Hierarchical Information Criterion for Variable Abstraction

Mark Mirtchouk MMIRTCHO@STEVENS.EDU
Computer Science
Stevens Institute of Technology
Hoboken, NJ, USA

Bharat Srikishan BSRIKISH@STEVENS.EDU
Computer Science
Stevens Institute of Technology
Hoboken, NJ, USA

Samantha Kleinberg SAMANTHA.KLEINBERG@STEVENS.EDU
Computer Science
Stevens Institute of Technology
Hoboken, NJ, USA

Abstract

Large biomedical datasets can contain thousands of variables, creating challenges for machine learning tasks such as causal inference and prediction. Feature selection and ranking methods have been developed to reduce the number of variables and determine which are most important. However in many cases, such as in classification from diagnosis codes, ontologies, and controlled vocabularies, we must choose not only which variables to include but also at what level of granularity. ICD-9 codes, for example, are arranged in a hierarchy, and a user must decide at what level codes should be analyzed. Thus it is currently up to a researcher to decide whether to use any diagnosis of diabetes or whether to distinguish between specific forms, such as Type 2 diabetes with renal complications versus without mention of complications. Currently, there is no existing method that can automatically make this determination and methods for feature selection do not exploit this hierarchical information, which is found in other areas including nutrition (hierarchies of foods), and bioinformatics (hierarchical relationship of genes). To address this, we propose a novel Hierarchical Information Criterion (HIC) that builds on mutual information and allows fully automated abstraction of variables. Using HIC allows us to rank hierarchical features and select the ones with the highest score. We show that this significantly improves performance by an average AUROC of 0.053 over traditional feature selection methods and hand crafted features on two mortality prediction tasks using MIMIC-III ICU data. Our method also improves on the state of the art (Fu et al., 2019) with an AUROC increase from 0.819 to 0.887.

1. Introduction

Healthcare datasets are often very large, with potentially thousands of variables. Using all of these variables for classification at the same time is computationally expensive, and not all variables are informative. Some may have little data, while others may require pre-processing to be most useful. For tasks such as classification and prediction researchers often take advantage of controlled vocabularies and ontologies such as the ICD-9 hierarchy, which provide a standardized way of structuring information such as diagnosis codes. In the ICD-9 hierarchy top level codes represent broad categories (e.g. diabetes, heart failure) while lower levels represent more specific diagnoses within these (e.g. Type 2 diabetes with ketoacidosis, acute diastolic heart failure). However these hierarchies can contain thousands of items and choosing the correct level of the hierarchy for a learning algorithm is often a manual task performed by researchers. Yet the choice of granularity – whether to consider all diagnoses in a tree as the same variable for a task – has a significant impact on what inferences are possible and whether they will be correct. For example, overweight and obesity may be distinct conditions when examining risk of one disease, but the distinction may not matter for another.

Relying on researchers to decide prior to learning what level of granularity to use for each variable is time intensive and can lead to suboptimal results. Singh et al. (2014) demonstrated that using just the root nodes of the ICD-9 code hierarchy has a significant impact on accuracy in comparison to probability based features for the tasks of forecasting chronic kidney disease progression and predicting heart failure. Further, there is a significant class imbalance problem, and using highly specific codes may mean there are few instances of a given condition. Existing methods for feature selection can aid in choosing which diagnosis codes may be most relevant for prediction, but do not account for the hierarchy or the depth of the variable in the hierarchy. This problem has been studied previously in the context of determining semantically similar concepts from medical notes (Pivovarov and Elhadad, 2012), but there is not yet an automated method for determining the best variable granularity for a given problem. Feature ranking is one way to solve this problem, where each feature is given a score and only the top N features are used for learning. While there have been tremendous advances in feature ranking (Huang et al., 2014; Kononenko, 1994; Urbanowicz et al., 2018), most methods treat each feature independently, ignoring useful information, such as hierarchical structure, which results in lower accuracy.

To address the problem of hierarchical variable abstraction, we introduce the Hierarchical Information Criterion (HIC), a novel hierarchical feature ranking approach. Our approach builds on mutual information, and incorporates information about variable structure as well as the amount of data available. This allows our approach to remain robust even with outlying values in sparse branches of a tree, and ensures the variables chosen contribute meaningfully to accuracy. In particular, we focus on the ICD-9 code hierarchy, which contains thousands of codes. For the task of mortality prediction, we use the hierarchically structured ICD-9 codes as features to classify the binary labels of in-hospital mortality. Given the size, it has been difficult to move beyond hand crafted features (e.g. using leaves in the hierarchy or a different set cutoff, when the right level can vary across conditions). HIC can be used not only to determine what level of granularity to use, but also can be used to improve upon manually created scores such as Clinical Classifications Software (CCS) (Elixhauser et al., 2015), which has thousands of such choices. HIC improves these scores by making them task specific. We apply this approach to the task of mortality prediction in intensive care unit (ICU) data using the MIMIC-III dataset (Johnson et al., 2016). HIC outperformed existing feature ranking baselines and the previous state of the art (Fu et al., 2019) on the task of mortality prediction for all patients in MIMIC-III by AUROC of 0.068. Our proposed method enables efficient feature selection that is robust to outliers, and improves performance on healthcare classification and prediction tasks with ICD-9 diagnosis codes.

Generalizable Insights about Machine Learning in the Context of Healthcare

- Exploiting the structure of and relationships between hierarchical features results in better classification performance when using ICD-9 codes.

- Integrating mutual information, statistical significance, and sample size improve hierarchical feature selection by identifying features with high predictive value while effectively excluding outliers.

2. Related Work

Hierarchical feature ranking and feature selection are critical for the accuracy and interpretability of healthcare applications such as classification and prediction. In addition to large numbers of features potentially leading to increases in training time, they may also lead to overfitting the data (e.g. prediction from ICD-9 codes that occur rarely). Additionally, choosing the correct level of hierarchical features is task dependent and has a significant impact on performance. Modern methods for feature ranking provide a partial solution to our problem, but are not tuned to hierarchical data and ignore variable level in the hierarchy, which improves predictive performance as shown in our experiments. Feature representation learning methods give good performance in terms of accuracy for healthcare tasks, but are difficult to interpret and do not tell the user which features are most important. Finally, existing ICD-9 code based scores rely on expert knowledge to create and cannot easily be adapted to new tasks. We now review existing feature ranking algorithms, feature representation learning methods, and healthcare research using ICD-9 codes.

2.1. Feature Ranking

Feature ranking algorithms fall into three categories: wrapper methods, embedded methods, and filter methods. Wrapper-based feature ranking methods build models and use the classification performance as a measure of which feature is the most useful. SVM-RFE (Guyon et al., 2004; Huang et al., 2014) trains an SVM, ranks all the features based on learned feature weights, and recursively eliminates the worst feature until the desired N are left. Computationally, this is very expensive as to get the top N features from M variables, one must build $M-N$ models, which could be thousands, depending on the original size and final number of variables. With thousands of variables in the ICD-9 hierarchy, these methods become infeasible and further do not make use of the hierarchical structure.

Embedded feature ranking methods build a model from the data and extract the top-K features based on a score function such as Gini impurity or entropy during the construction of the model. Menze et al. (2009) build a Random Forest using Gini impurity to rank the features. This approach builds one model, which improves computation time, but this approach gives preference to features with high cardinality. Further, correlated features will have lower importance than uncorrelated features. Embedded methods do not use the hierarchical information present in ICD-9 code data and this can result in selecting sub-optimal features.

Filter feature ranking methods take a different approach, examining the correlation between the input variable (X) and the outcome (Y). One such method is mutual information (MI) (Burks et al., 1951; Battiti, 1994). MI can be limited in that it does not take into account the sample size associated with different variables. In real-world health data, where variables may be diagnoses, there may be significant variation. Yet in MI, a feature that occurs only once may appear to be more important than one that occurs many times but only occurs along with an outcome 95% of the time. This is a major limitation when working with ICD-9 codes, as there may be hundreds of instances when grouping codes together at the highest level, while the individual leaf nodes may occur only once or twice. Despite their high MI, features based on such sparse data are not necessarily meaningful or trustworthy.

Relief (Kira and Rendell, 1992) is a filter based feature ranking method that inspired many others such as ReliefF (Kononenko, 1994), SURF (Greene et al., 2009), and MultiSURF (Urbanowicz et al., 2018). Relief calculates a feature score based on Euclidean distance (L2 norm) to the closest same-class and different-class instances. This algorithm must run multiple times, making it computationally expensive. ReliefF improves on Relief's performance by using the Manhattan distance (L1 norm) as well as finding the k nearest neighbors and averaging their contribution, but this approach may incorrectly rank features at the extremes (i.e. outliers). SURF improves upon ReliefF through eliminating outliers by using a distance threshold. MultiSURF introduces a dead zone, where some neighbors do not contribute to the score. In summary, Relief-based methods do not distinguish between redundant features and do not work well with low sample size. Yet ICD9-codes may be redundant in the context of a specific task and leaf nodes in the hierarchy often have low sample size because while rare diagnoses are common, few patients have each specific diagnosis.

None of the aforementioned works allow ranking of hierarchical features. Work on hierarchical feature selection includes Select Hierarchical Information-Preserving features (Wan et al., 2015), Select Most Relevant features (Wan et al., 2015), and the combination HIP–MR (Wan and Freitas, 2013). These methods focus on finding feature redundancy rather than optimizing the level of a hierarchy for each feature. That is, they output which features can be eliminated rather than which are most important at what level of granularity. Thus, these methods are not applicable to our problem of choosing which hierarchical features will perform best for mortality prediction with ICD-9 codes.

2.2. Feature representations

Feature representation learning methods focus on learning representations of electronic health records (EHRs) including ICD-9 codes. Pivovarov and Elhadad (2012) used medical notes with the Systemized Nomenclature of Medicine – Clinical Terms (SNOMED-CT), a hierarchical ontology connecting related medical concepts together to determine semantic similarity between concept pairs. This method incorporates a hierarchical ontology with unstructured data and focuses on finding the most semantically similar concepts, but does not rank which concepts are most important for a particular task such as prediction or classification.

Other approaches have focused on structured data and have used deep learning to learn non-linear representations of EHR data. Choi et al. (2018) used ICD-9 diagnosis codes, procedure codes, and medication codes to learn self-supervised embeddings for heart failure and sequential disease prediction. Both Mao et al. (2019) and Choi et al. (2020) combined different modalities of EHR data to learn medical entity representations using graph convolutional networks for medication recommendation and readmission prediction tasks respectively. GRAM is a graph-based model that combines a medical ontology with the EHR using an attention mechanism to improve diagnoses predictions (Choi et al., 2017). These feature representation methods may lead to high accuracy for a task, but do not give the researcher an idea of which features at what level of granularity are most important. Specifically, dense non-linear representations are difficult to interpret for researchers and clinicians because they do not correspond directly to ICD-9 codes but rather are complex functions of the data. They provide little insight into which ICD-9 codes are most important for a given task. Deep Dictionary Learning (DDL) has been used to create moe succinct data representations for EHR data Fu et al. (2019). However as this is not a feature selection approach, it has similar limitations to the other methods discussed. In contrast, our approach can be used prior to any classification or potentially causal inference task as a pre-processing step.

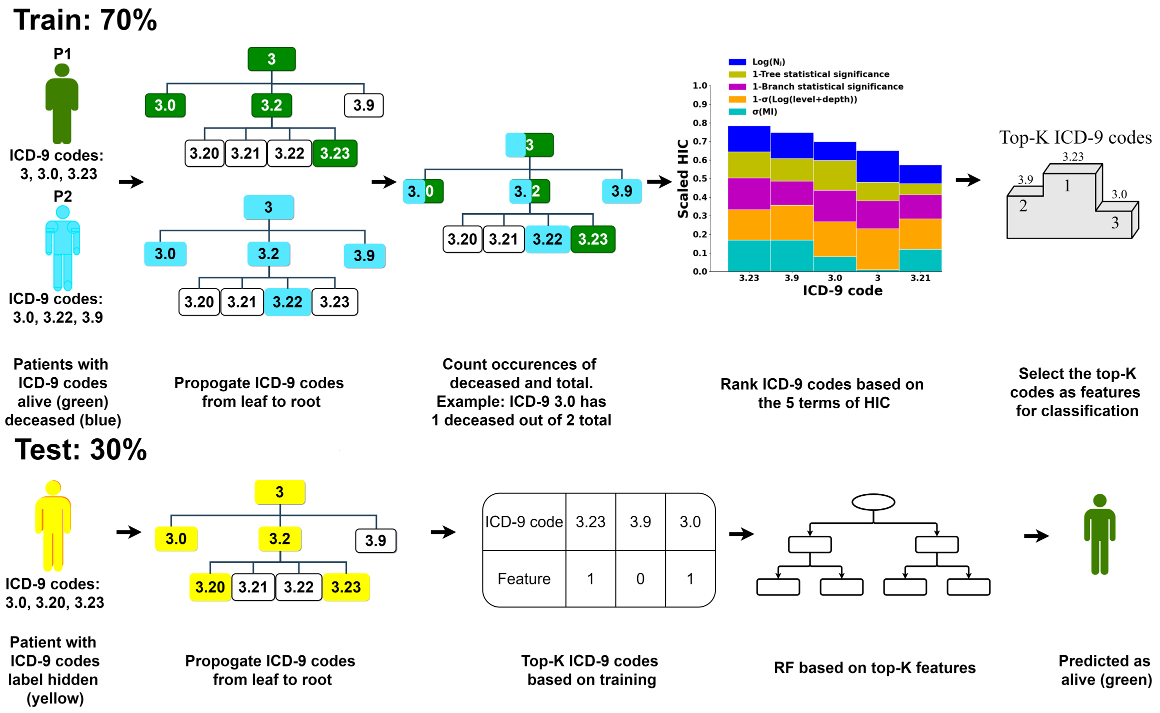
2.3. ICD-9 code based features

While no work aims to automatically select the best granularity for hierarchical features, there exist many manually created scores that use ICD-9 scores. A common approach is to group a subset of ICD-9 codes together into categories while assigning each category a score. This approach has been beneficial because these scores were created in collaboration with clinicians and they provide a baseline for machine learning models in healthcare. While these scores lose information through the aggregation of codes, they are consistent with existing medical knowledge. Diabetes Complications Severity Index (DCSI) (Young et al., 2008), Elixhauser Comorbidity Index (Elix) (Elixhauser et al., 1998), and Charlson Comorbidity Index (CCI) (Quan et al., 2005), all combine ICD-9 codes to create one score. Clinical Classifications Software (CCS) (Elixhauser et al., 2015) groups all ICD-9 codes into 283 mutually exclusive categories. Each of these categories contains predefined diagnosis codes which are a subset of the ICD-9 hierarchy and may not be the best predictive codes for a specific task. In our experiments, we demonstrate that these predefined scores perform worse than using HIC to automatically select the best performing ICD-9 diagnosis codes. Singh et al. (2014) used ICD-9 codes with 3 different feature representations to evaluate performance on the tasks of predicting chronic kidney disease progression and incident heart failure. They show that simply using the root nodes of the ICD-9 diagnosis hierarchy performs worse than using different levels of the hierarchy while incorporating hierarchical information during feature construction. Inspired by this, our method exploits hierarchical information in addition to mutual information and statistical significance to further improve accuracy for the task of mortality prediction.

Despite hierarchical feature ranking being critical for the accuracy and interpretability of machine learning models in healthcare, methods for such data are lacking. While existing feature ranking methods provide an ordering of features, they ignore hierarchical structure and are not appropriate for the hierarchical features in healthcare including ICD-9 diagnosis codes. Feature representation learning methods can perform well for specific tasks in healthcare but provide little interpretability for clinicians and cannot determine which hierarchical features are most important for a specific task. Existing ICD-9 code based features such as aggregate scores are static and do not adapt automatically to the healthcare task at hand, potentially ignoring useful features that could improve performance. Methods that assign ICD-9 codes to unstructured text do make use of the hierarchical structure of the data but do not address our goal of automated selection of the appropriate level of variable granularity. Building on the ideas of mutual information and exploiting hierarchical structure, we now introduce the hierarchical information criterion (HIC) for feature ranking with hierarchical variables.

3. Methods

The HIC method incorporates information about variable structure and the amount of data available into the existing ranking method of mutual information. This allows HIC to be robust to outliers, ensuring all features contribute positively to accuracy. Existing feature ranking methods could select features that negatively impact accuracy, including features that have high mutual information, but low support in the data distribution.


Train: 70%
P1
ICD-9 codes:
3, 3.0, 3.23
P2
ICD-9 codes:
3.0, 3.22, 3.9
Patients with ICD-9 codes alive (green) deceased (blue)
3
3.0
3.2
3.9
3.20
3.21
3.22
3.23
Propogate ICD-9 codes from leaf to root
Count occurences of deceased and total. Example: ICD-9 3.0 has 1 deceased out of 2 total
Log(Nⱼ)
1-Tree statistical significance
1-Branch statistical significance
1-σ(Log(level+depth))
σ(MI)
Scaled HIC
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
3.23
3.9
3.0
3
3.21
ICD-9 code
Rank ICD-9 codes based on the 5 terms of HIC
Top-K ICD-9 codes
3.23
3.9
3.0
1
2
3
Select the top-K codes as features for classification
Test: 30%
ICD-9 codes:
3.0, 3.20, 3.23
Patient with ICD-9 codes label hidden (yellow)
Propogate ICD-9 codes from leaf to root
Top-K ICD-9 codes based on training
RF based on top-K features
Predicted as alive (green)

ICD-9 code	3.23	3.9	3.0
Feature	1	0	1

Figure 1: Patients and ICD-9 codes are used to generate a classifier based on the Top-K features.

3.1. Overview

We focus on ranking variables that are structured in a hierarchy (e.g. Figure 1). Given data X and binary label Y, the first component of the HIC method is based on MI (Burks et al., 1951; Battiti, 1994), which calculates the dependence between variables X and Y. The higher the mutual information, the more predictive X is of Y. We compute mutual information (equation 8) by partitioning X into finite size bins. For our problem, each bin represents the number of people that have a specific ICD-9 code in their medical record. The second component of HIC is based on the level of the hierarchy, which penalizes the variable level and helps determine which level of granularity performs best. Level is used as a penalty term because of data sufficiency as features at higher levels have more support (e.g. more patients). The third and fourth component of HIC are based on the two-tailed z-score between all nodes in the same branch and tree respectively. The z-score is computed based on the conditional probability of the outcome given each node. We penalize nodes that have similar probabilities resulting in a more diverse range of nodes selected. The z-scores are computed at both at a branch and tree level. Both are important as the z-score per branch focuses on a specific code (e.g. a specific ICD-9 code) and the z-score per tree generalizes by looking at the whole hierarchy (e.g. the whole ICD-9 code tree). The fifth and last component of HIC is based on the number of times a feature occurs. This last component is important as it helps with handling low support features. A feature that has few occurrences could potentially have a high empirical probability (≈ 1), but the last term of HIC dampens the score.

3.2. Preliminaries

Let $X = \{X_1, X_2, \ldots, X_n\}$ be a two dimensional list with each X_i containing m binary features $x_{i,1}, x_{i,2}, \ldots, x_{i,m}$. Each $X_i \in X$ has a binary label $Y_i \in \{Y_1, Y_2, ..., Y_n\}$ respectively. The binary variable $x_{i,j}$ is 1 if person i has feature j, and is 0 otherwise. Let $X_{i,:}$ be the set of all features for person i, while $X_{:,j}$ is the set of all the people with feature j. In this work, $X_{i,j} = 1$ indicates that person i has ICD-9 code j in their record.

We refer to the depth of variable $X_{:,j}$ in the hierarchy H as $level_j$, starting at 1. Let the max level of the branch for $X_{:,j}$ be called max_level_j. For example, ICD-9 code 3.0 has $max\_level_{3.0} = 2$ but ICD-9 code 3.2 has $max\_level_{3.2} = 3$. Each $x_{i,j} \in X_{:,j}$ is associated with a node in the hierarchy. For convenience, let Q_j be defined as the list $[(X_{:,j}), (X_{:,\neg j})]$, where $\neg j$ are all the nodes that are not j. To compute probabilities, since we have binary labels, we iterate over each $j \in H$. Let M_j be the number of people with feature j that have the label of 1. Let $M_{\neg j}$ be the number of people that do not have feature j that have the label of 1. Let N_j be the number of people with feature j. Let $N_{\neg j}$ be the number of people that do not have feature j. Let E_y be the number of people with label y. The probability of each feature is based on its frequency. Thus, the probabilities of x, y, and $x \cap y$ are $P(x) = N_j/|X|$, $P(y) = E_y/|X|$, and $P(x, y) = M_j/N_j$ respectively. We use the sigmoid function (σ) to transform the answer space to a range of [0,1].

3.3. Statistical significance

Using the definition of two-tailed z-score, we construct two types of statistical significance metrics for hierarchical features. The first is a two-tailed z-score comparing each feature in a branch to every other feature in the same branch, which we call *branch statistical significance*.

Suppose we are calculating the branch statistical significance for a feature j in its branch V. We iterate over all features $v \in V, v \neq j$ and compare against j. For the running example of mortality prediction let M_v, N_v be the number of people dead and the total number of people respectively with feature v in the current branch. For each $v \in V, v \neq j$, let $\mathrm{P}(v) = M_v/N_v$. The overall sample proportion is given by the equation:

$$\hat{p} = \frac{M_j + M_v}{N_j + N_v} \tag{1}$$

The minimum z-score of the z-test for comparing two population proportions for each branch z_b is given by the equation:

$$z_b = \min_{v \in V, v \neq j} \frac{P(j) - P(v)}{\sqrt{\hat{p}(1-\hat{p})(\frac{1}{N_j} + \frac{1}{N_v})}} \tag{2}$$

The two-tailed probability based on the current branch minimum z-score, which we call branch statistical significance, is $P(Z > z_b \cup Z < -z_b)$.

Next we calculate a two-tailed z-score between all pairs of features in the tree, called *tree statistical significance*.

We iterate over each node in the hierarchy $h \in H, h \neq j$ (e.g. ICD-9 codes) and compare it against the current node j. For the running example of mortality prediction, let M_h, N_h be the number of people dead and the total number of people respectively having feature h in the tree. For each $h \in H, h \neq j$, let $\mathrm{P}(v) = M_h/N_h$. The overall sample proportion is given by the equation:

$$\tilde{p} = \frac{M_j + M_h}{N_j + N_h} \tag{3}$$

The minimum z-score of the z-test for comparing two population proportions for each tree z_t is given by the equation:

$$z_t = \min_{h \in H, h \neq j} \frac{P(j) - P(h)}{\sqrt{\tilde{p}(1 - \tilde{p})(\frac{1}{N_j} + \frac{1}{N_h})}} \tag{4}$$

The two-tailed probability based on the minimum z-score of whole tree, which we call tree statistical significance, is $P(Z > z_t \cup Z < -z_t)$.

3.4. Weights w_b and w_t

Both the branch and tree statistical significance have limitations. The branch statistical significance does not utilize every node in the hierarchy as it only compares the current node to features in the same branch. The tree statistical significance compares the current node to every other node in the hierarchy but does not perform well for features associated with less data in the tree as the z-score is less reliable when sample size is small. Because we are making a separate comparison against every node in the tree, the tree statistical significance has a much higher chance of being close to one. These limitations can be mitigated by combining both significance scores into one number. We combine the branch and tree statistical significance scores using a convex combination of weights based on the sample size.

The weights of branch and tree statistical significance are determined by equation 5 which calculates α as a ratio of the expected number of people with code j divided by the total number of data points and size of the hierarchy. As α increases, the weight of the tree statistical significance w_t goes up. Therefore the more people having code j in our data, the more α increases and correspondingly the weight of the tree statistical significance w_t for that code. If the number of people with code j is small in our data, we prefer to weight the branch statistical significance w_b more highly as the sample size will be too small for the tree statistical significance to be accurate.

$$\alpha = \frac{\sum_{j \in X} \left(P(j) * \log{(N_j)}\right)}{\log(|X|) * \log(|H|)} \tag{5}$$

$$w_b + \alpha * w_b = 1 \implies w_b = \frac{1}{\alpha + 1} \tag{6}$$

$$w_t + w_b = 1 \implies w_t = \frac{\alpha}{\alpha + 1} \tag{7}$$

3.5. Hierarchical Information Criterion

MI is used for measuring the dependence between variables X and Y, but it does not take into account statistical significance nor the level of each variable. The MI equation is:

$$MI(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x, y) \log \frac{P(x, y)}{P(x)P(y)}. \tag{8}$$

We propose a novel formula that incorporates not only MI, but also the level of the hierarchy, statistical significance of the variables in each hierarchy branch, and the statistical significance of the variables in the whole hierarchy. We dampen the effect of some variables by using the sigmoid function to avoid quantities going to infinity and therefore skewing the final result. Furthermore, we add a term to account for the number of samples N_j. The next two terms are based on statistical significance: z_b is based on each each individual branch $v \in V$ while z_t is based on the whole hierarchy (traverses all variables in H). The weights w_b and w_t are based on α as described in Section 3.4. We call this combined method Hierarchical Information Criterion (HIC). See algorithm 1 for details.

$$\begin{aligned} HIC(X,Y,j) = \sigma\Big(\sum_{x\in Q_j} \sum_{y\in Y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \Big) \\ -\sigma\Big(\log_{max\_level_j}(level_j + max\_level_j) \Big) \\ -w_b * P(Z > z_b \cup Z < -z_b) \\ -w_t * P(Z > z_t \cup Z < -z_t) \\ +\log_{|X|}(N_j) \end{aligned} \tag{9}$$

4. Experiments

We now evaluate the proposed hierarchical feature ranking approach using the MIMIC-III ICU dataset. MIMIC-III has been used for many tasks including mortality prediction and allows comparison against feature ranking baselines and prior works on mortality prediction (Fu et al., 2019) as well as mortality prediction specifically for individuals with diabetes (Anand et al., 2018). Similar to both papers, we use the area under the receiver operating characteristic curve (AUROC) as the main evaluation metric.

4.1. Data

MIMIC-III (Johnson et al., 2016) is an ICU dataset consisting of approximately 50,000 critical care patients at Beth Israel Deaconess Medical Center from the year 2001 until 2012. It includes different data sources and types such as demographics, lab tests, procedures, medications, and clinical notes. We focus here on feature ranking with hierarchical data, and thus use the ICD-9 code data from the diagnoses table.

4.2. Baseline methods

For both experiments, we compare our HIC method to the most common and best performing feature ranking methods found in the literature:

- MI (Burks et al., 1951): Features are ranked using mutual information (Equation 8). Similar to HIC, we approximate X by using Q_j: $Q_j \in \{(X_{:,j}), (X_{:,\neg j})\}$, where $\neg j$ are all the nodes that are not j.
- SVM-RFE (Guyon et al., 2004; Huang et al., 2014): We build a Support Vector Regression (SVR) with a linear kernel, regularization parameter $C = 1.0$, and set epsilon as 0.1. To rank features, we build a recursive feature eliminator (RFE) with a step size of 1.

Algorithm 1: HIC ranking

```
Input:
    $x \in H$, a node $x$ in the hierarchy of features $H$, with the size of $H$ being $size$
    $branch(x)$, a function that returns all the nodes associated to the branch of $x$
    $level(x), max\_level(x)$, functions that return the level and max_level of node $x$ respectively
    $y \in Y$, a binary label associated to each $x$
    $p$, a dictionary of probabilities for each $x$, $\neg x$, $y$, $\neg y$, and combination
    $w_b$, $w_t$, the weights of the branch and tree statistical significances respectively
    $num(x)$, a function that returns the number of samples having feature $x$
    $\sigma(a)$, the sigmoid function returns $\frac{1}{1+e^a}$
    $ztest(a, b)$, a function that returns the 2-proportion z-test between probabilities $a$ and $b$
    $zprob(a)$, a function that calculates $P(Z > a \cup Z < -a)$
Output:
    $X_{new}$: an array of the tuples: (node, corresponding HIC score)
    $X_{new} \leftarrow []$
    for $x \in H$ do
        $MI \leftarrow p[x, y = 1] * \log \frac{p[x,y=1]}{p[x]*p[y=1]}$
        $MI \leftarrow MI + p[\neg x, y = 1] * \log \frac{p[\neg x,y=1]}{p[\neg x]*p[y=1]}$
        $MI \leftarrow MI + p[x, y = 0] * \log \frac{p[x,y=0]}{p[x]*p[y=0]}$
        $MI \leftarrow MI + p[\neg x, y = 0] * \log \frac{p[\neg x,y=0]}{p[\neg x]*p[y=0]}$
        $lvlterm \leftarrow \log_{max\_level(x)} (level(x) + max\_level(x))$
        $min_b \leftarrow \infty, min_t \leftarrow \infty$
        for $b \in branch(x), b \neq x$ do
            $z \leftarrow ztest(p[x], p[b])$
            if $z < min_b$ then
                $min_b \leftarrow z$
        for $t \in H, t \neq x$ do
            $z \leftarrow ztest(p[x], p[t])$
            if $z < min_t$ then
                $min_t \leftarrow z$
        $HIC \leftarrow \sigma(MI) - \sigma(lvlterm) - w_b * zprob(min_b) - w_t * zprob(min_t) + \log_{size}\big(num(x)\big)$
        $X_{new}.append\big((x, HIC)\big)$
    return $X_{new}$
```

- RF Gini (Menze et al., 2009): We rank features with Random Forest using Gini impurity and the same parameters as (Anand et al., 2018).

- ReliefF (Kononenko, 1994): We set the parameter of nearest neighbors to 100 based on an approximation of the maximal number of nodes possible in a branch of the ICD-9 code hierarchy.

- SURF (Greene et al., 2009): Features are ranked using SURF based on the ReBATE python package with default parameters.

- MultiSURF (Urbanowicz et al., 2018): Features are ranked using multiSURF based on the ReBATE python package with default parameters.

4.3. Mortality prediction for individuals with diabetes

We compare our and other methods using the the task of mortality prediction for a cohort of individuals with diabetes as in Anand et al. (2018). The inclusion criteria is taken from Anand et al. (2018), namely: all patients with an ICD-9 code of 250.0-259.0 inclusive that have both hemoglobin A1c (HbA1c) and blood glucose values recorded. Based on Anand et al. (2018), we chose the admission for each patient (based on HADM_ID) with the highest values in terms of Diabetes Complications Severity Index (DCSI) for each patient. We extracted the same eleven features as Anand et al. (2018). Three features were based on ICD-9 codes: DCSI score, Elixhauser score, and CCI score. Eight features were not based on ICD-9 codes: mean HbA1c, mean glucose during stay, admission type, age bracket, gender, ethnicity, insurance, and insulin status for predicting mortality. These non-ICD-9 code features are not evaluated by HIC but still used for our diabetes mortality prediction experiments for the sake of a fair comparison with Anand et al. (2018). For classification, we use the same experimental setup as Anand et al. (2018): five-fold cross validation of Random Forest with 5000 trees, 3 features randomly selected at each node, a max tree depth of 10, and a max of 5 observations in leaf nodes with a data split of 70% for training and 30% for testing, with the label of in-hospital mortality.

Applying the criteria of Anand et al. (2018) yielded 4111 patients with 382 dead (Anand). Even though we followed their inclusion criteria, this yielded 4196 patients with 413 dead (Anand Recreate).

For the Restricted and All experiments, using the HIC method and other feature ranking baselines, we construct different versions of each of these scores (DCSI, Elix, CCI) using those respective methods to select which codes will be included in each score. We first use a subset of the ICD-9 codes (called Restricted) based on the ICD-9 code features present in DCSI score, Elixhauser score, CCI score. We then do not restrict the ICD-9 code hierarchy and run algorithms on all the ICD-9 codes 001-999 as well as V01-V99 (called All). For all three classification tasks of Restricted, All, and Top-K, we excluded the codes that begin with E (E000-E999) as they are supplementary and 348.82, 761.6, and 798 as they can leak information about mortality. The number of codes in each score are: Elix=320, CCI=71, DCSI=160, with a total of 551. Based on the formula in section 3.4, w_b and w_t are 0.452 and 0.548 respectively.

4.3.1. RESTRICTED

As mentioned above, we use the same number of features as Anand et al. (2018) for comparison. Using all feature ranking methods, we transform the ICD-9 code based scores of: DCSI score, Elixhauser score, and CCI score. To transform the scores, we apply feature selection to each branch of the each ICD-9 code in the score. In total, we have 11 features (same as Anand et al. (2018)).

4.3.2. ALL

In the All experiment, we use 11 features (same as Anand et al. (2018)): 8 non-ICD-9 code based features and 3 ICD-9 code based features. The construction of the 3 ICD-9 code based features differs from the Restricted experiment as described below.

Unlike the Restricted experiment where we can select a branch and chose between the < 100 different ICD-9 codes, in the All experiment we use the whole ICD-9 hierarchy that consists of 13270 ICD-9 codes. For each of the feature ranking methods, we rank all ICD-9 codes in the hierarchy (001-999 and V01-V91), excluding the 3 codes of 348.82, 761.6, and 798 that could cause leakage of the label. We then take the highest 551 codes (because CCI, DCSI, and Elix have 551 codes in total) and partition them into groups of the same size as the categories used by Anand et al. (2018).

As mentioned in the related work, CCI, DCSI, and Elix are based on points. Most categories are 1 point (meaning that you add 1 to the category if it exists in the patients' admission), while others are 2, 3, or 6 points. At the end, each score is calculated by the sum of its categories.

When constructing the three scores (CCI, DCSI, and Elix), each feature ranking method selects the same number of 6 point, 3 point, 2 point, and 1 point categories as Anand et al. (2018). This is to ensure a fair comparison between our scores and CCI, DCSI, and Elix. To assign a new set of codes into these categories, the categories are assigned an expected value based on their weight and the number of codes within that category. For example, the Hypothyroidism category for Elix has 5 codes and a point value of 1, so an expected point value of $1/5$ is assigned. Our partitioning selects the features with the highest point values and assigns them to the most heavily weighted categories (first by points then by expected points). We call this experiment All.

4.3.3. Top-K

In the Top-K experiment, we vary the number of features to use for classification. We aim to evaluate how the number of variables influences accuracy, and thus no longer restrict the ICD-9 codes to specific categories. Using all feature ranking methods including HIC, we select the top K ICD-9 codes where K is 5, 10, 50, 100, 200, 500, 1000, 5000, or 10000. For each N and each feature ranking method, we run mortality prediction for patients with diabetes using those top K ICD-9 codes as features for a random forest model based on Anand et al. (2018). We use five-fold cross validation of Random Forest with 5000 trees, a max of 5 observations in leaf nodes, with a data split of 70% for training and 30% for testing. Unlike Anand et al. (2018), we modify both the number of features randomly selected at each node and a max tree depth to $\sqrt{K}$ (as Anand et al. (2018) used 3 and 10 respectively, which are too small).

4.4. Mortality prediction

We also benchmark our work against that of Fu et al. (2019), who did mortality prediction for all patients in MIMIC-III. That work used Deep Dictionary learning with 283 features, which correspond to categories outlined in CCS (Elixhauser et al., 2015). CCS groups all ICD-9 codes into 283 categories. Based on the approach by Fu et al. (2019), we randomly split MIMIC-III into 70% train, 10% validation, and 20% test, with the label of in-hospital mortality. We run the experiment 5 separate times and present the average over the 5 runs. We construct a different version of CCS using each feature ranking method. For a fair comparison of the normal CCS score to our modified CCS scores, we use the Restricted method (looking at branches only) so that the class of codes in each category in our modified score is the same as the original CCS score. We only include the top-5 features for each category to compare our HIC feature ranking algorithm to other feature ranking algorithms. We limit each category to 5 ICD-9 codes to demonstrate that HIC chooses the best features even with fewer codes than CCS. In total, we have 283 features (same as Fu et al. (2019)). Based on the formula in section 3.4, w_b and w_t are 0.262 and 0.738 respectively.

Model	Anand		Anand Recreate		HIC All	
Threshold	Sens	Spec	Sens	Spec	Sens	Spec
0.04	0.81	0.61	0.90	0.46	**0.92**	**0.76**
0.06	0.70	0.74	0.72	0.66	**0.85**	**0.81**
0.08	0.59	0.81	0.54	0.75	**0.71**	**0.87**
0.10	0.53	0.85	0.49	0.83	**0.67**	**0.93**
0.12	0.41	0.89	0.38	0.88	**0.61**	**0.95**
0.14	0.38	0.92	0.34	0.95	**0.56**	**0.97**
0.16	0.35	0.94	0.30	0.96	**0.40**	**0.98**
Average	0.54	0.82	0.52	0.78	**0.67**	**0.90**

Table 1: Comparing sensitivity (sens) and specificity (spec) of mortality prediction on the diabetes cohort using Random Forest done by Anand et al. (2018), to our recreation (Anand Recreate), to HIC All. For each threshold, the optimal sensitivity and specificity is bolded.

4.5. Ablation study

Finally, we run an ablation study designed to (i) investigate how each component of HIC influences the accuracy and (ii) determine how each pair of terms interacts. Because our work builds on MI, we always include it as a term. The first set of ablation experiments (called MI + term) consists of 2 terms: MI and each of the other terms of HIC. The second set of ablation experiments (called MI + 2-terms) consists of 3 terms: MI and each pair the other terms of HIC. The last set of ablation experiments (called HIC - term) consists of 4 terms: MI and three of the other terms of HIC (excluding each term that is not MI). In total, there are 14 different experiments.

5. Results

5.1. Mortality prediction for individuals with diabetes

Table 1 shows the comparison of results reported in Anand et al. (2018) and Anand Recreate, to HIC and for various thresholds. When comparing HIC to Anand Recreate, we have an average increase of 0.15 sensitivity and 0.11 specificity. When comparing HIC to Anand et al. (2018), we have an average increase of 0.14 sensitivity and 0.07 specificity. Using our HIC method, we get a higher sensitivity and specificity for all thresholds of the Random Forest.

Table 2 compares results reported in Anand et al. (2018) and Anand Recreate, on the Restricted and All tasks against HIC and baseline algorithms. HIC All outperformed the other baseline methods by at least an AUROC of 0.059.

When comparing the standard Elix, CCI, and DCSI to the ICD-9 codes chosen by HIC Restricted as shown in Figure 2, we can see that the majority of level 3 codes stayed the same. Most of the level 2 codes stayed the same, while some moved down to level 3. Similarly, most of the level 1 codes moved down to level 3, while some changed to level 2. Overall, 333 out of 551 codes stayed the same, 178 moved down, and only 40 moved up. Therefore, the level 1 codes in the original scores were too general for classification, and accuracy improves when using the same code, but at a finer

Algorithm	AUROC
HIC Restricted	**0.844 ± 0.005**
HIC All	**0.895 ± 0.004**
Anand et al. (2018)	0.787
Anand Recreate	0.753 ± 0.035
MI Restricted	0.814 ± 0.006
MI All	0.822 ± 0.003
SVM-RFE Restricted	0.790 ± 0.008
SVM-RFE All	0.809 ± 0.005
RF Gini Restricted	0.778 ± 0.007
RF Gini All	0.786 ± 0.005
ReliefF Restricted	0.808 ± 0.005
ReliefF All	0.820 ± 0.004
SURF Restricted	0.815 ± 0.005
SURF All	0.828 ± 0.003
MultiSURF Restricted	0.822 ± 0.006
MultiSURF All	0.836 ± 0.002

Table 2: Averaged AUROC of mortality prediction on the diabetes cohort of MIMIC-III (with standard deviation). Anand et al. (2018) did not report standard deviation. The highest accuracy per experiment is bolded.

granularity (e.g. 250 vs 250.00). Accuracy improves because the lower level codes are more relevant to the specific problem.

One of the reasons that HIC outperformed all of the other baselines on the Restricted task is because it not only has the advantages of MI, but also overcomes some limitations by incorporating statistical significance and as well as the sample size. An example is in the CCI score for “Acute myocardial infarction", ICD-9 code 410. Most feature ranking methods (e.g. mutual information, SVM-RFE, RF-Gini, and ReliefF) chose ICD-9 code 410.61 which had a probability of death of $1/1 = 1$, whereas HIC took into account the sample size as well as the other terms and selected the root node of ICD-9 code 410, which had a probability of death of $54/544 = 0.099$. When only using the probability of death, MI would always choose the probability of death of 1 rather than 0.099, and not take into account the denominator (number of people). HIC suggests that even though 1 >> 0.099, we should use the feature with the most support and higher sample size. Overall, the feature ranking methods performed better on the All task than the Restricted one, as it allowed methods to identify more informative codes beyond the restricted branches.

Figure 3 shows the similarity between the original codes and HIC Restricted and HIC All. As we can see, HIC Restricted kept 60% of the original codes, while HIC All kept around 13%. Therefore, the classification could be improved if more relevant codes were chosen based on incorporating the level and evaluating ICD-9 codes as a hierarchy.

Out of the top 5 ICD-9 codes according to HIC All (250.1, 518.8, 530.81, 285.9, and 518.81), only 250.1 appears in Elix, CCI, or DCSI. The other codes (or their ancestors or children) do not appear in the original scores. This shows that the original scores of Elix, CCI, or DCSI might not be enough to best predict mortality and other codes should be taken into account.

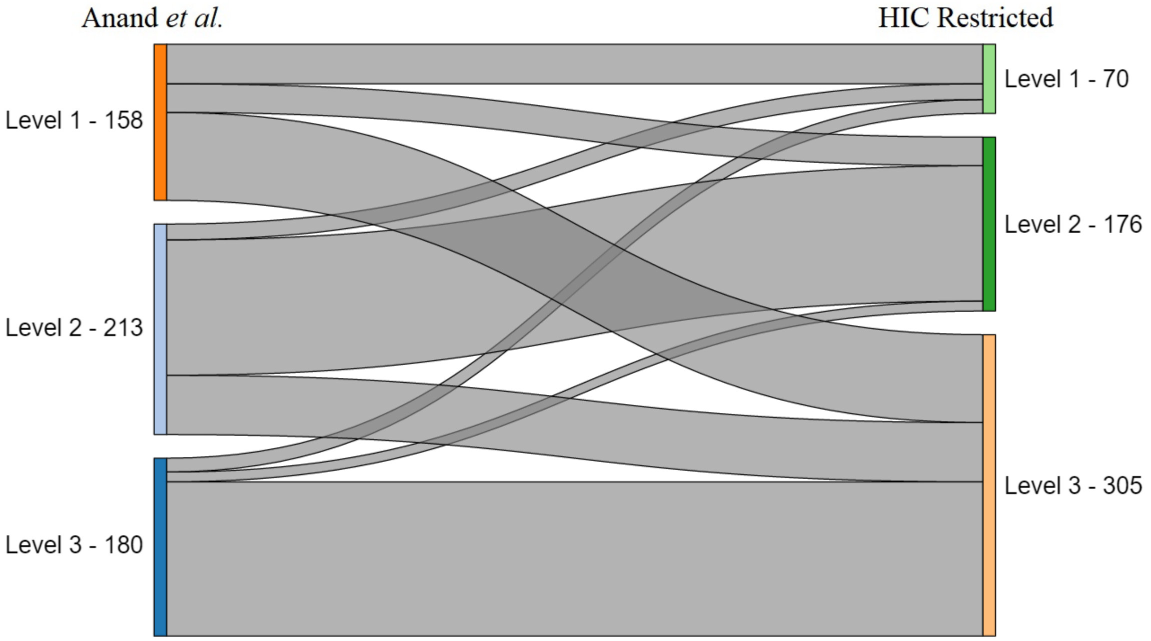


Figure 2: The Sankey diagram illustrates the movement of ICD-9 codes between Anand et al. (2018) (left) and HIC Restricted (right).

Recalling the Top-K experiment, we rank every ICD-9 code using HIC and select the top K ICD-9 codes for diabetes mortality prediction along with the 8 non-ICD-9 code based features. Analyzing the experimental results in Figure 4, the AUROC of HIC increases until $N = 500$ and then plateaus. The AUROC of HIC at $N = 500$ is 0.920, which shows that selecting each ICD-9 code feature individually based on HIC performs better than grouping them into scores. This AUROC of 0.920 outperforms all baseline feature ranking methods, even when they use 10000 features, because HIC selected ICD-9 codes at the correct level of granularity. For example, while MI, RF-Gini, SVM-RFE, ReliefF, and SURF chose the root node ICD-9 code 518, HIC selected specifically ICD-9 code 518.8. Although fewer people had ICD-9 code 518.8 than ICD-9 code 518, the predictive power of ICD-9 code 518.8 is better than ICD-9 code 518, leading to an increase in AUROC. This happened when K=10 for MI, RF-Gini, and SVM-RFE, and when K=50 for ReliefF and SURF. Because of the nature of the Top-K experiment, this selection by HIC and sub-optimal selection by the other feature ranking methods propagated to the larger values of K, leading HIC to outperform all of them.

5.2. Mortality prediction

As we can see from Table 3, HIC outperforms all other feature ranking methods as well as the original features used in Deep Dictionary Learning (DDL) (Fu et al., 2019). Every feature ranking methods selects at least 100 codes at each level of the hierarchy. Out of the 1358 codes chosen by HIC for CCS (Elixhauser et al., 2015), 880 codes changed: 562 became more specific and 318 changed to a

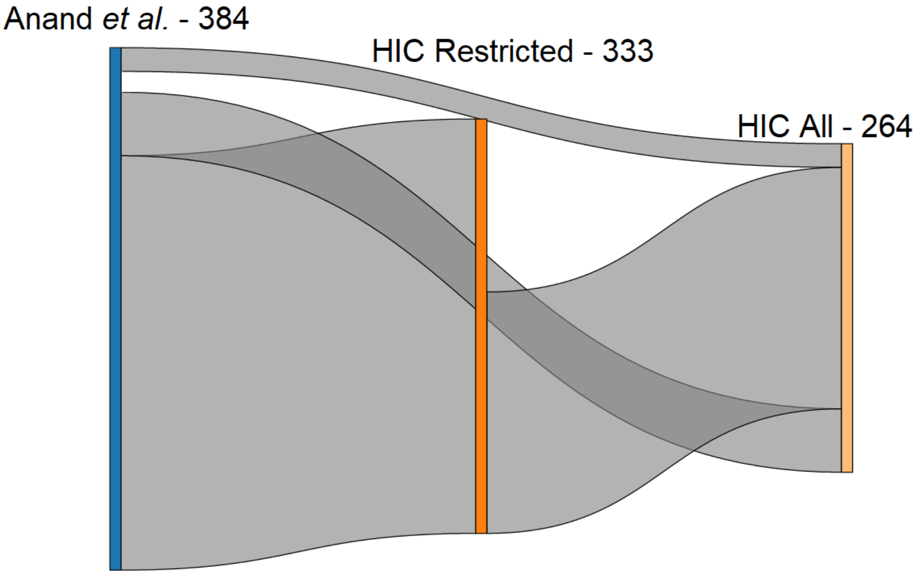


Figure 3: The Sankey diagram illustrates the number of codes in common between Anand et al. (2018), HIC Restricted, and HIC All experiments. The line in the middle shows 51 codes in common for all 3 experiments and the line on top shows the 19 codes in common between Anand et al. (2018) and HIC All.

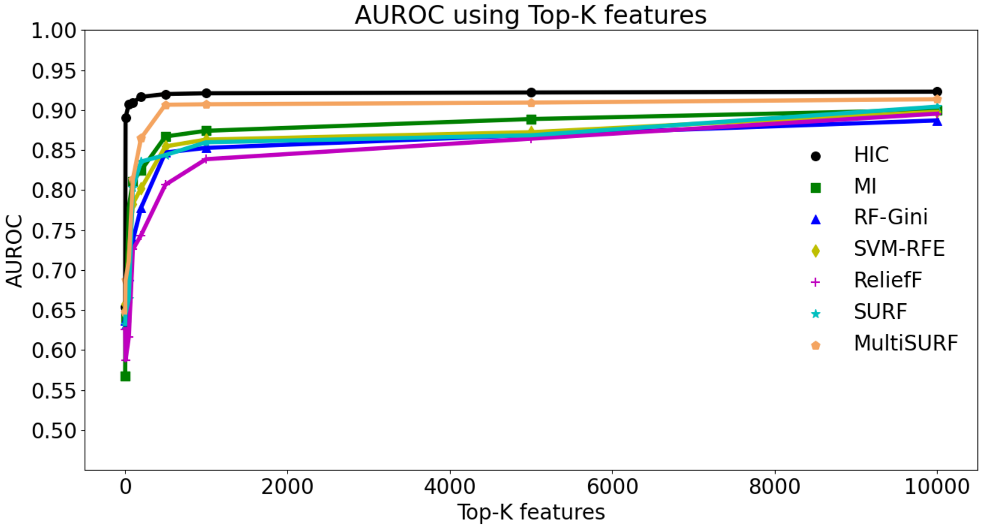


Figure 4: AUROC using the Top-K features of all feature ranking methods.

higher level of the hierarchy. For the other feature ranking methods, an average of 490 changed: 256 down (more specific) and 234 (less specific). Similarly to the AUROC for mortality prediction on

Algorithm	AUROC
HIC	**0.887 ± 0.004**
DDL (Fu et al., 2019)	0.819 ± 0.004
MI	0.799 ± 0.003
SVM-RFE	0.726 ± 0.006
RF Gini	0.754 ± 0.006
ReliefF	0.776 ± 0.005
SURF	0.782 ± 0.003
MultiSURF	0.808 ± 0.003

Table 3: Average AUROC of mortality prediction on all of the patients in MIMIC-III (with standard deviation). The highest accuracy is bolded.

	MI	lvl	Branch	Tree	Number	AUROC
MI + term	✓	✓				**0.845**
	✓		✓			0.830
	✓			✓		0.834
	✓				✓	0.841
MI + 2-terms	✓	✓	✓			0.853
	✓	✓		✓		0.854
	✓	✓			✓	**0.863**
	✓		✓	✓		0.851
	✓		✓		✓	0.853
	✓			✓	✓	0.855
HIC - term	✓	✓	✓	✓		0.864
	✓	✓	✓		✓	0.878
	✓	✓		✓	✓	**0.880**
	✓		✓	✓	✓	0.865

Table 4: Ablation study results when using Mutual Information (MI) and varying the other terms: level + max_level (lvl), branch statistical significance (Branch), tree statistical significance (Tree), and number of participants (Number). The highest AUROC per experiment is bolded.

individuals with diabetes, the reason HIC All outperformed the other baseline methods by at least an AUROC of 0.068 is because of the two terms based on MI and the level. Further, when analyzing the ablation study in Table 4, we can see that the most important terms are the level + max_level and the number of people per ICD-9 code as these terms led to the largest increase when compared to MI (MI + term), and the largest decrease when excluded and compared to HIC. When comparing HIC to the other feature ranking baselines, there were 400 ICD-9 codes that were selected by all methods. Furthermore, HIC selected 252 codes that were not selected by any other baseline method.

Similar to the Restricted example from Anand et al. (2018), most feature ranking methods (e.g. mutual information, SVM-RFE, ReliefF, and SURF) selected ICD-9 code 410.61 ($4/26 = 0.154$) in category 100 of CCS, whereas HIC selected the root node 410 ($650/3522 = 0.185$). Taking into account sample size and the hierarchy allows HIC to select the most important features.

6. Conclusion

We propose HIC, a novel approach for hierarchical variable abstraction. Our approach allows fully automated selection of not only what variables are most informative for a task, but also selects the right task-specific level of granularity for each. In experiments on the MIMIC-III ICU dataset our approach outperforms prior work on mortality prediction, as well as other feature ranking baselines. By combining terms that consider the predictive value of a feature, the amount of data associated with a node in the hierarchy (i.e. number of patients), and tradeoffs with other codes (i.e. up or down a level in a branch of the tree) HIC is robust against outlying values such as features with very high predictive value that have a small sample size. Further, our experiments show that using this approach on the full ICD-9 hierarchy with no restrictions on which or how many codes can be included leads to higher AUROC than feature engineering approaches that categorize ICD-9 codes and assign them weights. In future work we aim to expand this approach to other medical ontologies. We hope our results encourage more research on hierarchical feature ranking in the context of healthcare.

Limitations We assume the hierarchical features and labels are binary. While data for other tasks besides mortality prediction can be binarized, in future work we plan to extend the HIC method to handle continuous variables and non-binary labels.

Acknowledgements

This work was supported in part by the NLM of the NIH under Award Number R01LM013308.

References

R. S. Anand, P. Stey, S. S. Jain, D. R. Biron, H Bhatt, K. A Monteiro, E. Feller, M. L. Ranney, I. Neil Sarkar, and E. S. Chen. Predicting mortality in diabetic ICU patients using machine learning and severity indices. *AMIA Summits on Translational Science Proceedings*, 2018:310 – 319, 2018.

R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE transactions on neural networks*, 5 4:537–50, 1994.

A. Burks, C. Shannon, and W. Weaver. The mathematical theory of communication. *The Philosophical Review*, 60:398, 1951.

E. Choi, M. T. Bahadori, L. Song, W. Stewart, and J. Sun. Gram: Graph-based attention model for healthcare representation learning. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

E. Choi, C. Xiao, W. Stewart, and J. Sun. Mime: Multilevel medical embedding of electronic health records for predictive healthcare. *ArXiv*, abs/1810.09593, 2018.

E. Choi, Z. Xu, Y. Li, M. W. Dusenberry, G. Flores, Y. Xue, and A. M. Dai. Learning the graphical structure of electronic health records with graph convolutional transformer. In *AAAI*, 2020.

A. Elixhauser, C. Steiner, D. R. Harris, and R. Coffey. Comorbidity measures for use with administrative data. *Medical care*, 36 1:8–27, 1998.

A. Elixhauser, C. Steiner, and L. Palmer. *Clinical Classifications Software (CCS)*. U.S. Agency for Healthcare Research and Quality, http://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp, 2015.

T. Fu, T. N. Hoang, C. Xiao, and J. Sun. DDL: Deep dictionary learning for predictive phenotyping. In *IJCAI*, 2019.

C. Greene, N. Penrod, J. Kiralis, and J. Moore. Spatially uniform relieff (SURF) for computationally-efficient filtering of gene-gene interactions. *BioData Mining*, 2:5 – 5, 2009.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2004.

M. Huang, Y. Hung, W. Lee, R. Li, and B. Jiang. SVM-RFE based feature selection and taguchi parameters optimization for multiclass svm classifier. *The Scientific World Journal*, 2014.

A. E.W. Johnson, T. J. Pollard, L. Shen, L. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3, 2016. doi: 10.1038/sdata.2016.35.

K. Kira and L. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, 1992.

I. Kononenko. Estimating attributes: Analysis and extensions of relief. In *ECML*, 1994.

C. Mao, L. Yao, and Y. Luo. Medgcn: Graph convolutional networks for multiple medical tasks. *ArXiv*, abs/1904.00326, 2019.

B. Menze, B. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich, and F. Hamprecht. A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC Bioinformatics*, 10:213 – 213, 2009.

R. Pivovarov and N. Elhadad. A hybrid knowledge-based and data-driven approach to identifying semantically similar concepts. *Journal of biomedical informatics*, 45 3:471–481, 2012.

H. Quan, V. Sundararajan, P. Halfon, A. Fong, B. Burnand, J. Luthi, L. D. Saunders, C. Beck, T. Feasby, and W. Ghali. Coding algorithms for defining comorbidities in ICD-9-CM and ICD-10 administrative data. *Medical Care*, 43:1130–1139, 2005.

A. Singh, G. Nadkarni, J. Guttag, and E. Bottinger. Leveraging hierarchy in medical codes for predictive modeling. *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2014.

R. Urbanowicz, R. S. Olson, P. Schmitt, M. Meeker, and J. Moore. Benchmarking relief-based feature selection methods for bioinformatics data mining. *Journal of biomedical informatics*, 85: 168–188, 2018.

C. Wan and A. Freitas. Prediction of the pro-longevity or anti-longevity effect of caenorhabditis elegans genes based on bayesian classification methods. In *2013 IEEE International Conference on Bioinformatics and Biomedicine*, pages 373–380, 2013. doi: 10.1109/BIBM.2013.6732521.

C. Wan, A. A. Freitas, and J. P. de Magalhães. Predicting the pro-longevity or anti-longevity effect of model organism genes with new hierarchical feature selection methods. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 12(2):262–275, 2015. doi: 10.1109/TCBB.2014.2355218.

B. Young, E. Lin, M. V. Von Korff, G. Simon, P. Ciechanowski, E. Ludman, S. Everson-Stewart, L. Kinder, M. Oliver, E. Boyko, and W. Katon. Diabetes complications severity index and risk of mortality, hospitalization, and healthcare utilization. *The American journal of managed care*, 14 1:15–23, 2008.