors for them). We could then use negative correlations between clinical state variables to disambiguate commonly confused clinical states.

Although our algorithm can provide useful predictions without any labeled data, its use within a clinical setting presents opportunities to gather additional data to improve its performance. One direction for future work would be to develop a feedback mechanism (implicit or explicit) to learn from the algorithm's use. We also plan to integrate our predictions into an active learning algorithm to be used with the current system that asks questions prospectively at the time of disposition from the ED. We currently rotate through the questions asked so as to not overburden the clinician. Instead, we can carefully select which questions to ask for the specific patient at hand so as to best improve our prediction algorithms for a range of clinical state variables.

The most important next step will be to test the generalization of anchor-based learning in departments other than the ED and in other institutions. We are also continuing to use the user interface to specify anchors for dozens of additional clinical state variables, using these to enable new contextual user interfaces and to trigger decision support.

Acknowledgments

This work is partially supported by a Google Faculty Research Award, grant UL1 TR000038 from NCATS, NIH and CIMIT Award No. 12-1262 under U.S. Army Medical Research Acquisition Activity Cooperative Agreement W81XWH-09-2-0001. Yoni Halpern was supported by an NSERC Postgraduate Scholarship. The information contained herein does not necessarily reflect the position or policy of the Government, and no official endorsement should be inferred.

References

[1] Nachimuthu SK, Haug PJ. Early detection of sepsis in the emergency department using dynamic Bayesian networks. In: AMIA Annual Symposium Proceedings. vol. 2012. American Medical Informatics Association; 2012. p. 653.

[2] Kawaler E, Cobian A, Peissig P, Cross D, Yale S, Craven M. Learning to predict post-hospitalization VTE risk from EHR data. In: AMIA Annual Symposium Proceedings. vol. 2012. American Medical Informatics Association; 2012. p. 436.

[3] DeLisle S, South B, Anthony JA, Kalp E, Gundlapallli A, Curriero FC, et al. Combining free text and structured electronic medical record entries to detect acute respiratory infections. PloS one. 2010;5(10):e13377.

[4] Liu M, Shah A, Jiang M, Peterson NB, Dai Q, Aldrich MC, et al. A study of transportability of an existing smoking status detection module across institutions. In: AMIA Annual Symposium Proceedings. vol. 2012. American Medical Informatics Association; 2012. p. 577.

[5] Carroll RJ, Thompson WK, Eyler AE, Mandelin AM, Cai T, Zink RM, et al. Portability of an algorithm to identify rheumatoid arthritis in electronic health records. Journal of the American Medical Informatics Association. 2012;19(e1):e162–e169.

[6] Shivade C, Raghavan P, Fosler-Lussier E, Embi PJ, Elhadad N, Johnson SB, et al. A review of approaches to identifying patient phenotype cohorts using electronic health records. Journal of the American Medical Informatics Association. 2013;.

[7] Friedman C, Elhadad N. Natural language processing in health care and biomedicine. In: Biomedical Informatics. Springer; 2014. p. 255–284.

[8] Smith SW, Koppel R. Healthcare information technology's relativity problems: a typology of how patients' physical reality, clinicians' mental models, and healthcare information technology differ. Journal of the American Medical Informatics Association. 2013;.

[9] Elkan C, Noto K. Learning classifiers from only positive and unlabeled data. In: KDD; 2008. p. 213–220.

[10] Natarajan N, Dhillon I, Ravikumar P, Tewari A. Learning with noisy labels. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ, editors. Advances in Neural Information Processing Systems 26; 2013. p. 1196–1204.

[11] Tanz RR, Gerber MA, Kabat W, Rippe J, Seshadri R, Shulman ST. Performance of a rapid antigen-detection test and throat culture in community pediatric offices: implications for management of pharyngitis. Pediatrics. 2009;123(2):437–44.

[12] Dasgupta S. Two faces of active learning. Theoretical Computer Science. 2011;412(19):1767 – 1781. Algorithmic Learning Theory (ALT 2009).

[13] Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BG. A simple algorithm for identifying negated findings and diseases in discharge summaries. Journal of biomedical informatics. 2001;34(5):301–310.

[14] Jernite Y, Halpern Y, Horng S, Sontag D. Predicting chief complaints at triage time in the emergency department. NIPS 2013 Workshop on Machine Learning for Clinical Data Analysis and Healthcare. 2013;.

[15] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al.. Scikit-learn: machine learning in python; 2011. `http://scikit-learn.org/0.14/`.

[16] Joachims T. Transductive inference for text classification using support vector machines. In: International Conference on Machine Learning (ICML). Bled, Slowenien; 1999. p. 200–209.