Selecting Relevant Features from the Electronic Health Record for Clinical Code Prediction

Elyne Scheurwegs[a,b,*], Boris Cule[a], Kim Luyckx[c], Léon Luyten[d], Walter Daelemans[b]

[a] *University of Antwerp, Advanced Database Research and Modelling research group (ADReM) Middelheimlaan 1, B-2020 Antwerp, Belgium*
[b] *University of Antwerp, Computational Linguistics and Psycholinguistics (CLiPS) research center Lange Winkelstraat 40-42, B-2000 Antwerp, Belgium*
[c] *Antwerp University Hospital, ICT department Wilrijkstraat 10, B-2650 Edegem, Belgium*
[d] *Antwerp University Hospital, Medical information department Wilrijkstraat 10, B-2650 Edegem, Belgium*

Abstract

A multitude of information sources is present in the electronic health record (EHR), each of which can contain clues to automatically assign diagnosis and procedure codes. These sources however show information overlap and quality differences, which complicates the retrieval of these clues. Through feature selection, a denser representation with a consistent quality and less information overlap can be obtained. We introduce and compare coverage-based feature selection methods, based on confidence and information gain. These approaches were evaluated over a range of medical specialties, with seven different medical specialties for ICD-9-CM code prediction (six at the Antwerp University Hospital and one in the MIMIC-III dataset) and two different medical specialties for ICD-10-CM code prediction. Using confidence coverage to integrate all sources in an EHR shows a consistent improvement in F-measure (49.83% for diagnosis codes on average), both compared with the baseline (44.25% for diagnosis codes on average) and with using the best standalone source (44.41% for diagnosis codes on average). Confidence coverage creates a concise patient stay representation independent of a rigid framework such as UMLS, and contains easily interpretable features. Confidence coverage has several advantages to a baseline setup. In our baseline setup, feature selection was limited to a filter removing features with less than five total occurrences in the trainingset. Prediction results improved consistently when using multiple heterogeneous sources to predict clinical codes, while reducing the number of features and the processing time.

Keywords: Feature selection, Data integration, EHR mining, Clinical coding, Data representation

1. Introduction

Electronic health records (EHRs) contain different types of information about patients and their stays in health facilities [1]. Clinical codes reflect diagnoses and procedures related to a patient stay and are primarily assigned for reporting and reimbursement purposes. Their widespread adoption in hospitals makes them a viable information source in research and monitoring applications [2, 3].

Clinical codes are predefined in a classification system (e.g., ICD [4], ICPC [5]) and are assigned to a patient stay by clinical coders who analyse the patient's medical information. With the recent transition to ICD-10-CM/PCS, the coding complexity grew since there are up to nineteen times as many procedure codes and five times as many diagnosis codes. This increased complexity requires better techniques to assist clinical coders. Existing (proprietary) applications either focus on making the code system easily browsable or offer computer-assisted coding. Most of those applications operate on English data.

Computer-assisted coding helps clinical coders by

*Corresponding author
Email address: elyne.scheurwegs@uantwerpen.be (Elyne Scheurwegs)

pointing out relevant information, suggesting codes, or in simple cases automatically assigning codes without manual review [6]. Most systems described by Stanfill et al. are used in controlled settings (e.g., assuming a very specific notation format in medical data), predict only a limited set of codes, and are mainly based on information from discharge summaries or radiology reports [7]. While some of these approaches perform well, they are difficult to scale to larger datasets or port to different environments. Recent research moved on to using real-world data such as the MIMIC-III dataset [8]. In this paper, we propose several feature selection methods to assist an automatic coding algorithm, trained on multiple data sources. These methods are used to reduce redundancy between features extracted from different information sources and thus create a denser representation of the information with minimal loss of quality.

1.1. Background

Recently, two techniques were proposed to leverage the sparsity of the output codes with layered prediction and to make classification on a complex, real-world dataset more effective [9, 10]. For evaluation of new techniques, the MIMIC dataset is often used as a benchmark [8]. Perotte et al. predict diagnosis codes for MIMIC-II by using the code hierarchy to learn codes incrementally: they initially predict higher-level diagnosis categories (the first n digits of the code) and then predict the complete codes [9]. Subotin et al. predict ICD-10-PCS codes from a large set of discharge summaries by predicting parts of the code, since digits in ICD-10-PCS point to different properties of the procedure [10].

These approaches still depend on the availability of discharge summaries keeping a uniform, descriptive, and complete notation. However, information is often missing from discharge summaries and most hospitals (and clinicians) have their own formatting and writing style. By supplementing the information found in discharge summaries with information found in other data sources, the missing information can be compensated for. This technique is already being used for clinical tasks such as identifying patient cohorts [11] and high-throughput phenotyping for patient cohort identification [3, 12].

Pathak et al. mapped structured and unstructured data onto standardized thesauri (e.g., UMLS), resulting in a unified data view [12, 13]. While this approach can be powerful, it also creates a strong dependency on the ontologies used and the completeness of the mapping. Mapping local ontologies (e.g., RIZIV [14]) or terms found in (Dutch) clinical notes, requires substantial research effort. Scheurwegs et al. performed early and late data integration with structured sources directly represented as features and unstructured sources converted into features through a bag-of-words representation [15]. Due to the difference in feature information density between sources - a bag-of-words representation has more and weaker features than a representation of structured sources - an early data integration approach did not work. Feeding the predictions per data source into a meta classifier proved to be a better approach. However, due to the compression of each source to a single data point for each class, a lot of information was lost.

In a recent article, a deep neural network, consisting of stacked autoencoders, was created to represent patients as a dense vector (called 'deep patient') [16]. The structured and unstructured sources used were generalized during preprocessing (e.g., notes are represented in 300 generalized categories, using topic modeling [17]). The resulting representation is claimed to be applicable to a variety of medical applications. Rather than transforming all data into a dense representation, this paper reports on research where the most interesting features are selected. This influences the usability of the algorithm: we optimise for the clinical coding task, whereas 'deep patient' will generate a general-purpose representation. We expect our approach to focus on specific information with strong correlations to a particular clinical code, while capturing the information that may have been removed from the dense vector representation.

Apart from getting complementary information from different sources and balancing the informativeness, a dense representation, created by either a feature selection algorithm or an algorithm that creates an entirely new feature space, can make it easier to deal with sources that are more noisy for some specialties when compared to a late data integration approach [15]. The latter would disregard the source entirely, while a dense representation would still be able to retain a few features originating from that source. Since the quality and amount of information found in certain sources differs for all specialties [15], an approach that learns from interesting features from each source without knowing the most informative sources beforehand makes the algorithm generalisable over multiple datasets.

Feature selection algorithms often improve algorithm performance, particularly in situations where training data is limited [18, 19]. In classification, successful feature selection approaches range from techniques optimizing the goodness-of-fit towards a class by ranking features to techniques that specifically look for a minimal set of features by reducing inner-feature redundancies [20]. In a multi-label problem, we have a multitude of labels for which we want to optimize goodness-of-fit, while we also suffer from redundancies in our feature set. Ideally, we want to optimize both goodness-of-fit and inner-feature redundancy, as the goodness-of-fit can help us determine which of the strongly correlated features optimally represents the task at hand, while solely using goodness-of-fit leads to the selection of features that might be strongly correlated and do not contribute to the final prediction [21].

To tackle both, techniques such as mRMR [22] have been introduced, but computationally they do not perform well on datasets with a large number of features and a large number of classes, due to a pairwise feature-based redundancy calculation, and a separate calculation of these metrics for each label. Other techniques used for feature selection, such as markov-blanket algorithms [23, 24], yield good results for feature selection, at the cost of computational efficiency, as they require the induction of a (partial) Bayesian network.

Database coverage-based algorithms gain a computational advantage over pairwise feature-based methods that compare redundancy, since the former only require one comparison per feature with the joint coverage of all previously selected features. This does require the usage of a ranking of features, which is not always required for algorithms that identify redundant features in a pairwise fashion.

Using the entropy of a feature on the entire dataset on a scoring mechanism for feature relevance can also be problematic due to the influence of both sparse features and classes in EHRs. We investigate these issues by comparing techniques using different measures for the informativeness of a feature and by using a technique that considers redundancy of features based on the dataset coverage of all previously selected features for a certain class instead of doing a pairwise comparison between features. We monitored the influence this has on process time as well.

1.2. Significance

In this paper, we show that complementary information can be extracted from multiple sources efficiently by selecting the most reliable information for predicting diagnosis and procedure codes in an early data integration approach. We introduce confidence-coverage as a feature selection method that uses a co-occurrence based scoring mechanism, combined with an instance-oriented selection designed to reduce information overlap between different extracted features. We evaluate performance of four feature selection methods on several datasets of the Antwerp University Hospital (UZA), six of which have ICD-9-CM (procedural and diagnosis) codes and two of which have ICD-10-CM and ICD-10-PCS codes assigned. For benchmarking purposes, we also evaluate on the MIMIC-III dataset.

2. Materials and Methods

To prepare the raw data, present in both textual and structured sources, a representation of each source is created. These representations are then presented to four different feature selection algorithms. The four different representations that result after feature selection, and the original representation, are then presented to the classifiers and evaluated in the experimental section. The original representation serves as a baseline, and will show the effect of applying any feature selection algorithms.

The prediction of clinical codes is cast as a multi-label classification task, in which each clinical code is treated as a separate class. Classes are not mutually exclusive. Diagnosis and procedure codes are treated as different classification tasks. Each distinct classifier in the setup makes a decision based on the features selected for all codes, not only for the code in focus.

2.1. Datasets

We will use three anonymized clinical datasets consisting of a collection of patient stays, with multiple associated structured and unstructured data sources. Each dataset is divided into medical specialties (see Table 1). The clinical codes show a Zipfian distribution: a small number of codes occurs in a large number of patient stays, whereas most codes are sparse.

Both UZA1 and UZA2 were extracted from the UZA data warehouse. UZA1 covers a two-year period, and is divided into six medical specialties. It

Table 1: Overview of the datasets. The table shows the number of patient stays, the number of unique diagnosis and procedure codes occurring in each medical specialty and the label cardinality, which represents the average number of labels per patient stay.

Dataset	**Medical specialty**	**Number of patient stays**	**Procedure codes**		**Diagnosis codes**	
			Unique codes	**Label cardinality**	**Unique codes**	**Label cardinality**
UZA1	Cardiology	10000	235	3.04	2148	7.90
UZA1	Oncology	10000	156	1.01	1696	12.74
UZA1	Urology	3440	282	1.10	1422	5.83
UZA1	Gastroenterology	7440	232	0.87	2165	4.87
UZA1	Ophthalmology	4510	187	1.75	1136	3.31
UZA1	Pneumology	3430	151	0.81	1884	9.69
UZA2	Cardiology	1680	169	2.41	987	5.96
UZA2	Oncology	2920	110	1.07	824	8.94
MIMIC-III	Intensive Care	10000	1367	3.91	1769	10.59

contains ICD-9-CM procedure and diagnosis codes. UZA2 covers a three-month period, and contains stays of two medical specialties. Diagnosis codes are represented as ICD-10-CM codes and procedure codes as ICD-10-PCS codes. Both datasets contain structured and unstructured data (in Dutch). The structured sources are lab results (local code), inpatient medication prescriptions (ATC), oncological pathology codes (CODAP), medical departments where patients were seen or treated, demographics (year of birth and gender), and RIZIV procedure codes [14]. RIZIV codes are a local coding system referring to medical procedures or interventions. Unstructured sources include discharge summaries, letters, radiology reports, notes, surgery reports, attestations, and requests.

MIMIC-III is a public dataset about intensive care in which ICD-9-CM diagnosis and procedure codes are present [4, 8]. Structured sources extracted from MIMIC-III are NDC codes (inpatient medication prescriptions), LOINC codes (representing lab tests), and departmental information. All available unstructured data (including discharge summaries) is included as well. We extracted a random subset of 10,000 stays for our experiments.

All three datasets are extractions of real-world data. The ICD-9-CM datasets are representative in both size and complexity of a typical medical specialty of a hospital, while the ICD-10-CM/PCS dataset is still limited in size and used for early experiments in ICD-10 code assignment.

2.2. Data representation

Structured sources, that are used as input, are represented by the number of occurrences of a certain code. For example, the feature 'ATC:N02BE01=2' represents the ATC-code 'N02BE01' occuring twice in a patient stay. For lab tests, we use value-unit pairs as features and calculate derived measures, such as deviation from the mean, maximum and minimum values, and whether a predefined limit has been passed (e.g., the upper limit of normal or ULN). Time dependencies in laboratory values were represented using averaged trends over time, the variability in results over time, and the minimum and maximum value.

Textual sources are represented as medical multiword expressions (mMWEs), consisting of a lemmatized string (i.e., dictionary form). For example, from the sentence fragment 'suffering from an ischemic stroke', '*ischemic_stroke*' is extracted as a mMWE. To featurize the mMWEs, two different methods provide a different representation of all texts. For the first representation, we applied an IDF-weighted fuzzy matching dictionary-based approach using several ontologies (e.g., UMLS, 3BT, medication brand names, ...) as underlying ontologies. The second representation consists of an unsupervised approach based on linguistic pattern matching and mutual information (LMI) [25].

The latter technique uses the differences between an unannotated clinical corpus and other corpora (SoNaR subcorpora, which consists of e.g., newspaper articles, wikipedia, emails) to extract clinically relevant text, by weighing word frequency against each of the other corpora. A word is considered medical if it is more related to the medical corpus. The frequency of candidates consisting of multiple words are averaged, in which the average frequency has to be higher for the clinical corpus. By using two independent techniques to represent texts, we are able to account for terms that do not have a formal definition in a lexicon and are medically relevant (e.g., synonyms that are not defined in UMLS), while we can also make use of the (limited) set of Dutch term definitions that we do have.

The representation of texts as a list of mMWEs does not use negation detection, co-reference resolution, or temporality detection. Despite interesting results in other domains and languages, we have found that these components do not adequately perform on the Dutch clinical texts in our datasets. These clinical texts are not well formed, often lack sentence structures and contain a lot of non-standard terminology that can indicate negation or uncertainty (such as hedging). Preliminary experiments showed that adding them to our pipeline did not lead to an improvement in the results.

Diagnostic and procedural codes, which are used as output classes, are binarized: if a code is assigned at least once during the patient stay, it is considered present, otherwise, it is considered absent.

2.3. Feature selection

All data sources provide information we can consider features for predicting diagnosis or procedure codes. However, the data sources vary significantly in terms of the number of features and feature quality. To balance the information content between features, remove noise, and ultimately select the most important features, we propose four methods, one of which is a gain ratio baseline. A frequency filter is applied before all feature selection methods, which removes all features that do not occur at least five times in the training set. This frequency filter is used for the baseline, consisting of the original representation, as well.

2.3.1. Gain ratio(GainRatio)

Gain ratio is a normalized version of information gain (IG), in which a feature's IG is normalized by the number of different values of that feature in the dataset. This counteracts the IG bias towards features with a large number of different values. Gain Ratio is used as a reference method in this paper.

The Shannon entropy of dataset D can be defined as:

$$H(D) = -\sum_{i=1}^{|C|} P(c_i) log_2 P(c_i)$$

where $|C|$ represents the total number of classes and $P(c_i)$ represents the probability mass of the ith class in the dataset D. In a multi-label environment, we consider a code being present and a code not being present as the classes.

The information gain of a feature F is defined as the ratio of Shannon entropy between all data instances and the weighted average entropy of the subsets of instances where the value of the feature for a given instance f is equal to v for each unique value v in all instances for feature F:

$$IG(f) = H(D) - \sum_{v \in vals(f)} \frac{|\{d \in D | val(d,f) = v\}|}{|D|} \cdot H(\{d \in D | val(d,f) = v\}) \quad (1)$$

2.3.2. Confidence coverage (ConfCov)

Confidence coverage models the direct correlation between a feature and a class without penalizing it for low frequency and attempts to select features that cover all training instances. The confidence coverage method originates from a pruning algorithm used in rule-generating classifiers such as IREP, RIPPER, and CMAR [26, 27, 28]. In IREP and RIPPER, rules are pruned incrementally during learning, while CMAR prunes after learning. In CMAR, rules are based on frequently co-occurring elements in instances by considering their relation with classes assigned to those same instances.

Confidence is a metric that represents how specific a certain feature is to a class. The confidence of a (binarized) feature f for a class c_i in dataset D can be defined as the ratio between the number of instances where f and c_i are present and the number of instances where f is present:

$$conf(f) = \frac{|\{d \in D | hasFeature(d,f) \& hasClass(d,c_i)\}|}{|\{d \in D | hasFeature(d,f)\}|}$$

These feature-class associations, which can be de-

fined as rules, are then ordered by their confidence score and pruned using their database coverage. The database coverage of a rule is calculated by looking at the specific training instances that are covered by a certain rule. This allows for the removal of rules that are covered by higher ranking rules, while rules that cover previously uncovered instances are still preserved. In figure 1, an example is shown. Rule A has a confidence of 0.82 and covers instances 1, 2, and 3. Rule B, with a confidence of 0.75, covers instances 1 and 2. xRule C has a confidence of 0.56 and covers instances 3 and 4. When the two most confident rules are selected, instance 4 is not covered, whereas the coverage-based method covers all instances by selecting only rules A and C.

To make the algorithm less strict, we allow for some degree of overlap between rules and prune rules with low information content. An instance is removed from the coverage list once it has been covered by three different rules. A rule is preserved when it covers at least 3 instances in the coverage list. A minimum threshold of 10% on confidence is imposed.

When assigning clinical codes based on a multitude of sources, a similar problem as with rule generation arises: an input space with a large number of (potentially noisy) features. By associating the (binarized) feature to the different clinical codes (classes), the feature-class associations can be considered rules. Confidence coverage then addresses the reduction of the substantial number of generated rules by ordering them based on confidence and then pruning them based on their coverage of the dataset. In our setup, we binarize the features before associating them with classes in confidence-coverage. Binarization captures most of the information from binary and count-based features (which are the majority in our setup), but does not lend itself as well for features describing trends and values in lab tests, since it essentially reduces these features to whether or not a trend has been calculated for that specific patient, and thus allows the feature to only be selected when calculating a trend is indicative for a certain class.

Feature-class associations are not used as rules, as in CMAR and RIPPER, but rather as indicators for feature selection performed prior to classification. That enables us to recover the original feature representation (i.e., binary, counts, and values) instead of the binary format used by the confidence coverage algorithm.

2.3.3. Information Gain Coverage (InfGainCov)

As a second method, we adapted the confidence coverage method described above to work with information gain as a scoring mechanism. As described in section 2.3.1, information gain uses Shannon entropy, which is calculated using the feature's correlation with both the positive class (i.e., the clinical code is present) and the negative class (i.e., the clinical code is not present). Since the coverage used in confidence-coverage only looks at the coverage of positive instances, it is adapted as well.

2.3.4. Confidence coverage with negative feature mining (ConfCovNeg)

Confidence coverage does not cover relations between the negative class and a feature. For example, if a patient does not have hypothermia (ICD-9-CM code 991.6) and one of the features is 'sufficiently_warm', this feature could be evidence to not assign the code. We would however not find this feature with the previously suggested confidence coverage method, because the confidence score between 'sufficiently_warm' and the class '991.6' will be low.

Since confidence coverage is designed to work well in sparse-class environments, we do not consider it for the negative class. This is because of the unbalanced nature of the data (e.g., patient stays not coded with 'hypothermia' would be more prevalent than patient stays coded with 'hypothermia'). When confidence is used to score features for a densely populated class, unrelated features can reach a high confidence by coincidence, since a feature will have more opportunities to be present in a frequently occurring class. This can be counteracted by introducing the complement of all binarized features. When running the confidence coverage method in the previously described manner, we will find the correlation between the complement of a feature and the positive class. A selected complement is again replaced with the original feature after feature selection. In the example, this would lead to the complement of a feature 'NOT sufficiently_warm' getting a high confidence value for predicting the class '991.6', and being selected as a relevant feature. The complement of a feature represents all cases where 'sufficiently_warm' was not explicitly mentioned. When proceeding to the classification stage, 'NOT sufficiently_warm' is replaced by the original feature 'sufficiently_warm'.

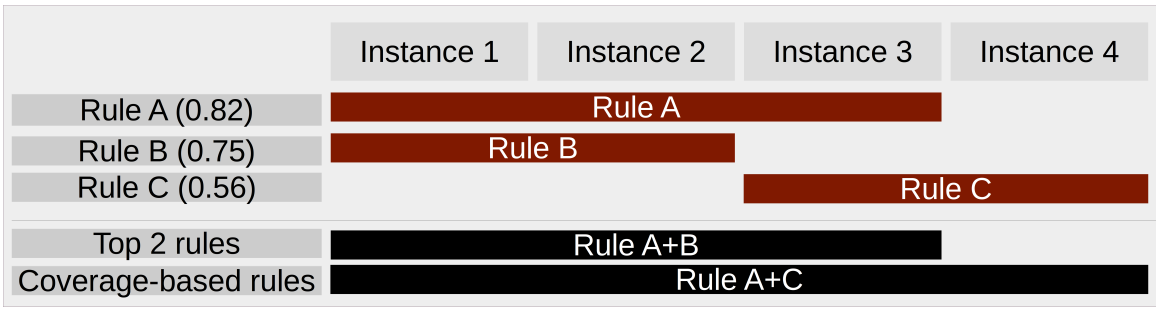

Figure 1: Visualisation of the instances covered by individual rules and the difference in coverage when using either top-N or unique coverage to select rules. A rule scoring higher on confidence is preferred.

2.4. Experimental setup

Each experiment is conducted with the four feature selection methods described above and with the original representation. For classification, we require a classifier that can handle a moderate to high number of features, such as Random Forests [29]. When presenting a classifier with all selected features, including the ones that were relevant to other clinical codes according to the feature selection method, the number of features is still quite high. All experiments are conducted in a ten-fold cross-validation setting. When cross-validation is used at the same time for both hyperparameter optimization and the comparison of different models, an overoptimistic estimation is likely, due to the large number of different variants of a model yielded by hyperparameter optimization [30]. A first solution, performing cross-validation only over the train set and using the resulting optimised classifiers on a single test set, will have the results reflect any accidental bias that might be present in the selected test set. A more robust way to optimally estimate hyperparameters is nested cross-validation, but this implies a high computational cost, certainly when many parameters need to be optimized.

Instead, parameter optimization for our experiments is performed within a ten-fold cross-validation loop by using a train/development/test split, where one of the nine data partitions reserved for training was used as a development set. With this development set, we have optimized parameters for the feature selection algorithm (such as cutoff thresholds, the number of retained features per class). No parameter optimization was used for Random Forests. A rough search of parameters mitigates overfitting due to small gains in performance on a development set. This is particularly relevant when no nested cross-fold validation is employed to assess these optimal parameters. For each fold, a final model is then trained, using both train and development sets with the optimal parameters from the previous phase.

Results achieved with other classification algorithms were performed as a separate experiment, and are presented in additional materials (e.g., C4.5, naive Bayes).

Micro-averaged F-measure is the main reported measure, which is the harmonic mean of precision and recall. In micro-averaging, each individual prediction is weighted equally, which means that a highly frequent class contributed more to the final result. Micro-averaging provides a good overall indicator of the performance of the model, while macro-averaging over class is a stronger indicator of performance for sparse classes, and macro-averaging over instances is an indicator on how well a model performs for individual instances.

These three averaging methods differ in how they normalise the predictions, so a different aspect of the multi-label task is being highlighted [31]: Micro-averaged F-measure does not normalise over any aspect, so each prediction is reflected equally. With $L = \alpha_j : j = 1..q$ being the set of all labels, $S = \beta_j : j = 1..n$ being the set of all instances, the metrics can be represented as follows:

$$F_{micro} = 2\frac{precision * recall}{precision + recall}$$

where

$$precision = \frac{\sum_{\alpha=1}^{q}(TP_\alpha)}{\sum_{\alpha=1}^{q}(TP_\alpha) + \sum_{\alpha=1}^{q}(FP_\alpha)}$$

and

$$recall = \frac{\sum_{\alpha=1}^{q} (TP_\alpha)}{\sum_{\alpha=1}^{q} (TP_\alpha) + \sum_{\alpha=1}^{q} (FN_\alpha)}$$

$$F_{macroClass} = \frac{1}{q} \sum_{\alpha=1}^{q} \left(2 * \frac{precision_\alpha * recall_\alpha}{precision_\alpha + recall_\alpha} \right)$$

$$F_{macroInstance} = \frac{1}{n} \sum_{\beta=1}^{n} \left(2 \frac{precision_\beta * recall_\beta}{precision_\beta + recall_\beta} \right)$$

All results should be considered in their own specialty. For comparability however, we are also referring to the averages over the different specialties. These averages are calculated by weighing the results of each individual specialty with the number of patient stays occurring in each specialty. This reduces the contribution of smaller datasets to the overall results.

3. Results

To test whether combining multiple sources increases performance, we first perform a learning curve experiment. We combine n data sources and retrieve the best integration for each specific combination. Instead of testing all possible combinations (which would require 255 experiments for each specialty, given 8 different input sources), the combinatory experiments were approached iteratively: in a first step, we rank the scores of all single source experiments. The sources are then combined in a pairwise fashion, where the two best scoring sources are joined, the next two sources are also joined, etc. We then expand the best scoring pair with the source that scored best in the single source experiments. If that source is already part of the pair, we use the highest ranking source that is not. This is repeated until we converge to a setup where all sources are present.

We preferred this technique over using a technique which performs a random search over parameters (which is proposed as an alternative to a full grid search over parameters). With a random search, N number of random combinations of sources are tested out, which similarly enables approaching the optimum combination of sources. While our pragmatic search technique is not exhaustive, it does not influence results for the best single source and total integration.

In Figure 2, an example of the learning curve is shown for predicting ICD-9-CM diagnosis codes for the UZA1-urology dataset (Tables 2 and 3 present results for best performing combinations for other specialties). The top edge of the area corresponds to the best integration for each set of n sources. This best integration is used for further experiments: we first determine the best integration of sources for each dataset, and then experiment with the feature selection methods on that specific combination of data sources. Our experiments showed that LMI-based concepts (which are one of two available representations for free texts) are the best scoring individual source for predicting diagnostic codes in 8 out of 9 datasets, and RIZIV-codes are the best scoring individual source for procedure codes in 7 out of 9 datasets.

Figure 3 shows very similar results (in micro-averaged F-measure) for the confidence coverage method (ConfCov) and the ConfCov variant with negative feedback (ConfCovNeg), across four datasets. Both methods outperform the baseline, the gain ratio method and the information gain coverage method (InfoGainCov).

Tables 2 and 3 show the micro-averaged F-measures obtained with the best individual source, the best integration of sources, and the combination of all sources for all feature selection methods and datasets. Table 2 shows results for diagnosis code prediction and Table 3 shows results for procedure code prediction. The significance of these results is demonstrated using approximate random testing against the baseline and can be found in supplementary materials [32].

For diagnosis codes, confidence coverage yields the highest micro-averaged F-measure overall. When comparing total integration to selecting the best individual source, performance of coverage-based methods seems to be affected more. We see an average increase of 5.43% for ConfCov, 5.35% for ConfCovNeg, 5.07% for InfGainCov, 3.54% for GainRatio, and 2.79% for the baseline. The resulting F-measure for total integration is at or very close to the F-measure seen in the best integration, which is not always the case for gain ratio or the baseline (an average difference of 0.08% for ConfCov, 0.41% for GainRatio, and 0.64% for the baseline).

For procedure codes, the same trend emerges, with confidence coverage scoring best overall and

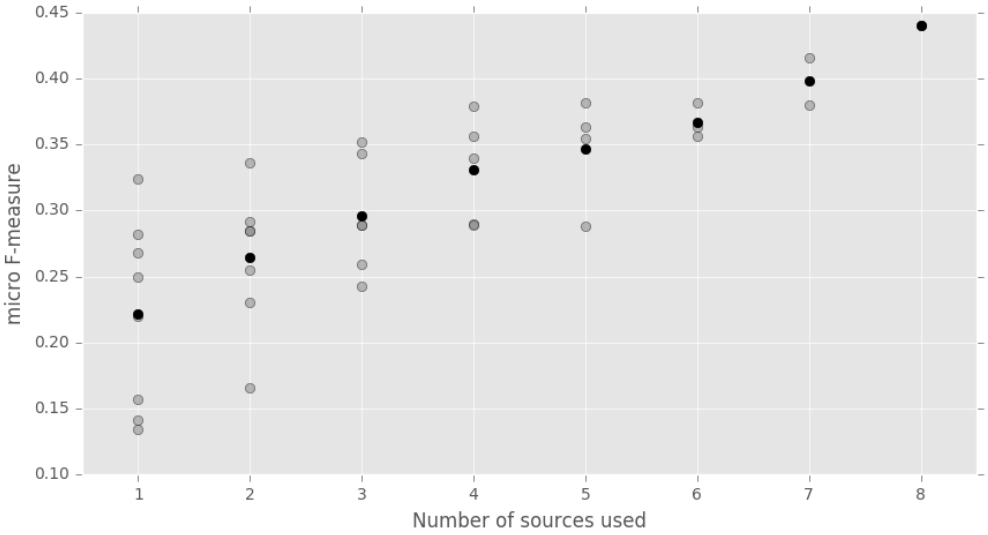

Figure 2: Micro-averaged F-measure range for confidence coverage with increasing number of data sources used, for predicting ICD-9-CM diagnosis codes in the UZA1 dataset urology. The black dots indicate the average performance of the combinations, while the grey dots indicate specific combinations.

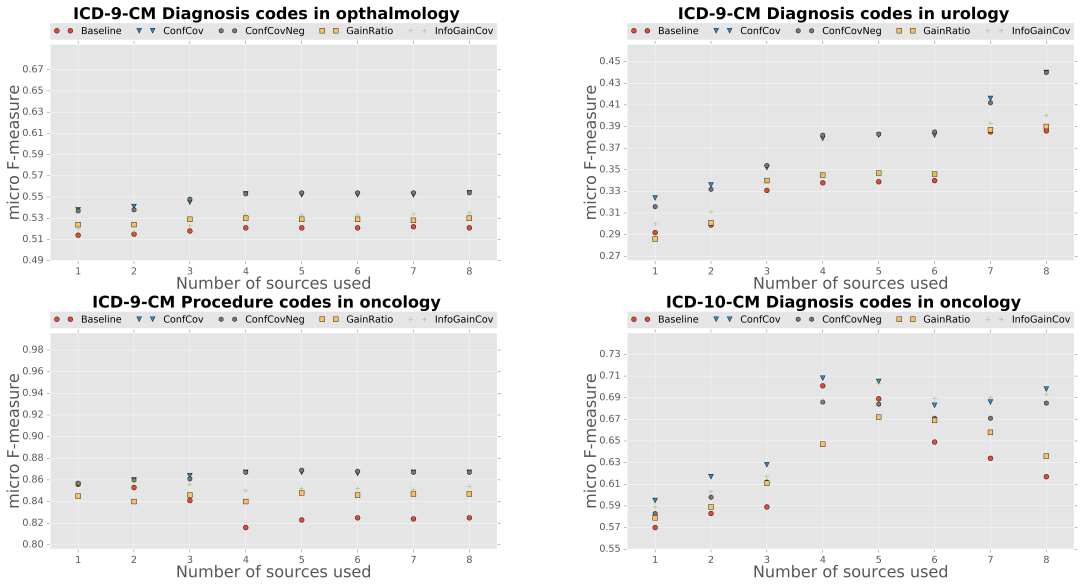

Figure 3: Comparison of feature selection methods with micro-averaged F-measure with the best integration of n data sources for prediction of diagnosis and procedure codes in four datasets. ConfCov represents the confidence coverage method, ConfCovNeg represents the confidence coverage method with negative feedback, InfoGainCov represents the information gain coverage method and GainRatio represents the gain ratio baseline.

an average difference in micro-averaged F-measure between using the best individual source and applying total integration of 2.52% for ConfCov, 2.40% for ConfCovNeg 0.07% for InfGainCov, 1.69% for GainRatio, and -0.86% for the baseline. The difference between the best integration of sources and total integration is larger, with an average difference of 0.16% for ConfCov, 0.50% for GainRatio, 1.57% for the baseline, and, surprisingly, 1.47% for InfGainCov. Overall, information gain coverage seems

to struggle more. Using GainRatio as feature selection is near to the baseline level, with a worse performance for some specialties, but integration with GainRatio works better.

Finding negative correlations between features and classes (ConfCovNeg) does not significantly improve results yielded by the confidence coverage method. The number of negative correlations found was low. This can also be seen in Table 4, where the number of selected features between ConfCov and ConfCov-Neg hardly differs.

3.1. Runtime complexity and number of features

Table 4 shows that confidence coverage variants select the lowest number of features, whereas gain ratio tends to select the most features. For procedure codes, we see that information gain coverage also selects more features compared to the other methods. The results suggest that coverage-based algorithms outperform non-coverage-based methods while needing a much lower number of features. This is mostly due to the removal of similar features, which decreases redundancy between features in the classifiers.

In Table 5, we present runtimes of the experiments on a server with 32 CPUs and 40 GB of RAM. These runtimes are indications only of the complexity of the experiments, due to limited control over memory usage and parallelisation. They include the time needed to train a feature selection model and to train a classification model for 10 folds, but do not include inner-fold parameter optimization, data extraction from the databases, or the calculation of evaluation metrics. The experiments train classifiers for different codes in parallel, but this is automatically managed depending on the memory still available. The runtimes should therefore only be treated as indications, and can vary depending on server architecture and other tasks running on the server. Feature selection and the training of a classification model are both taken into account, as the former has a large influence on the runtime of the latter.

If we compare 'total integration' and 'single source' experiments, we can see that the impact of additional data sources on runtime is not large for most experiments, except when gain ratio is used. Gain ratio in general is a component that requires a lot of calculations, compared to the (simpler) confidence-coverage. The relation between the number of features (seen in Table 4) and the runtime does not increase exponentially.

We see that experiments with confidence-coverage have a lower runtime complexity than experiments with other techniques, except for the cardiology and MIMIC-III datasets. When we compare runtime for diagnostic codes and procedure codes in the same specialty, we see that an increase in the number of classes directly leads to a roughly linear increase in runtime. This is to be expected: the used feature selection methods are calculated per class, and a separate classifier was trained for each individual class.

A comparison of runtime over specialties, to see the effect of having larger or smaller datasets, shows that runtime increases linearly with the size of the dataset as well, for procedure codes. For diagnostic codes, we see that the runtime is increased by both the higher number of instances and the higher number of classes, leading to a quadratic increase in the worst case.

Theoretically, database-coverage-based methods scale linearly with an increase in features, as a comparison is made directly with the unison coverage of all other features, instead of a pairwise comparison between features, as is often the case in feature selection methods considering redundancy. Confidence as a metric also scales linearly with the number of classes, as it is calculated for each class separately. The combination of both thus scales linearly in complexity with an increase in features and classes, but increases with their product when both are increased. This can be a problem when new instances are added to a dataset, as long as they bring new features and classes, but with time, the number of features and classes will converge (which would be the case when a dataset is completely representative for their medical specialty).

3.2. Instance-based and class-based evaluation.

In Table 6, different measures can be seen for all feature selection methods, averaged over the different datasets. Instance-based F-measure and exact match both reflect how well the algorithm performed for individual instances (for the average, each dataset is weighted with the number of instances in its set). With instance-based F-measure, each instance contributes evenly to the final results. Exact match shows the ratio of instances where all codes were predicted correctly for all classes. For class-based F-measure, each class contributes evenly to the results. For the average, each dataset is weighted with the number of classes in its set.

Table 2: micro-averaged F-measure (in %) reached when predicting diagnosis codes. Different datasets are shown in columns, different feature selection methods are shown in rows. For each feature selection method, we show the best scoring individual source used, the best scoring combination of sources ('best integration'), and the results when all sources were used to make predictions ('total integration').

	ICD-9-CM						ICD-10-CM		MIMIC-III
	Cardio	**Gastro**	**Onco**	**Ophtal**	**Pneumo**	**Uro**	**Cardio**	**Onco**	**ICU**
Baseline single source	43.5	26.5	54.8	51.4	44.7	29.2	36.6	57	32.1
Baseline best integration	45.1	27.7	61.8	52.2	51.9	38.6	37.7	64.9	32.4
Baseline total integration	45.1	27.7	61.8	52.1	45	38.6	37.1	61.7	32.4
GainRatio single source	44.4	26.9	54.4	52.4	40.4	28.6	38.2	57.9	32.1
GainRatio best integration	46.1	29	62.2	53	48.1	39	39.7	67.2	32.8
GainRatio total integration	46.1	28.8	62.2	53	44.9	39	39.7	63.6	32.8
ConfCov single source	45.1	33	56	53.8	44.7	32.4	39.6	59.5	36.8
ConfCov best integration	48.4	36.1	68.4	55.4	52.3	44	42.4	70.8	37.1
ConfCov total integration	48.3	36.1	68.4	55.4	52.3	44	42.3	69.8	37.1
ConfCov-Neg single source	45	32.9	55	53.7	44.7	31.6	39.6	58.3	36.8
ConfCov-Neg best integration	48.2	36.2	66.9	55.4	51.5	44	42.4	68.6	37.1
ConfCov-Neg total integration	48.2	36.2	66.9	55.4	51.5	44	41.9	68.5	37.1
InfGainCov single source	44.9	27.2	55.1	52	44.7	30	38.5	58.9	29.5
InfGainCov best integration	46.4	30	69.3	53.5	49.6	40	40	70.3	30.7
InfGainCov total integration	45.9	30	69.3	53.5	49.5	40	39.8	69.3	30.3

Table 3: micro-averaged F-measure (in %) reached when predicting procedure codes. Different datasets are shown in columns, different feature selection methods are shown in rows. For each feature selection method, we show the best scoring individual source used, the best scoring combination of sources ('best integration'), and the results when all sources were used to make predictions ('total integration').

	ICD-9-CM						ICD-10-PCS		MIMIC-III
	Cardio	**Gastro**	**Onco**	**Ophtal**	**Pneumo**	**Uro**	**Cardio**	**Onco**	**ICU**
Baseline single source	81.4	68.2	85.6	88.3	74.4	65.1	69.4	72.2	47.5
Baseline best integration	82.2	68.2	85.6	88.7	74.4	65.3	70.7	75.2	49.3
Baseline total integration	81.4	66.1	82.5	88.6	71.6	64.5	70.6	69.2	49.3
GainRatio single source	83.2	67.6	84.5	90.2	74.8	64.3	71.5	68.6	48.7
GainRatio best integration	86.5	72	84.8	91	74.8	67.4	74.8	71.6	50.4
GainRatio total integration	85.9	70.1	84.7	91	74.3	67.1	74.6	71.6	50.4
ConfCov single source	83.6	70.5	85.6	90	74.8	65.3	72.4	71.9	52.1
ConfCov best integration	87.9	74	86.7	90.9	77	68.4	76.3	75.4	53.9
ConfCov total integration	87.8	74	86.7	90.8	77	68.1	76.2	75.4	53.9
ConfCov-Neg single source	83.6	70.4	85.7	90	74.8	65.9	72.2	71.7	52.1
ConfCov-Neg best integration	87.5	73.5	86.9	90.9	77	68.5	76.2	75.7	53.9
ConfCov-Neg total integration	87.5	73.5	86.7	90.8	77	68.5	76.1	75.7	53.9
InfGainCov single source	82.3	68.9	85.6	89.9	75.1	64.9	69.2	62	44.9
InfGainCov best integration	85	68.9	85.9	90.9	75.1	65.7	73.5	66.8	47.3
InfGainCov total integration	84.7	61.9	85.4	90.8	74	65.4	73.5	65.9	46.2

Only classes occurring more than 20 times in the training set are taken into account for the class-based F-measure, in all other measures they are included in the calculations.

In general, the same trends seen in micro-averaged F-measure are seen in these measures as well. For an instance-based evaluation, we see that the difference between the best integration setup and total integration is more pronounced. For diagnostic codes, assigning all labels in an instance correctly is quite hard with an exact match of 11.89% for complete integration with confidence coverage. The lowered performance when weighing classes equally in the final score compared to either instance-based averaging and micro-averaging shows that classes that are more frequent are also predicted better.

4. Discussion

4.1. Findings

Overall, confidence coverage yields better performance than the other feature selection methods, due to its focus on selecting features that provide coverage of the entire dataset. Most feature selection methods find the best correlations between fea-

Table 4: Number of features selected when predicting clinical codes. Each column represents a dataset, and each row represents a feature selection method.

	UZA1						UZA2		MIMIC
	Cardio	**Gastro**	**Onco**	**Ophtal**	**Pneumo**	**Uro**	**Cardio**	**Onco**	**ICU**
Diagnostic codes									
Baseline single source	4848	5992	3934	1996	35	2278	1645	1084	6815
Baseline total integration	7389	8768	6006	2891	4995	3655	3546	2334	7950
GainRatio single source	3683	4329	3338	1086	26	1397	767	801	6181
GainRatio total integration	5358	6114	5029	1641	3741	2357	1226	1402	7065
ConfCov single source	2447	1725	2609	563	35	806	492	610	1769
ConfCov total integration	3008	2025	3128	600	1440	950	589	819	1938
ConfCov-Neg single source	2462	1726	2593	567	33	812	522	608	1769
ConfCov-Neg total integration	3060	2025	3137	605	1453	959	605	841	1938
InfGainCov single source	2415	2132	2635	908	32	982	366	544	3068
InfGainCov total integration	2886	2527	3333	1049	1239	1434	493	814	3110
Procedure codes									
Baseline single source	634	521	467	1996	415	347	1645	243	6815
Baseline total integration	7389	8768	6006	2891	4995	3655	3546	2334	7950
GainRatio single source	234	134	140	742	220	148	352	63	4777
GainRatio total integration	2013	2822	1971	1155	1470	1036	653	336	5534
ConfCov single source	216	191	219	156	120	164	162	109	1367
ConfCov total integration	478	492	409	150	181	274	177	205	1425
ConfCov-Neg single source	215	191	212	158	120	164	171	101	1367
ConfCov-Neg total integration	480	494	406	151	181	275	187	195	1425
InfGainCov single source	572	397	378	950	306	293	395	152	2884
InfGainCov total integration	2558	2038	2275	1113	830	1214	505	528	3016

Table 5: Runtime for feature selection and classification when predicting clinical codes. Each column represents a dataset, and each row represents a feature selection method. The time shown represents hours:minutes:seconds.

	UZA1						UZA2		MIMIC
	Cardio	**Gastro**	**Onco**	**Ophtal**	**Pneumo**	**Uro**	**Cardio**	**Onco**	**ICU**
Diagnostic codes									
Baseline single source	03:01:13	02:10:30	06:34:19	00:16:26	00:14:18	00:14:22	00:04:38	00:19:47	02:36:02
Baseline total integration	04:42:01	02:03:33	05:33:41	00:16:19	00:44:03	00:25:50	00:04:13	00:21:35	02:33:30
ConfCov single source	06:03:56	01:54:20	04:23:25	00:07:13	00:05:55	00:11:50	00:02:56	00:12:50	06:05:40
ConfCov total integration	06:50:17	02:00:58	04:17:36	00:07:32	00:28:02	00:19:47	00:03:40	00:08:15	06:07:37
ConfCovNeg single source	02:39:57	01:59:44	04:51:31	00:08:27	00:05:28	00:12:11	00:03:21	00:12:50	05:51:20
ConfCovNeg total integration	03:11:20	02:05:09	04:12:42	00:09:35	00:29:27	00:18:56	00:04:42	00:10:03	06:31:20
GainRatio single source	05:25:13	02:55:44	08:35:41	00:11:57	00:04:48	00:15:51	00:03:39	00:17:23	12:33:45
GainRatio total integration	07:37:59	04:13:46	09:45:47	00:14:05	00:50:55	00:30:30	00:04:47	00:11:46	15:15:27
InfoGainCov single source	04:42:07	02:19:58	05:57:50	00:12:51	00:05:39	00:16:23	00:03:34	00:12:52	10:36:26
InfoGainCov total integration	05:23:35	02:59:49	07:11:27	00:14:31	00:29:41	00:28:58	00:04:32	00:10:22	13:59:00
Procedure codes									
Baseline single source	00:14:31	00:08:25	00:09:25	00:04:37	00:01:52	00:01:34	00:02:32	00:02:08	00:53:29
Baseline total integration	00:25:43	00:16:47	00:21:29	00:04:43	00:04:34	00:04:10	00:02:44	00:05:00	00:53:26
ConfCov single source	00:15:16	00:02:43	00:03:34	00:01:30	00:00:45	00:00:58	00:01:17	00:01:49	02:17:40
ConfCov total integration	00:28:54	00:10:25	00:08:05	00:01:52	00:01:57	00:02:09	00:01:47	00:02:55	02:14:39
ConfCovNeg single source	00:07:59	00:03:21	00:04:38	00:02:33	00:00:51	00:01:03	00:01:36	00:01:53	02:16:48
ConfCovNeg total integration	00:21:10	00:16:04	00:15:08	00:03:42	00:03:36	00:03:04	00:02:34	00:03:35	02:14:55
GainRatio single source	00:12:57	00:04:14	00:05:35	00:02:51	00:00:52	00:00:59	00:01:36	00:01:54	04:07:24
GainRatio total integration	01:05:34	00:27:13	00:35:44	00:03:53	00:04:27	00:04:42	00:02:13	00:04:13	04:51:21
InfoGainCov single source	00:14:15	00:04:39	00:06:22	00:03:27	00:01:00	00:01:18	00:01:56	00:02:05	03:32:51
InfoGainCov total integration	00:49:22	00:26:13	00:29:18	00:04:06	00:03:08	00:04:45	00:02:23	00:04:17	03:50:26

Table 6: Measures averaged over the different datasets. Different measures are shown in columns, different feature selection methods are shown in rows. Instance-based F-measure and exact match are averaged over instances, class-based F-measure is averaged over classes.

	Diagnostic codes			Procedure codes		
	Instance-based F-measure	Exact match	Class-based F-measure	Instance-based F-measure	Exact match	Class-based F-measure
Baseline single source	41.78%	10.11%	28.67%	61.09%	57.33%	45.82%
Baseline best integration	47.21%	11.53%	33.68%	61.98%	57.71%	46.74%
Baseline total integration	44.77%	10.66%	31.96%	59.73%	55.70%	45.31%
GainRatio single source	42.66%	10.25%	28.20%	61.80%	58.22%	47.05%
GainRatio best integration	47.30%	11.51%	33.49%	63.71%	59.85%	48.95%
GainRatio total integration	46.34%	10.98%	32.80%	61.52%	59.32%	48.46%
ConfCov single source	46.50%	10.54%	31.64%	60.67%	59.17%	49.52%
ConfCov best integration	53.66%	12.46%	38.71%	63.06%	61.89%	51.97%
ConfCov total integration	51.09%	11.89%	38.66%	61.52%	61.59%	51.76%
ConfCov-Neg single source	43.41%	10.53%	31.16%	62.54%	59.22%	49.36%
ConfCov-Neg best integration	49.66%	12.11%	38.23%	64.45%	61.83%	51.70%
ConfCov-Neg total integration	48.81%	11.61%	38.18%	62.13%	61.63%	51.59%
InfGainCov single source	43.47%	10.49%	29.62%	60.92%	57.79%	45.70%
InfGainCov best integration	49.01%	12.54%	36.10%	63.45%	59.47%	47.31%
InfGainCov total integration	46.16%	11.82%	35.77%	61.58%	57.79%	46.58%

tures and classes, but disregard the specific training instances corresponding to that feature. Confidence coverage considers both the association with the class (confidence) and the number of instances supported by a feature-class association (coverage) and optimizes for both factors. Coverage works best in a sparse-class environment, as the number of selected features correlates linearly with the number of instances of a certain class. For a sparse class, fewer features get selected. Confidence proves to be a strong metric in this environment. In contrast to information gain, the confidence score of a feature is not influenced by the frequency of a feature in the dataset, since it disregards instances that it does not occur in. This improves integration qualities: a feature that occurs less often, but co-occurs often with a certain class has a high confidence.

One of the advantages of using a feature selection method that takes dataset coverage into account is that it decreases ambiguity. If two features have a strong correlation, the method will preserve the feature with the highest confidence score. This effect is also seen when applying feature selection methods such as mRMR [22] or other co-occurrence based algorithms. However, the algorithm is computationally more efficient and directly evaluates redundancy of a feature for the task, since features are compared on dataset coverage instead of compared against other features. When applying confidence coverage, the highest F-measure on a combination of data sources comes close to the F-measure obtained by using all sources. This indicates that a confidence coverage-based method is able to deal with noisy sources and therefore is more suitable to be applied when there is uncertainty about the quality of any given source.

Information gain coverage and gain ratio underperform in the current setup, partly due to the combination with Random Forests. Random Forests use information gain internally to create branches within trees. Selecting features with a higher information gain causes more uniformity within the individual trees, which in turn causes overfitting, leading to worse results. This also explains the results of gain ratio being close to the baseline: combining gain ratio and Random Forests makes gain ratio act partly as a pretraining phase. When replacing Random Forests with a naive Bayes clas-

sifier, information gain coverage outperforms confidence coverage (results shown in additional materials). The introduced algorithms still outperform gain ratio. The models presented here outperform the late data integration technique (LDI) because LDI overgeneralizes by reducing each data source to a single data point [15]. The concept-based text representations included in the current models but not in LDI (where a bag-of-words representation was selected) are another reason why LDI is outperformed. For the MIMIC-III dataset, the improvement when combining data sources is minimal, but confidence coverage still yields better performance than gain ratio. Perotte et al. reached an F-measure of 29.3% on MIMIC-II with hierarchical prediction, when they only consider the labels assigned as true positives [9]. A less strict evaluation - considering a code that is hierarchically close to the clinical code assigned (i.e., an ancestor or descendant) as a true positive - caused substantial improvement (39% F-measure). However, our approaches score substantially better, while still taking into account a strict evaluation.

In addition to the results on the subset of MIMIC-III, we have run preliminary experiments for confidence-coverage on the entire MIMIC-III dataset, consisting of 58,970 patient stays, 6,834 unique diagnostic codes, and 1,978 procedural codes. In this setup, we have used 10% of the data as testing data, and configured the pipeline based on the optimally determined results in our previous MIMIC-III experiments. For diagnostic codes, this resulted in a micro F-measure of 42.4% for the best individual source and 42.8% for total integration. Procedure codes were predicted with an F-measure of 53.9% for the best individual source and 55.5% for total integration. These results confirm the same trends seen in the subset, but do show that the algorithm can increase in overall performance when more data is provided.

For procedural codes specifically, we see that RIZIV-codes (which in itself represent procedures, although less complex than ICD-codes) are very informative and are often the best individual source to assign ICD-9 and ICD-10 procedure codes. It is certainly possible to create a partial map between RIZIV codes and ICD-9 codes (for ICD-10 codes, this is less often possible), and the results show that the algorithm succeeds quite well in finding the correlations between RIZIV and ICD.

In general, a correlation between the number of unique codes, label cardinality in a specialty, and the achieved results can be seen. Because an increased number of unique codes and higher label cardinality indicate that a dataset is more complex, this complexity also influences the achieved results. This however does not mean that introducing another coding system in a specialty implies clinical code assignment will become harder. Our early results on ICD-10-CM/PCS encoded stays show that our system is able to cope with this new clinical coding system. It is expected that when ICD-10-CM/PCS coding is used for a longer period, the results may be affected, but this is true for all approaches currently developed, including the ones we compare with. The effect of having more fine-grained codes on performance is unknown: while this can introduce more ambiguities, it also resolves some issues with insufficiently granular ICD-9-CM codes.

4.2. Runtime

While runtime is an important issue that proposed algorithms must deal with, the computational complexity during development is significantly higher than during usage in a real-world application. The latter only requires training an algorithm once, with optionally the recalculation of hyperparameters, if these were not yet determined during development. During development, multiple iterations (for cross-validation) need to be performed, and a whole series of hyperparameters need to be determined for each fold in that cross-validation setup, with each unique combination of hyperparameters requiring a separate trained model (or multiple, if nested cross-validation is considered).

Further improvements on our setup regarding runtime can be made if techniques other than binary relevance are considered for multi-label classification, as training a separate classifier for each class remains costly [33].

4.3. Evaluating a component in a larger prediction pipeline

In a prediction pipeline, many components have an effect on the final results [34]. When evaluating a single component, it is essential to leave all other components either unchanged or optimized on a development set. When investigating the feature selection component, we must take into account that other components may be affected. For example a classification algorithm (or a certain parameter set)

might work better with one algorithm than with another. By using algorithms where many hyperparameters are optimized, this effect is stronger, as its chances for overfitting with a certain parameter set greatly increase [35, 30].

In our work, we specifically look for components that require less optimization, because training many extra models to determine the optimal parameter set internally is computationally infeasible and increases the risk of overfitting on the development set [35]. One of the advantages of Random Forests is that it performs well without tuning any hyperparameters, as its main criterion, the number of trees, converges to its best result when the number of trees is increased, and the number of randomly selected features per tree is near-optimal with the proposed default settings [29, 36]. All components before feature selection were kept unchanged over the different experiments, and were not determined with a feature selection algorithm in mind.

4.4. Feasibility and advantages of data integration

In general, we see that unstructured information, albeit noisy, is one of the few sources that is always present in an electronic health record. Structured information often is as well, but uses a different structure and notation over different hospitals, and clinical coders have varying degrees of access to these sources, while they often have access to the discharge file of a patient. For procedure codes, we see that a structured source (RIZIV) is often the main contributing factor in our datasets, but as a local Belgian standard, the presence of this source is not a given for datasets from other hospitals. For diagnostic codes, we see that unstructured sources actually contain most relevant information (as seen by the main contributing source being LMI-based concepts in most cases).

The need for 'total integration depends on the form in which data is available in a hospital, and whether the information about individual patients presents itself in a single source or in multiple sources. Integrating all sources available has two main advantages for the hospital: if data is fragmented and incomplete, spread over a multitude of sources, and of varying quality over those sources, our technique can integrate information present in all of them. Secondly, identifying the most important source of information is not always trivial, and our proposed algorithm removes the need to do that. In general, we see that the main contribution of confidence-coverage is not necessarily the integrative part of the sources, but its ability to process sources that contain a lot of redundant information from a data perspective, such as unstructured clinical notes.

In a dataset where information is concentrated in a single source (which is the case for the MIMIC-III dataset, from which we extracted clinical notes, NDC codes, LOINC codes, and departmental information), we see that the confidence-coverage algorithm succeeds in significantly outperforming both the baseline and gain ratio. The best performing single source used in MIMIC-III contains redundant information, since it is made up of mMWEs extracted from multiple unstructured texts associated with a patient. In the UZA-based datasets, most information is captured using unstructured sources as well, but we see that discharge files often only contain a general description of certain procedures. To find specifics about a certain procedure, we need to look at the surgery report, or at the RIZIV codes associated with them. The confidence-coverage technique is an automatic, independent assessment of a dataset, relieving us of the need for determining a priori which data sources (or which element in the data source) are important. Unfounded assumptions about which data source contains the relevant information should be avoided and confidence-coverage allows for that.

4.5. Portability

Because the datasets used to predict ICD-10-CM and ICD-10-PCS codes are rather small, the results do not scale to a bigger ICD-10 dataset, which would contain a higher code diversity. However, the techniques presented behave similarly on ICD-9-CM and ICD-10-CM datasets, indicating that the approach is applicable to predict both coding systems and suggesting portability to other coding systems. The proposed method represents any clinical source without the need to map to a rigid framework such as UMLS (e.g., converting localized lab tests to LOINC codes to make them compatible with UMLS-based concepts). Please note that while UMLS-based concepts are being used as one of the two representations for free text, the algorithm itself would be able to produce results without this representation and does not depend on it. This is a clear advantage, since the algorithm is able to deal with a larger variety of notations.

For our approach, we preferred a technique that performs well on integrating as many sources as possible, even in cases where the difference between results on the best individual source and total integration is minimal. The algorithm chooses features that are of most interest to a clinical code, but it can also benefit from less interesting features (and sources) when a particular source for a specialty differs in information content as compared to its information content in other specialties. If an algorithm is able to select features from all sources without having a loss in performance, there is no need to investigate the informativeness from all individual sources, for each different specialty, beforehand.

4.6. Abstraction of features

The representation generated by using confidence coverage is not abstracted at all, but rather summarizes the original data by selecting a portion of the features. Not using an abstraction (which would combine multiple features into one) can be a constraint. For example, if 'chronic_lymphocytic_leukemia' is found as a concept in text, and the ICD-O code '9823/3' is found in a structured source, both of these could be mapped to a class representing 'CLL/SLL'. If both these features map to roughly the same instances, confidence coverage only selects one, but it does not capture a compound of both. On the other hand, a severe abstraction that is generated by an autoencoder-based representation of patients [16] or a late data integration approach [15] can also be a constraint, as they tend to overgeneralize, losing the granularity of information needed to predict closely related clinical codes (e.g., both ICD-10-CM 'I35.1' and 'I35.2' represent 'Nonrheumatic aortic (valve) insufficiency', but the latter also includes stenosis).

4.7. Assisting clinical coders

The system described in this paper automatically suggests diagnosis and procedure codes. The algorithm is therefore positioned more as part of the backbone of a computer-assisted coding application. During the development process, the implementation was specifically tailored with such an application in mind. It allows for extra information being provided, such as the reason(s) why a certain code is proposed and probabilities of certain codes occurring. This enables ranking of the codes predicted and tuning recall and precision to suit the needs of the application.

When assisting human coders, the algorithm can be employed in several applications. For quality control, the algorithm allows for correcting errors in the assigned clinical codes. An application that automatically codes 'easy' patient stays, with codes that are often assigned and are predicted with a high precision, would not require human intervention and speed up the flow of clinical coders. To assist clinical coders directly in their flow, an application should employ a ranking approach, where the clinical codes are ranked based on the confidence the algorithm has. This would allow a human coder to look at suggestions, speeding up the process as well, and requiring a higher recall (as all possible codes should be provided as a suggestion).

The patient stays used in the experiments were unfiltered (except for de-identification) and provided as-is to the classifier to replicate the setup that a clinical coder would have. This has the effect that multiple stays of the same patient can end up in the dataset. A patient stay only contains information that was documented during the period of their stay. This information is linked to a single stay only, which ensures that no direct information is transferred between stays. Indirectly, stays from the same patient will be more similar to each other, and can thus yield an advantage to the classifier, but this advantage is also present when other algorithms, or clinical coders, are assigning codes.

All of these applications have their own requirements for algorithm performance, and require different measures to be prioritized. In this study, we were not directly evaluating the efficacy of the algorithms for a specific application and thus report on measures that present the overall performance.

5. Conclusions

Predicting clinical codes in a real-world dataset based on all information linked to a patient stay is more successful than limiting oneself to discharge summaries, or any other single data source. To successfully use a machine learning approach for this task, we evaluated the relevance of the clinical information. We experimented with confidence coverage and information gain coverage as feature selection methods to efficiently and effectively represent a patient's information to a clinical coding algorithm. This yields better performance than using a traditional feature selection method such as gain ratio.

A confidence coverage approach allows for using information found in any clinical source, without the need to link it to a rigid framework, such as the UMLS metathesaurus. The features can be derived directly from the source data, which allows for a better interpretation of the selection. The resulting representation contains less redundancy since only one of multiple similar features would be selected due to the database coverage aspect. This representation also shows a strong reduction of the number of extracted features. We evaluated across a range of medical specialties, on ICD-9-CM, ICD-10-CM and ICD-10-PCS codes, on datasets in Dutch and the MIMIC-III dataset in English.

6. Acknowledgements

This work was supported by the Agency for Innovation by Science and Technology in Flanders (IWT) grant number 131137.

7. Competing interests

None

References

[1] C.-J. Hsiao, E. Hing, Use and characteristics of electronic health record systems among office-based physician practices: United States, 2001-2012., NCHS data brief (111) (2012) 1–8.

[2] G. S. Alotaibi, C. Wu, A. Senthilselvan, M. S. McMurtry, The validity of ICD codes coupled with imaging procedure codes for identifying acute venous thromboembolism using administrative data, Vascular Medicine 20 (4) (2015) 364–368. doi:10.1177/1358863X15573839.

[3] W. Q. Wei, P. L. Teixeira, H. Mo, R. M. Cronin, J. L. Warner, J. C. Denny, Combining billing codes, clinical notes, and medications from electronic health records provides superior phenotyping performance, Journal of the American Medical Informatics Association 23 (e1) (2016) 20–27. doi:10.1093/jamia/ocv130.

[4] World Health Organization, International classification of diseases (icd) (2012) .

[5] H. Lamberts, I. Okkes, et al., Icpc-2, international classification of primary care (1998) .

[6] S. Pakhomov, J. D. Buntrock, C. G. Chute, Automating the Assignment of Diagnosis Codes to Patient Encounters Using Example-based and Machine Learning Techniques, Journal of the American Medical Informatics Association 13 (5) (2006) 516–525. doi:10.1197/jamia.M2077.

[7] M. H. Stanfill, M. Williams, S. H. Fenton, R. A. Jenders, W. R. Hersh, A systematic literature review of automated clinical coding and classification systems., J Am Med Inform Assoc 17 (6) (2010) 646–651. doi:10.1136/jamia.2009.001024.

[8] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-W. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, R. G. Mark, MIMIC-III, a freely accessible critical care database., Scientific data 3 (2016) 160035. doi:10.1038/sdata.2016.35.

[9] A. Perotte, R. Pivovarov, K. Natarajan, N. Weiskopf, F. Wood, N. Elhadad, Diagnosis code assignment: models and evaluation metrics, Journal of the American Medical Informatics Association 21 (2) (2014) 231–237. doi:10.1136/amiajnl-2013-002159.

[10] M. Subotin, A. R. Davis, A System for Predicting ICD-10-PCS Codes from Electronic Health Records, Workshop on BioNLP (BioNLP) (2014) 59–67.

[11] S. Abhyankar, D. Demner-Fushman, F. Callaghan, Combining structured and unstructured data to identify a cohort of ICU patients who received dialysis, Journal of the American Medical Informatics Association 21 (5) (2014) 801–807.

[12] J. Pathak, K. R. Bailey, C. E. Beebe, S. Bethard, D. S. Carrell, P. J. Chen, D. Dligach, C. M. Endle, L. A. Hart, P. J. Haug, S. M. Huff, V. C. Kaggal, D. Li, H. Liu, K. Marchant, J. Masanz, T. Miller, T. A. Oniki, M. Palmer, K. J. Peterson, S. Rea, G. K. Savova, C. R. Stancl, S. Sohn, H. R. Solbrig, D. B. Suesse, C. Tao, D. P. Taylor, L. Westberg, S. Wu, N. Zhuo, C. G. Chute, Normalization and standardization of electronic health records for high-throughput phenotyping: the SHARPn consortium, Journal of the American Medical Informatics Association 20 (e2) (2013) e341–e348. doi:10.1136/amiajnl-2013-001939.

[13] O. Bodenreider, The Unified Medical Language System (UMLS): integrating biomedical terminology, Nucleic Acids Research 32 (suppl 1) (2004) D267–D270. doi:10.1093/nar/gkh061.

[14] RIZIV, Rijksinstituut voor ziekte- en invaliditeitsuitkeringen nomenclature, `http://www.riziv.fgov.be/NL/nomenclatuur/Paginas/default.aspx`, accessed: 2017-02-21.

[15] E. Scheurwegs, K. Luyckx, L. Luyten, W. Daelemans, T. Van den Bulcke, Data integration of structured and unstructured sources for assigning clinical codes to patient stays, Journal of the American Medical Informatics Association 23 (e1) (2016) 11–19. doi:10.1093/jamia/ocv115.

[16] R. Miotto, L. Li, B. A. Kidd, J. T. Dudley, Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records., Scientific reports 6 (April) (2016) 26094. doi:10.1038/srep26094.

[17] R. Cohen, M. Elhadad, N. Elhadad, Redundancy in electronic health record corpora: analysis, impact on text mining performance and mitigation strategies., BMC Bioinformatics 14 (1) (2013) 10. doi:10.1186/1471-2105-14-10.

[18] S. M. Vieira, L. F. Mendonça, G. J. Farinha, J. M. C. Sousa, Modified binary {PSO} for feature selection using {SVM} applied to mortality prediction of septic patients, Applied Soft Computing 13 (8) (2013) 3494–3504.

[19] T. Botsis, M. D. Nguyen, E. J. Woo, M. Markatou, R. Ball, Text mining for the Vaccine Adverse Event Reporting System: medical text classification using informative feature selection, Journal of the American Medical Informatics Association 18 (5) (2011) 631–638.

[20] I. Guyon, A. Elisseeff, An Introduction to Variable

and Feature Selection, Journal of Machine Learning Research 3 (2003) 1157–1182. arXiv:1111.6189v1, doi:10.1016/j.aca.2011.07.027.

[21] R. Kohavi, G. H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1-2) (1997) 273–324. doi:10.1016/S0004-3702(97)00043-X.

[22] H. C. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8) (2005) 1226–1238.

[23] S. Fu, M. C. Desmarais, Markov Blanket Based Feature Selection: a Review of Past Decade, in: Proceedings of the World Congress on Engineering 2010, Vol. I, 2010, pp. 321–328.

[24] I. Tsamardinos, L. E. Brown, C. F. Aliferis, The max-min hill-climbing Bayesian network structure learning algorithm, Machine Learning 65 (1) (2006) 31–78. doi:10.1007/s10994-006-6889-7.

[25] E. Scheurwegs, K. Luyckx, L. Luyten, B. Goethals, W. Daelemans, Assigning clinical codes with data-driven concept representation, Journal of Biomedical Informatics.
URL http://dx.doi.org/10.1016/j.jbi.2017.04.007

[26] J. Fürnkranz, G. Widmer, Incremental Reduced Error Pruning, International Conference on Machine Learning (1994) 70–77doi:10.1.1.43.7813.

[27] W. W. Cohen, Fast effective rule induction, Twelfth International Conference on Machine Learning (1995) 115–123doi:10.1.1.50.8204.

[28] W. Li, J. Han, J. P. Cmar, Accurate and efficient classification based on multiple class-association rules, In Proc. of ICDM pages (2001) 369–376. doi:10.1109/ICDM.2001.989541.

[29] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32. doi:10.1023/A:1010933404324.

[30] I. Tsamardinos, A. Rakhshani, V. Lagani, Performance-Estimation Properties of Cross-Validation-Based Protocols with Simultaneous Hyper-Parameter Optimization, International Journal on Artificial Intelligence Tools 24 (05) (2015) 1540023. doi:10.1142/S0218213015400230.
URL http://www.worldscientific.com/doi/abs/10.1142/S0218213015400230

[31] G. Tsoumakas, I. Katakis, I. Vlahavas, Mining multi-label data, Data mining and knowledge discovery handbook (2010) 667–685.

[32] A. Yeh, More accurate tests for the statistical significance of result differences, in: Proceedings of the 18th conference on Computational linguistics-Volume 2, Association for Computational Linguistics, 2000, pp. 947–953.

[33] M. L. Zhang, Z. H. Zhou, A review on multi-label learning algorithms, IEEE Transactions on Knowledge and Data Engineering 26 (8) (2014) 1819–1837. doi:10.1109/TKDE.2013.39.

[34] S. Strother, S. La Conte, L. Kai Hansen, J. Anderson, J. Zhang, S. Pulapura, D. Rottenberg, Optimizing the fMRI data-processing pipeline using prediction and reproducibility performance metrics I. A preliminary group analysis, NeuroImage 23 (2004) 196–207. doi:10.1016/j.neuroimage.2004.07.022.

[35] D. D. Jensen, P. R. Cohen, Multiple Comparisons in Induction Algorithms, Machine Learning 38 (3) (1999) 309–338. arXiv:0005074v1, doi:10.1023/A.

[36] S. Bernard, L. Heutte, S. Adam, Influence of Hyperparameters on Random Forest Accuracy, Lecture Notes in Computer Science 5519 (2009) 171–180. doi:10.1007/978-3-642-02326-2_18.