Decision Support for Data Segmentation (DS2):

Technical and Architectural Considerations

Mike Berry, Noam Arzt, Daryl Chertcoff, Martin French, Carl A. Gunter

June 2, 2014

Acknowledgements

The authors would like to thank the Illinois Office of Health Information Technology (OHIT) and the SHARPS DS2 team for contributing to the project: Bernie A'cs, Vincent Bindschadler, Ellick Chan, Mark Chudzinski, Alexander Danel, Mario Felarca, Sid Gupta, Ivan Handler, Jason Jacobs, Jean Kirupaharan, John Lekich, Maiko Minami, Antonios Michalos, Muhammad Naveed, Sam Nicolary, Seeun Oh, Sireesha Perepu, Thomas Redman, Tom Schultz, Pete Sfiridis, David Stumpf, Ting Wu.

This work was partially supported by HHS 90TR0003-01 (SHARPS) and the Illinois Office of Health Information Technology (OHIT). The views expressed are those of the authors only.

Contents

1 Abstract

In response to demand for solutions to implement privacy protections for certain types of electronic healthcare data while allowing sensitive health information to flow to authorized users, the health information technology community has been developing technical standards and solutions in a topic area known as Data Segmentation for Privacy, or DS4P.

This paper presents the results of a research-oriented project to demonstrate that certain DS4P tasks can be enhanced through the use of clinical decision support (CDS) technology. It advances a novel use of CDS tools to 1) identify and sequester certain types of information from electronic medical records and to 2) help mitigate the potential risks of exchanging records from which data have been sequestered.

The approach is called Decision Support for Data Segmentation, or DS2. It builds upon standards-based open source CDS technology to create a familiar CDS-based platform for the development and testing of functions to identify and redact selected conditions from clinical summary documents in various contexts including Health Information Exchange (HIE) between healthcare providers. The DS2 prototype demonstrates how deterministic clinical rules and machine learning-based classifiers can work together to detect clinical facts that may *imply* a condition even if they are not directly related to the condition and how CDS at the point-of-care can potentially make use of clinical information even after it has been sequestered.

2 Introduction

The term "data segmentation" refers to the process of sequestering from capture, access or view certain data elements that are perceived by a legal entity, institution, organization, or individual as being undesirable to share[1]. Data Segmentation can be used to help implement privacy protections for certain types of electronic healthcare data while allowing sensitive health information to flow more freely to authorized users; and is part of a broader vision to provide patients with more fine-grained control over the sharing of their electronic health records[2].

To develop technical standards and solutions related to data segmentation, the federal government, health IT vendors, healthcare providers, and other stakeholders collaborated within the Standards and Interoperability Framework[3] on an initiative known as Data Segmentation for Privacy (DS4P). The S&I Framework DS4P Initiative published use cases; created and standardized an Implementation Guide for

[1] Goldstein, M. M., & Rein, A. L. (2010). Consumer consent options for electronic health information exchange: policy considerations and analysis. Prepared for the Office of the National Coordinator for Health IT. Washington, DC: George Washington University Medical Center. Available at http://www.healthit.gov/sites/default/files/privacy-security/choice-model-final032610.pdf.

[2] For a complete discussion of this vision, see our companion paper, French, M., Nissenbaum, H., Berry, M., Arzt, N., Gunter, C. A. (2014). Decision Support for Data Segmentation (DS2): Contextual Integrity Considerations.

[3] The S&I Framework is an approach adopted by the Office of the National Coordinator for Health Information Technology (ONC) to establish and operate a forum for stakeholders to establish standards and other implementation guidance related to interoperability specifications for health information technology. For more information see http://www.siframework.org/.

using existing standards to implement DS4P solutions; and sponsored a number of pilot projects to demonstrate the effectiveness of standards-based data segmentation in the context of the use cases[4].

One of the challenges in certain types of segmentation is that the sequestration of a condition and its related clinical facts sometimes leaves clues – residual facts such as comorbidities and co-occurrences – that could still reveal the condition to an informed observer. Therefore a data segmentation system may need to take into account the *inferences* that might be made by an observer, in order to ensure that a condition is effectively segmented. As Clinical Decision Support (CDS) tools are increasingly being used to assist with inferencing as part of the clinical decision making process, we postulate that CDS can contribute to Data Segmentation for Privacy as well.

The goal of this paper is to introduce a technical architecture and open source prototype – Decision Support for Data Segmentation (DS2) – that leverages clinical decision support technology for the identification and sequestration of certain types of sensitive information from patient records flowing through a Health Information Exchange (HIE). We begin by proposing a framework based on three core functions – predicates, reducers, and safety checks – and then place those functions inside a service-based architecture driven by a clinical decision support system. We discuss some of the unique challenges in data segmentation and describe how knowledge-based and machine learning-based clinical decision support can assist with these challenges. We also propose a number of areas for future research, including potential uses of DS2 outside of the DS4P domain – in public health, clinical research, and clinical decision making. Finally, we describe our open source prototype software system that implements the DS2 architecture in an HIE environment.

3 Functions for Automated Data Segmentation

The DS2 approach assumes the existence of three types of functions[5]:

1. **“Predicates”** decide if a patient’s clinical document[6] contains a particular type of information that may need to be sequestered. For example, a predicate might be defined to check if a record reveals evidence of a mental health condition or treatment. If the record does contain such data, the mental health predicate returns “True;” otherwise it returns “False.”
2. **“Reducers”** remove or redact parts of the patient’s clinical document, based on asserted predicates and other clinical facts, to satisfy a predicate. For instance, a mental health reducer might be defined to remove data associated with or indicative of a mental health condition or treatment.
3. **“Safety Checks”** evaluate a proposed treatment or medication against the patient’s non-redacted clinical document to warn about any contraindications or other safety issues that may exist but are not visible in the redacted document. For instance, if a mental health medication

[4] DS4P Workgroup artifacts, including the use cases, Implementation Guide, and pilot projects, are available at http://wiki.siframework.org/Data+Segmentation+for+Privacy+Homepage.

[5] For a mathematical definition of the Predicate/Reducer model, see Chan, E. M., Lam, P. E., & Mitchell, J. C. (2013). Understanding the Challenges with Medical Data Segmentation for Privacy. In Presented as part of the 2013 USENIX Workshop on Health Information Technologies. Available at https://www.usenix.org/conference/healthtech13/workshop-program/presentation/Chan.

[6] Where the term “clinical document” is used in this paper, it refers to a Continuity of Care Document (CCD); specifically, the HL7 CCDA (HL7 Implementation Guide for CDA® Release 2: IHE Health Story Consolidation, Release 1.1 - US Realm, also known as the Consolidated CDA, or Consolidated Clinical Document Architecture).

in the record has a known interaction with a new prescription, the Safety Checker would produce a warning.

The three types of functions work together like this: An HIE has a predicate and reducer for each category of sensitive medical information that may need to be redacted from a patient's record upon transmission (depending on policy, patient consent, or other circumstances). For instance, the HIE might have a predicate and reducer for HIV-related information, and a patient may consent to participate in the HIE by means of a Consent Directive[7] that restricts the sharing of HIV-related information. In this case, the patient's doctor may have access to the patient's clinical document via the HIE, but only after the HIV reducer has been applied. When the doctor derives a plan for the patient and recommends, say, a medication, the HIE takes this plan and subjects it to a safety check against the full patient record (the record before applying the HIV reducer). If a warning is raised, the doctor must decide what to do next – either ask the patient for more information, or ask the patient for more access. Without the safety check the doctor would have had no reason to believe a potential problem with the care plan exists.

3.1 Predicate/Reducer Dichotomy

Predicates and reducers split the redaction task into two parts. The relationship between predicates and reducers can vary: Separating them provides flexibility in developing data segmentation strategies.

For example, a simple predicate that checks for specific coded concepts in a patient's problem list (such as the SNOMED CT[8] concept "Human immunodeficiency virus infection" and its child concepts) would return "True" upon finding any such code; its corresponding reducer could simply remove each such code. In this simple case that there may be no clear benefit to separating predicates from reducers. But more complex predicates might be triggered by *combinations* of clinical facts – such as problems, medications, and procedures – and other factors. So there is not always a straight-forward relationship between the clinical facts that trigger predicates and the clinical facts that might be removed by reducers taking that predicate into consideration.

An expected property of predicates and reducers is the following rule of idempotence: If a reducer is applied to a clinical document that does not trigger its corresponding predicate, then it has no effect. A corollary is that it is possible to derive predicates from reducers by saying that a predicate is true when the reducer is not idempotent on its input. Conversely, reducers can be derived from predicates: A "predicate-derived reducer" might test a clinical document repeatedly against a predicate – trying every possible permutation of redacted clinical facts – and return the clinical document after having applied a particular combination of redactions that satisfies the predicate with the fewest number of clinical facts having been redacted.

For some predicates, the best reducer may be unacceptably broad in scope. For example, the clinical document of an HIV patient with numerous opportunistic infections and other HIV-related conditions may not satisfy an HIV predicate until a large portion of the patient's record is removed. In this case it may be preferable to not return anything at all, as opposed to returning a heavily-redacted record. This

[7] Where the term "Consent Directive" is used in this document, it refers to the HL7 Implementation Guide for CDA® Release 2 Consent Directive – Draft Standard for Trial Use (DSTU), January 2012.
[8] SNOMED CT is a widely used clinical healthcare terminology; http://www.ihtsdo.org/snomed-ct/.

could be a function of a safety checker if its role is expanded beyond a check of the care plan to a check of the redaction: If the redacted document is judged to be too different from the original one then the redaction could be judged unsafe and a warning given.

3.2 Predicate/Reducer as a Service

In the DS2 architecture, predicates and reducers run in a service called the "Predicate/Reducer." The Predicate/Reducer accepts a CCD and a Consent Directive: The CCD is the document that contains a patient's clinical summary; and the Consent Directive is the document that specifies which types of data should be redacted. After processing both documents, the Predicate/Reducer returns a modified CCD that has been acted on by the necessary predicates and reducers, and is appropriately redacted.

The core of the Predicate/Reducer is powered by OpenCDS[9], open source clinical decision support software that implements a subset of the OMG/HL7 Clinical Decision Support Service (CDSS) standard[10] and uses the Drools Rules Engine, a business-logic integration platform. A distinguishing feature of the OpenCDS is its use of the HL7 Virtual Medical Record (vMR), a data model that is optimized for analysis in the clinical decision support context. Addressing the proliferation of mature but incompatible CDS systems, the HL7 vMR seeks to promote interoperable and scalable CDS efforts by establishing a standard information model for clinical inputs and outputs exchanged between CDS systems and clinical information systems[11].

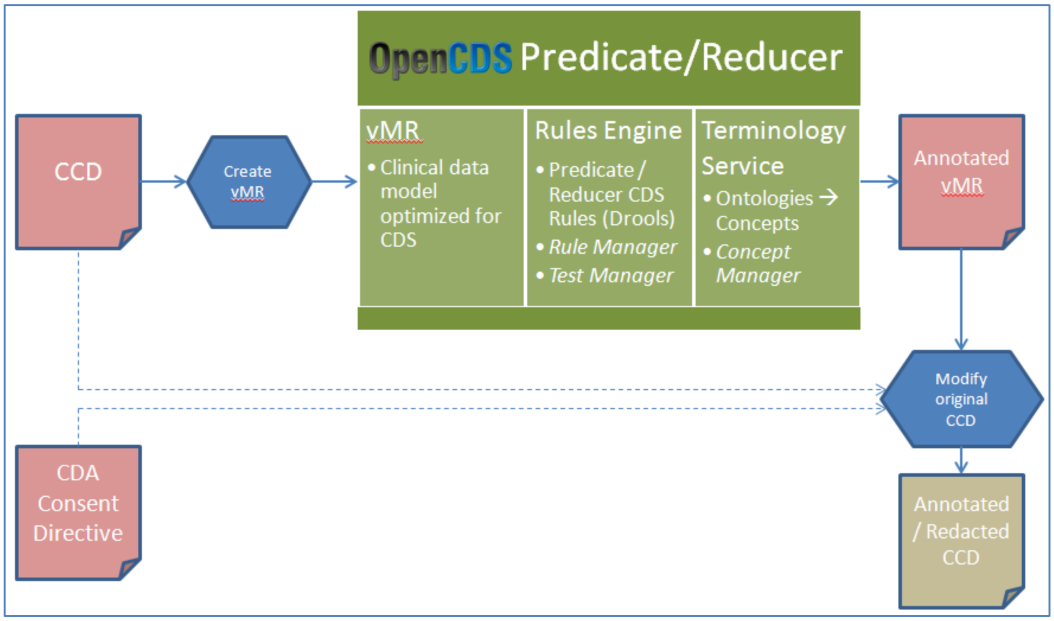


Figure 1: Predicate/Reducer Workflow

[9] OpenCDS is a multi-institutional, collaborative effort to develop open-source, standards-based clinical decision support (CDS) tools and resources. More information about OpenCDS is available at http://www.opencds.org/.
[10] The CDSS specifications are available at http://www.omg.org/spec/CDSS/ and http://www.hl7.org/implement/standards/product_brief.cfm?product_id=12.
[11] Kawamoto, K., et al. (2010). Multi-national, multi-institutional analysis of clinical decision support data needs to inform development of the HL7 virtual medical record standard. In AMIA Annual Symposium Proceedings (Vol. 2010, p. 377). American Medical Informatics Association.

The Predicate/Reducer workflow is illustrated in Figure 1: The patient's CCD is converted into a vMR, which is passed to the OpenCDS Predicate/Reducer. The OpenCDS Predicate/Reducer then executes predicate rules and reducer rules within a Rules Engine architecture that connects the rules to:

- The vMR data model, containing the patient's clinical data; and
- A Terminology Service, providing concepts from vocabularies such as SNOMED CT, LOINC, ICD9/10, CPT, MeSH, NDC, RxNorm, HL7-defined value sets; as well as custom-defined concepts based on these and other ontologies.

In the OpenCDS Predicate/Reducer, the predicate rules and reducer rules are executed; reducer rules insert "reducer tags" in the vMR; and these tags indicate the specific clinical entries in the vMR that would need to be redacted in order to satisfy one or more patient preferences. Finally, the vMR's reducer instructions are compared to the patient's preferences from the CDA Consent Directive, and, if necessary, cross-referenced to the appropriate entries in the original CCD so that those entries can be redacted. The end result is a redacted CCD. As mentioned in the discussion of predicates and reducers, in the case of a heavily redacted CCD, it may be preferable to instead return no record at all.

A primary objective of the architecture is to show how predicate and reducer rules can be implemented and tested in the Predicate/Reducer in a customizable manner. To that end, the Predicate/Reducer architecture includes authoring and testing tools to encourage development and experimentation. The *Rule Manager* is a template-based authoring tool that allows users to create predicate rules and reducer rules using pre-existing templates, and to easily reference concepts from the Terminology Service. Advanced users can develop new templates, or bypass the Rule Manager and work directly with *Guvnor*, the rules authoring environment provided by Drools. The *Test Manager* is a user interface that allows a tester to manage a set of test records and expected outcomes; run predicates and reducers against them and analyze the results. The Test Manager makes it possible for the tester to see which predicate and reducer rules "fired," and in what order. Finally, the *Concept Manager* provides a streamlined user interface to the Terminology Service, which provides access to clinical vocabularies and custom-defined concepts for use by the predicate and reducer rules.

3.3 Types of Predicates

Suppose one wanted to sequester all information in a CCD related to an HIV infection[12]. For simplicity, assume the CCD contains just problems and medications:

Problem List:	**Medication List:**
• HIV infection	• Combivir
• Candidiasis of lung	• Norvir
• Bacterial infection, unspecified	• Procrit
	• Azithromycin
	• Fluconazole

In a Predicate/Reducer system, the process might unfold as follows:

1. An HIV predicate might evaluate the CCD and return "True" (an HIV condition is present in the document).
2. A simple HIV reducer might remove the HIV diagnoses and medications (HIV infection, Combivir, Norvir).
3. The HIV predicate might evaluate the newly redacted CCD and return "True" again because of the co-occurrent clinical facts remaining in the record which might lead an observer to infer that the patient has HIV[13].
4. A more complex HIV reducer might continue to remove co-occurrent clinical facts until the predicate returns "False." Ultimately, in addition to the HIV, Combivir, and Norvir, it might also remove the Candidiasis, Fluconazole, and Procrit – even though none of them are directly related to HIV. Alternatively, it might remove the bacterial infection, Azithromycin, and Procrit.

It is evident from the example above that some clinical facts – such as the HIV diagnosis, and the HIV medications Combivir and Norvir – are directly related to the "target" condition that is the subject of the redaction; and other clinical facts are not directly related, but may still need to be redacted, because an informed observer could potentially infer that patient has HIV based on their presence.

To implement redaction of both types of clinical facts, we rely on two different kinds of predicates: deterministic and probabilistic. Deterministic predicates leverage a knowledge-based approach to clinical decision support technology, whereas probabilistic predicates leverage pattern recognition and machine learning approaches.

[12] For this and other examples that follow, we select HIV as the target condition for sequestration because it is a condition frequently subject to restrictions based on law, policy, or patient preference. Moreover, it is sometimes a challenging condition to redact due to its large number of comorbidities and co-occurrences and yet it is also sometimes relatively easy to redact if patients do not have HIV-related symptoms.

[13] Candidiasis and Fluconazole may suggest an opportunistic infection and an immunocompromised patient. When this is combined with the bacterial infection and antibiotic, the suggestion may be stronger; and Procrit, a medication sometimes used to treat a side effect of the HIV medication, could add further support to the HIV inference.

We define three classes of deterministic predicates:

- **Level 1** deterministic predicates are the least complex predicates. They are written to fire in the presence of obvious concepts – concepts that are known to be equivalent to or closely related to the target condition. For example, if "HIV" or any clinical fact known to indicate or treat HIV is present, then the predicate would return "True."
- **Level 2** deterministic predicates are predicates of moderate complexity. They are written to fire in the presence of correlated concepts, such as comorbidities or co-occurrences, but only if certain conditions are met. For example, the predicate would return "True" when HIV comorbidities are present, but only if the record had a "Level 1" concept to begin with.
- **Level 3** deterministic predicates are predicates of significant complexity based on specific clinical rules. For example, if "HIV" is target, then a level 3 predicate may fire when two or more indirectly related concepts that suggest HIV is (in fact likely to be) present. Or it may fire when a comorbidity is present and is consistent with an HIV-related laboratory result such as a CD4 count within a particular range.

The challenge with deterministic predicates is in striking a balance between redacting too much and too little. If ***no*** co-occurring (*i.e.*, "Level 2") conditions are redacted, the door is left open to inferring a condition that is supposed to be hidden; if ***all*** possible co-occurring conditions are redacted, it is likely some conditions will have been redacted needlessly. For example, recurrent pneumonia and chronic Herpes simplex ulcers are among the 27 AIDS-defining conditions[14], but not all pneumonia and herpes simplex patients have HIV. So, *always* redacting these conditions as a means to hide HIV would likely result in unnecessary redaction of records, especially those belonging to non-HIV patients. An additional challenge is that, in the case of "Level 3" deterministic predicates, their complexity and reliability would be limited by the data quality of the record being evaluated; problem lists in particular are known for being poorly maintained[15].

A probabilistic approach can help with these challenges by applying real-world probabilities and machine learning techniques to develop predicates that learn to understand the density of ties between networked concepts. A Predicate/Reducer system based on such approaches can be optimized to redact the fewest number of clinical facts while still successfully preventing the inference of the targeted condition. For example, in the HIV record discussed above, the probability of inferring an HIV diagnosis after redacting Candidiasis, Fluconazole, and Procrit may be calculated to be lower than the probability of inferring HIV after redacting bacterial infection, Azithromycin, and Procrit – resulting in a decision to redact the former set of facts as opposed to the latter set. But if the redaction of *only* Candidiasis and Fluconazole sufficiently lowers the probability of inference, it may be the preferred choice even if the inference probability after redaction is higher than the other choices. Ideally one would make such decisions in light of the medical significance of the redaction from the perspective of how the record is intended to be used, but this sort of decision making is beyond the scope of the current paper.

[14] CDC MMWR December 5, 2008 / 57(RR10);9, Appendix A: AIDS-Defining Conditions. Available at http://www.cdc.gov/mmwr/preview/mmwrhtml/rr5710a2.htm.

[15] See, for example, Galanter, W. L., Hier, D. B., Jao, C., & Sarne, D. (2010). Computerized physician order entry of medications and clinical decision support can improve problem list documentation compliance. International Journal of Medical Informatics, 79(5), 332-338.

4 Machine Learning Classifiers as Predicates

Machine learning is the application of statistical and probabilistic techniques to create models that can make predictions – for example, to classify a list of symptoms as representative of cancer or not – based on training data. Machine learning classifiers have been used in diagnostic medicine for nearly 30 years, and over 1,500 papers have been published on machine learning and cancer diagnosis alone[16].

We hypothesized that a machine learning classifier – trained to predict a target clinical fact based on a set of known clinical facts – could simulate human inference and act as a substitute for the Level 2 and Level 3 deterministic predicates described in the previous section. In other words, if a classifier predicts a condition that is supposed to be hidden, then perhaps a human would be able to predict it as well.

We limited the scope of the problem to predicting an informed observer's *initial perception* as opposed to predicting the outcome of more iterative processes such as hypothetico-deductive reasoning or patient inquiry. After all, once the observer starts asking questions, even remotely related clinical facts could open the door to revealing a sequestered condition.

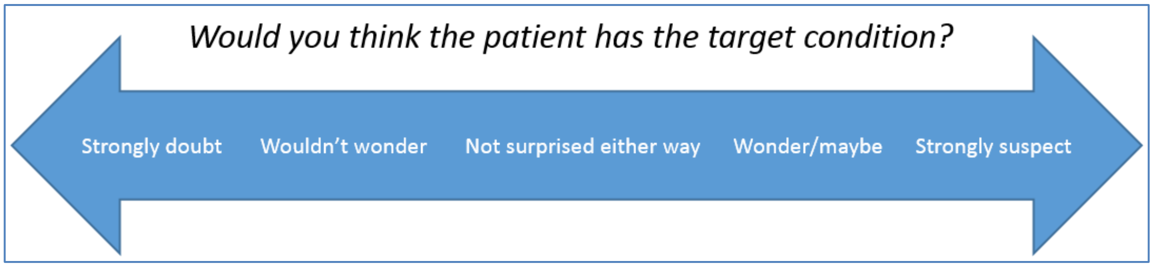
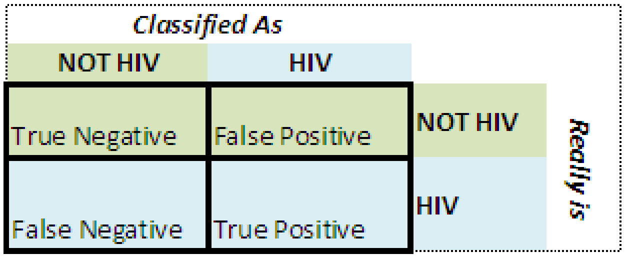


Figure 2: Spectrum of initial perception

The goal, then, is to develop predicates that fire on the right side of the spectrum illustrated in Figure 2, and reducers that move the record just far enough leftward so that the observer would not think that the target clinical fact exists.

Measuring the performance of such predicates is challenging. Predicting whether a patient ***has*** a condition is a different task than predicting whether an informed observer might ***infer*** that the patient has the condition. While we can easily rate a classifier against the former, the latter can only be measured by surveying humans – which was out of scope for this effort. So the traditional metrics of classifier performance were used in the absence of a human survey, based on the premise that the more accurate classifiers would also more closely simulate human perception – up to a point.

[16] Cruz, J. A., & Wishart, D. S. (2006). Applications of machine learning in cancer prediction and prognosis. Cancer Informatics, 2, 59.



	Classified As	
Really is	NOT HIV	HIV
NOT HIV	True Negative	False Positive
HIV	False Negative	True Positive

Figure 3: HIV Classifier Confusion Matrix

Patients who have the target condition but who don't have any distinguishing characteristics in the record (*e.g.*, they are asymptomatic) are False Negatives in the traditional analysis but are actually *True* Negatives as far as human perception is concerned. Patients who do not have the target condition, but exhibit characteristics of patients who do, are False Positives in the traditional analysis but are actually *True* Positives the privacy context. As mentioned, determining the *actual* True/False boundary would fall in the domain of human subject research; in the meantime we have therefore developed a tool to help individuals get a "feel" for where the boundary lies.

4.1 Classifier Development Approach

Our approach to classifier development, evaluation, and integration involved three steps:

1. Train, evaluate, and select candidate classifiers based on the actual presence or absence of the target condition in test data, using WEKA[17] – a widely-used, general purpose data mining tool.
2. Experiment with candidate classifiers in the *Inference Analyzer* – a visual environment custom-developed as part of the DS2 project to present individual patient records and show the results of reducers derived from the classifier-based predicates.
3. Plug classifiers into OpenCDS and the larger Predicate/Reducer architecture shown in Figure 1 in order to use them to help redact conditions from CCDs.

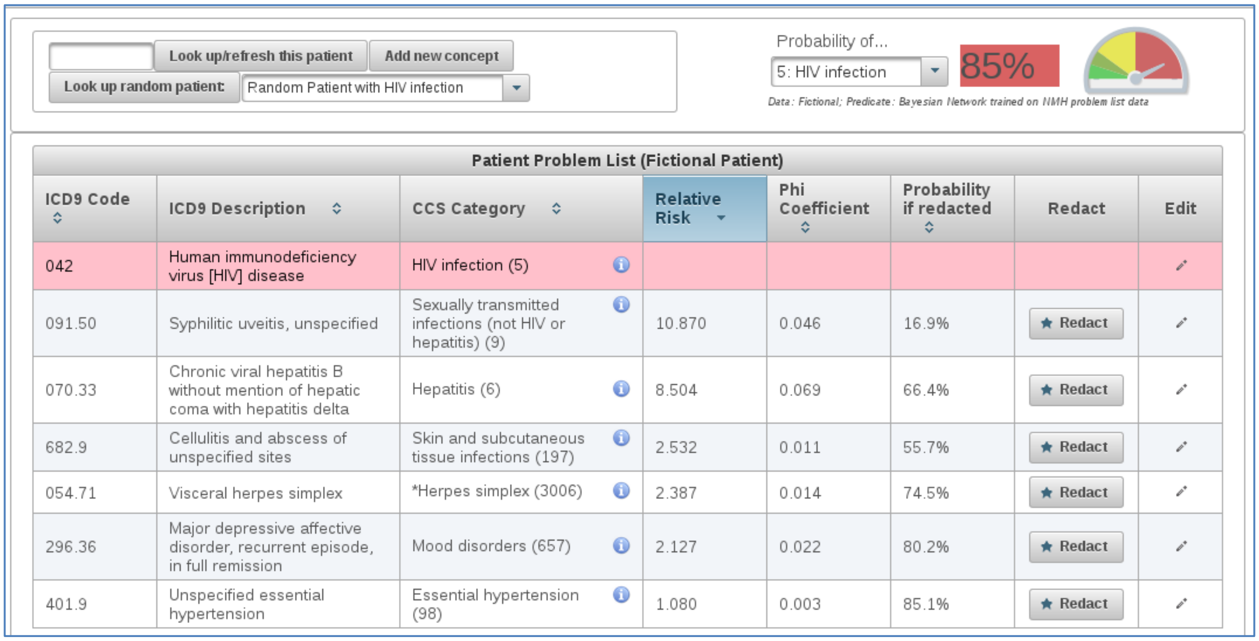
We designed two Application Programming Interfaces (APIs)[18] to connect the classifiers developed in step 1 to the Inference Analyzer and OpenCDS Predicate/Reducer in steps 2 and 3:

- ***SimpleProbabilisticPredicate*** – For classifiers that work on one section of the medical record at a time, this API passes a simple one-dimensional list of clinical facts, such as a list of problem diagnoses or a list of medications, to the classifier.
- ***ProbabilisticPredicate*** – For classifiers that work on the entire patient record, this API passes a vMR object, containing all components of the patient's medical record, to the classifier.

To demonstrate a machine learning-based predicate in our prototype, we used the ***SimpleProbabilisticPredicate*** API and focused on the problem list section, with HIV as the target condition.

[17] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1. Available at http://www.cs.waikato.ac.nz/ml/weka/index.html.

[18] Technical documentation for the probabilistic predicate APIs is available at https://sharps-ds2.atlassian.net/wiki/display/DS2/DS2+Predicate+APIs.



Random Patient with HIV infection

Probability of...

5: HIV infection

85%

Data: Fictional; Predicate: Bayesian Network trained on NMH problem list data

Patient Problem List (Fictional Patient)

ICD9 Code	ICD9 Description	CCS Category	Relative Risk	Phi Coefficient	Probability if redacted	Redact	Edit
042	Human immunodeficiency virus [HIV] disease	HIV infection (5) ⓘ					✎
091.50	Syphilitic uveitis, unspecified	Sexually transmitted infections (not HIV or hepatitis) (9) ⓘ	10.870	0.046	16.9%	★ Redact	✎
070.33	Chronic viral hepatitis B without mention of hepatic coma with hepatitis delta	Hepatitis (6) ⓘ	8.504	0.069	66.4%	★ Redact	✎
682.9	Cellulitis and abscess of unspecified sites	Skin and subcutaneous tissue infections (197) ⓘ	2.532	0.011	55.7%	★ Redact	✎
054.71	Visceral herpes simplex	*Herpes simplex (3006) ⓘ	2.387	0.014	74.5%	★ Redact	✎
296.36	Major depressive affective disorder, recurrent episode, in full remission	Mood disorders (657) ⓘ	2.127	0.022	80.2%	★ Redact	✎
401.9	Unspecified essential hypertension	Essential hypertension (98) ⓘ	1.080	0.003	85.1%	★ Redact	✎

Figure 4: The Inference Analyzer, connected to a Bayesian Network predicate trained on NMH problem list data, showing a fictional HIV patient. The patient's HIV diagnosis has been automatically redacted (therefore it appears highlighted in light red) and is therefore invisible to the predicate. The predicate result appears in the upper-right corner (greater than 50% means "TRUE" for HIV). DS2 is based on the idea that because the predicate predicts the patient has HIV based on all the other conditions, a human observer might think so as well – and therefore additional conditions should be redacted in order to effectively sequester the HIV condition. The tool allows the user to interactively redact additional conditions to observe how the predicate reacts.

4.2 Feature Selection

With tens of thousands of possible diagnoses, a method was needed to combine similar diagnoses and reduce the large number of potential attributes for use in a classifier. For machine learning classifiers, we mapped SNOMED CT and ICD9 codes to Clinical Classification System (CCS)[19] categories to reduce the number of attributes. CCS is a freely available diagnosis and procedure categorization scheme, developed by the US Agency for Healthcare Research and Quality (AHRQ), with approximately 300 clinically meaningful categories. The categories include such conditions as mycoses, HIV infection, viral infections, hepatitis, sexually transmitted diseases other than HIV or hepatitis, various cancers, meningitis, etc.

In order to achieve reasonable training time and accuracy for some classifiers, our reduced set of a few hundred attributes needed to be reduced further by filtering out irrelevant features[20]. We applied a feature selection algorithm, based on information gain of the attribute with respect to the target CCS attribute, and selected the top 50 attributes. For HIV, this resulted in retaining certain correlated categories, such as mycoses and hepatitis, as well as certain negatively correlated categories, such as menopausal disorders. We did not use demographic features in this study, but doing so would make the classifiers more accurate. For instance, HIV is more correlated to men than women, hence negative correlation for menopausal disorders is unsurprising.

[19] See the AHRQ CCS web site at http://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp.

[20] See, for example, Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. Artificial Intelligence, 97(1), 245-271; also, Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. The Journal of Machine Learning Research, 3, 1157-1182.

In our testing we noticed that many conditions that are known to be highly correlated with HIV are intermixed inside CCS categories with conditions that are significantly less correlated (for example Kaposi's Sarcoma is mixed with other types of cancers), and suspected that this was impacting the performance of the classifier for HIV. Based on the list of AIDS-defining conditions[21], we curated a custom set of additional CCS categories, and moved the diagnoses of these conditions from their original CCS categories into the new categories. Our testing demonstrated that this improved classification metrics across all classifiers and data sets[22]. The custom CCS categories are listed below:

- Burkitt's Tumor
- Candidiasis
- Coccidioidomycosis/Isosporiasis
- Cryptococcosis
- Cryptosporidiosis
- Cytomegalovirus
- Encephalopathy
- Herpes simplex
- Histoplasmosis
- Kaposi sarcoma
- Leukoencephalopathy
- Lymphoid interstitial pneumonia
- Mycobacterium
- Pneumocystis
- Salmonella septicemia
- Toxoplasmosis of brain
- Wasting syndrome

Custom-curated CCS categories based on AIDS-defining conditions (cervical cancer, lymphomas, bacterial infection, pneumonia, and tuberculosis already have dedicated categories in CCS and were not included).

4.3 Test Data

Northwestern Memorial Hospital (NMH) made available two de-identified datasets to SHARPS researchers under a confidential data use agreement: An Audit Log dataset (for audit log research) and an EMR dataset. The EMR dataset consisted of encounter diagnoses, medications, procedures, and problem lists for a subset of de-identified NMH patients, and we used this data to train and evaluate classifiers.

The NMH data, due to its data use agreement, could not be used for collaboration with colleagues who were not part of the agreement. So we sought an additional source of test data that could be shared more freely. Between 1996 and 2010, the Centers for Disease Control (CDC) published public use data files from its National Hospital Discharge Survey – an annual probability sample survey of discharges from non-federal, general, and short-stay hospitals. The data is publicly available for download and no license or application is required. We used the discharge diagnoses from these files as a substitute for problem lists to train and evaluate classifiers.

The CDC data, however, is not a perfect substitute for problem lists that might populate an HIE. For example, Berkson's Bias[23] demonstrates that although there may be a correlation between, say, Toxoplasmosis and HIV in hospital discharge diagnoses, this does not necessarily imply that the correlation exists in outpatients or in the population in general. It is important to be aware that primary

[21] CDC MMWR December 5, 2008 / 57(RR10);9, Appendix A: AIDS-Defining Conditions. Available at http://www.cdc.gov/mmwr/preview/mmwrhtml/rr5710a2.htm.

[22] For example, using curated CCS categories instead of standard CCS categories, precision of Bayesian classifiers on problem list data increased by 48-60% at 15% recall; and by 16-25% at 30% recall. Note however that the improvement is diminished at higher recall levels, resulting in similar or lower areas under the ROC curve using curated categories; we explain in Section 4.4 why the focus in DS2 is on the recall levels less than 50%.

[23] Roberts, R. S., Spitzer, W. O., Delmore, T., & Sackett, D. L. (1978). An empirical demonstration of Berkson's bias. Journal of Chronic Diseases, 31(2), 119-128.

care problem lists are different than hospital discharge diagnoses, which are different from outpatient encounter diagnoses and that different correlation statistics should ideally be used for each of those. Furthermore, discharges in the CDC data are weighted by the ratio of a hospital's random sample size to its actual volume of discharges, but we could not apply these weights because doing so would result in artificial duplicates in the test data.

Still, the high-level observations made from predicate experiments performed on hospital discharge data are similar those made from the NMH problem list data (see results below and in Appendix A), even though the correlations on which they are based are not the same.

4.4 Classifier Test Results

Test results of various machine learning classifiers for predicting a hidden HIV diagnosis, using NMH problem list data, are shown in Results Table 1; results using CDC hospital discharge data are shown in Appendix A. We focused on evaluating classifier performance in the 15-30% recall range, based on the informal observations that 1) a significant proportion of HIV patients are asymptomatic for many years after the initial acute infection period; and 2) examination of randomly selected HIV patients using the Inference Analyzer revealed that roughly two-thirds of the patients appeared not to have *any* other features that could conceivably suggest HIV (for example, a patient with only two diagnoses in the problem list: HIV and tobacco use disorder).

As an example, consider the first row in the Results Table 1: In order for the Naïve Bayes classifier to correctly predict HIV – assuming the "Level 1" HIV concepts had already been redacted – for 15% of the patients who have HIV diagnoses in the data set, we would need to accept that the classifier would

Results table 1: Problem List HIV predicates using various machine learning classifiers

- **Dataset: NMH Extended Problem List; 130,415 problem lists, excluding patients with *only* HIV in problem list (about 0.1%, or 12% of NMH HIV patients)**
- **Evaluation: 10-fold cross-validation; hidden attribute to predict: CCS attribute #5 (HIV)**
- **Classifiers: See Appendix B for classifier configuration details; except where "all attributes" is noted, each uses attribute selection of the top 60 custom-curated CCS attributes based on information gain for HIV, against each training fold; classifiers are listed in the table in order of execution time**

Classifier	Execution Time*	Area under ROC Curve (AUC)	False positive rate (%)	Precision (%)	Accuracy (%)	False positive rate (%)	Precision (%)	Accuracy (%)
			@ 30% recall			@ 15% recall		
Naïve Bayes	11s	0.776	3.37	5.76	96.18	0.73	12.37	98.70
Bayes Network	14s	0.774	2.58	7.39	96.97	0.40	20.62	99.03
AODE**	15s	0.780	2.78	6.86	96.76	0.48	17.78	98.95
Naïve Bayes (all attributes)	16s	0.782	3.75	5.18	95.80	0.83	11.03	98.60
AdaBoost	16s	0.583	22.18	1.09	77.54	1.97	5.06	97.47
Radial Basis Function Network	26s	0.703	5.52	3.57	94.04	0.84	10.87	98.59
Decision Tree	49s	0.522	30.03	0.75	69.72	20.03	0.79	79.59
AODE** (all attributes)	155s	0.781	3.02	6.36	96.53	0.54	15.91	98.88
Random Forest	359s	0.666	4.34	4.51	95.22	0.80	11.42	98.63
Support Vector Machine	1292s	0.516	N/A			N/A		
K-Nearest Neighbors***	6021s	0.615	8.34	2.40	91.24	1.81	5.36	97.62

* Combined time to build model (including attribute selection where used) and to test model once on full training set; see Appendix B for computing platform details

** AODE is sometimes referred to as "Not so Naïve Bayes"; see Webb, G. I., Boughton, J., Wang, Z. (2005). "Not So Naive Bayes: Aggregating One-Dependence Estimators". Machine Learning, 58(1), 5–24.

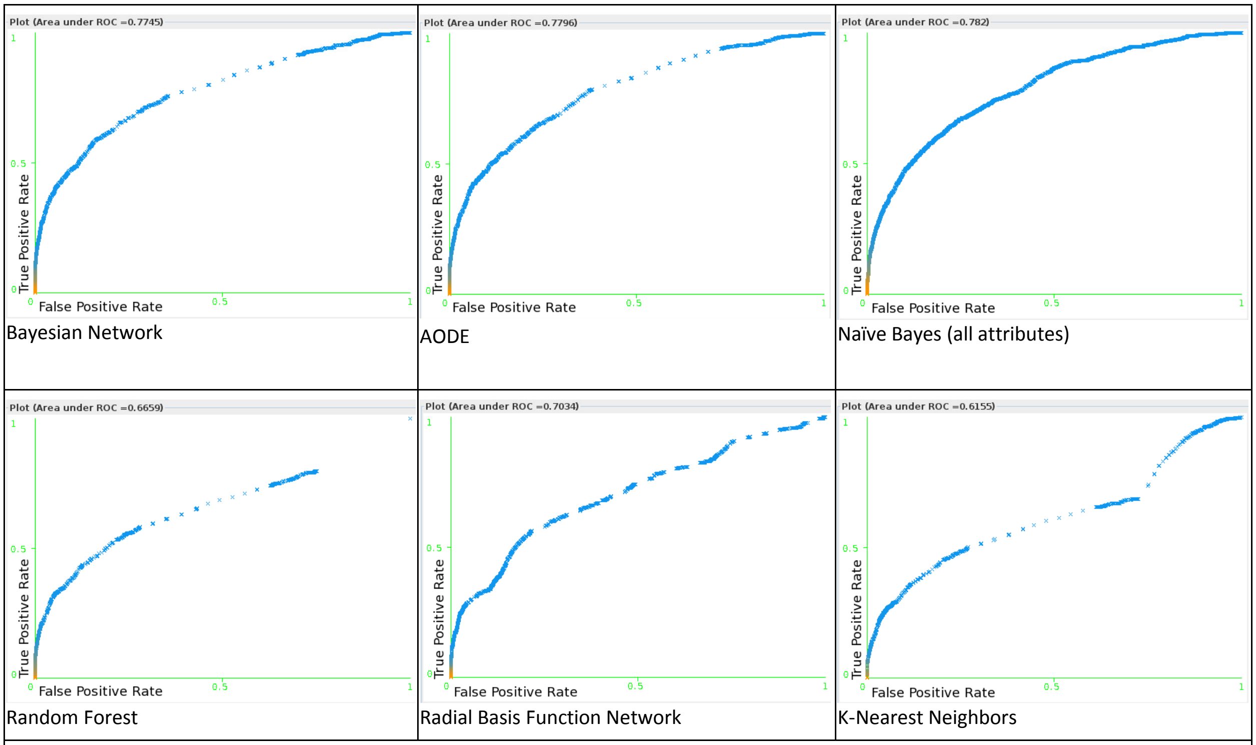
*** K-Nearest Neighbors is a "lazy learner," where computation is deferred until classification time; so while training is fast, classification is significantly slower than other classifiers.

Classifiers ranked by accuracy (at 15% and 30% recall) for prediction of a hidden HIV diagnosis in NMH problem lists

1. Bayesian Network
2. Bayesian Averaged One-Dependence Estimators
3. Naïve Bayes
4. Random Forest
5. Radial Basis Function Network
6. K-Nearest Neighbors
7. AdaBoost
8. Decision Tree

incorrectly predict HIV (that is, predict HIV for patients who actually have no HIV diagnosis) for 0.73% of the entire patient population. Furthermore, of all of the patients with HIV predictions made by the classifier in this case, only 12.37% of them actually have an HIV diagnosis.

The best performer in our testing was the Bayesian Network classifier, which builds and utilizes a directed acyclic graph of attributes – that is, CCS conditions – in order to make decisions based on conditional probabilities between the attributes and their parents in the graph. The Bayesian Network implementation we tested uses a learning algorithm called "K2" to search for the most probable belief network structure in order to build the graph[24], and we found that a maximum of 5 parents per attribute produced the optimal result. Though Bayesian Network does not have the largest area under the ROC curve[25] among the classifiers tested (see ROC curves below), its relative weakness is in parts of the curve that are not relevant to DS2 (greater than 50% recall, *i.e.*, the top half of the ROC curve).



Bayesian Network

AODE

Naïve Bayes (all attributes)

Random Forest

Radial Basis Function Network

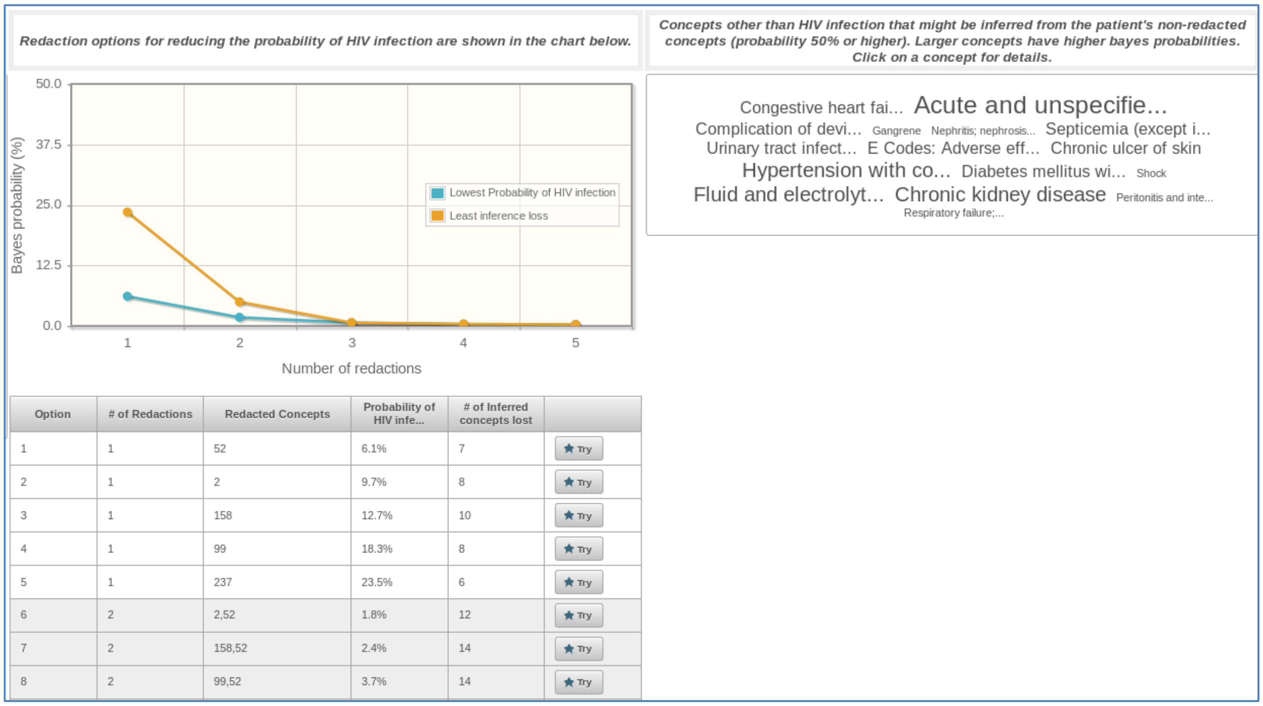
K-Nearest Neighbors

Figure 5: Receiver Operating Characteristic (ROC) curves for six classifiers as tested in Results Table 1.

Classifier threshold indicated by color: 0% 50% 100%

[24] Cooper, G. F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. Machine Learning, 9(4), 309-347.

[25] See, for example, Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition, 30(7), 1145-1159.



Redaction options for reducing the probability of HIV infection are shown in the chart below.

Concepts other than HIV infection that might be inferred from the patient's non-redacted concepts (probability 50% or higher). Larger concepts have higher bayes probabilities. Click on a concept for details.

Option	# of Redactions	Redacted Concepts	Probability of HIV infe...	# of Inferred concepts lost	
1	1	52	6.1%	7	★ Try
2	1	2	9.7%	8	★ Try
3	1	158	12.7%	10	★ Try
4	1	99	18.3%	8	★ Try
5	1	237	23.5%	6	★ Try
6	2	2,52	1.8%	12	★ Try
7	2	158,52	2.4%	14	★ Try
8	2	99,52	3.7%	14	★ Try

Figure 6: Predicate-derived reducer function and inference-loss analysis in Inference Analyzer. The tag cloud to the right shows the conditions that are not in a patient's problem list, but could potentially be inferred based on other problems in the list. Each row in the table at the bottom shows a different predicate-derived reducer – a set of redactions that would satisfy the predicate. The line chart illustrates how different redaction combinations impact the inferred list differently.

Naïve Bayes, though it does not have the best accuracy in the 15-30% recall range, has a number of advantages over the other classifiers: It is very fast, has low storage requirements, and is somewhat robust to irrelevant features (therefore the information-gain feature selection step can be skipped, though the impact on accuracy depends on the data set). For these reasons, it can easily be trained against ***all*** CCS categories as target conditions, as opposed to just selected targets such as HIV or mental health. This in turn made it possible to create a reducer based on "inference loss," where inference loss is defined as the number of ***other*** target CCS categories that would have been predicated had a particular clinical fact not been redacted (the goal being to select the redaction that results in the least amount of inference loss; see Figure 6 above).

Another advantage of Naïve Bayes is that its low computational cost makes it possible to re-train, and incrementally train, against new data. In an HIE environment this feature could be used to train the classifier as new provider participants, with unique data quality and correlation profiles, join the HIE.

Naïve Bayes relies on an independence assumption between attributes, which may intuitively seem to be invalid in the clinical environment, but which works well anyway because of several well-known factors[26]. The other classifiers shown in the results table do not have an independence assumption like Naïve Bayes, and some perform better (though at higher computational cost).

[26] See, for example, Zhang, H. (2004). The optimality of naive Bayes. In Proceedings of the FLAIRS Conference (Vol. 1, No. 2, pp. 3-9). Available at http://www.cs.unb.ca/profs/hzhang/publications/FLAIRS04ZhangH.pdf.

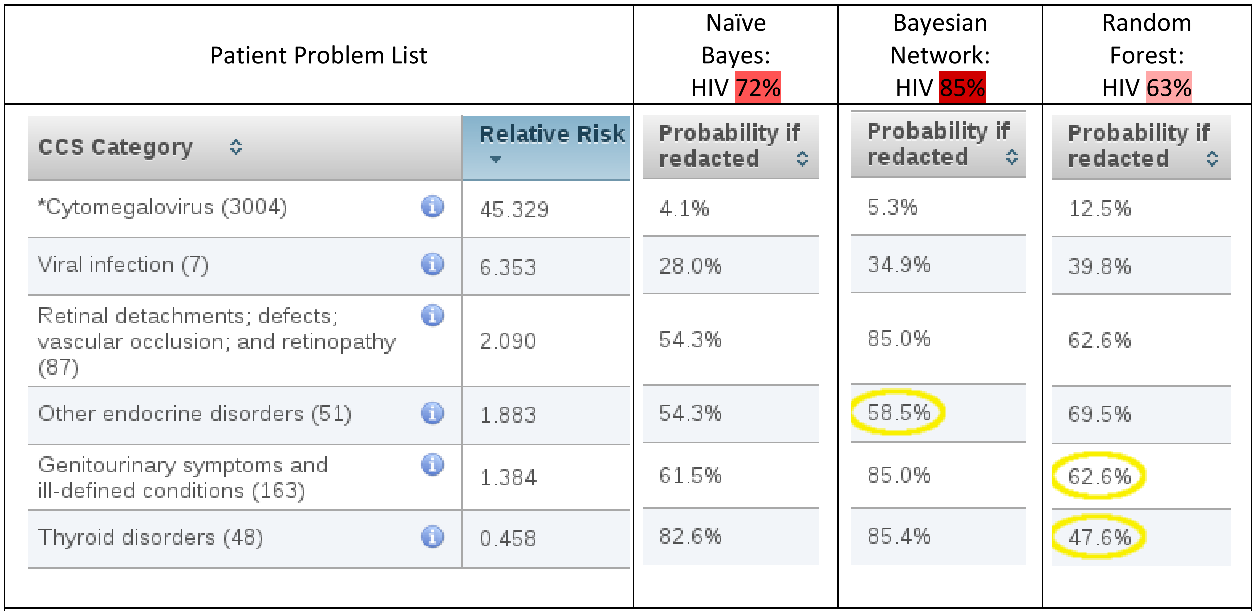
5 Predicate/Reducer Systems in Practice

By testing different types of deterministic and machine learning predicates, with each other and in combination with different reducer strategies, we can make a number of observations regarding the relative strengths of different Predicate/Reducer approaches, and scope out some areas for future research.

5.1 Classifiers and Predicate-derived Redaction

Recall from Section 3.1 that reducers can be derived from predicates: A ***predicate-derived reducer*** tests a clinical document repeatedly against a predicate – trying multiple permutations of redacted clinical facts – and returns the clinical document after having applied a particular combination of redactions that satisfies the predicate.

In addition to its distinguishing characteristics outlined in Section 4, the Naïve Bayes classifier differs from the other classifiers in another important way: Its behavior in predicate-derived redaction. While Naïve Bayes-derived reducers generally redact clinical facts in the order of their correlation to the target condition, classifiers that do not have an independence assumption tend to redact in a less straightforward way. Consider the following example from the Inference Analyzer:



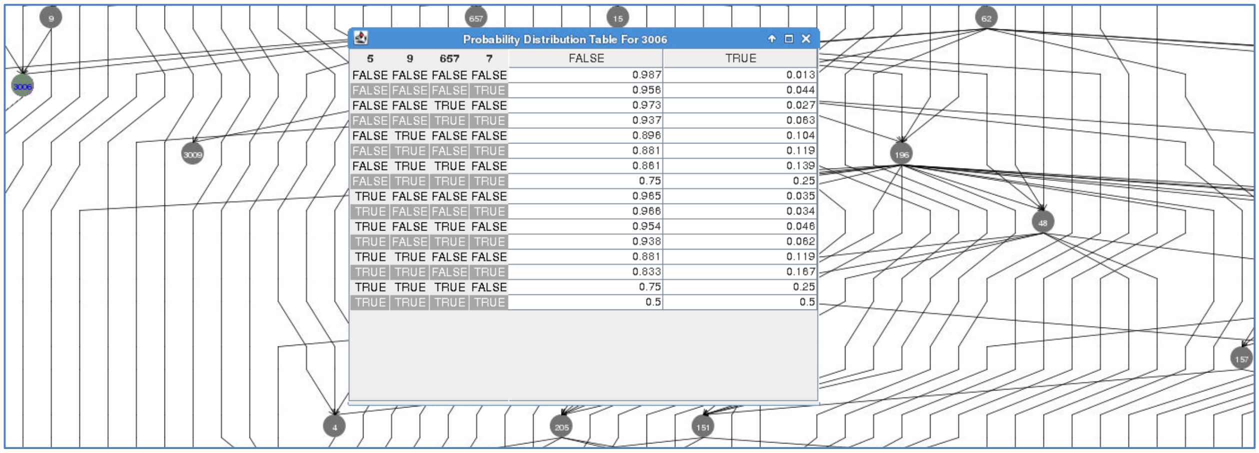
Patient Problem List		Naïve Bayes: HIV 72%	Bayesian Network: HIV 85%	Random Forest: HIV 63%
CCS Category	Relative Risk	Probability if redacted	Probability if redacted	Probability if redacted
*Cytomegalovirus (3004)	45.329	4.1%	5.3%	12.5%
Viral infection (7)	6.353	28.0%	34.9%	39.8%
Retinal detachments; defects; vascular occlusion; and retinopathy (87)	2.090	54.3%	85.0%	62.6%
Other endocrine disorders (51)	1.883	54.3%	58.5%	69.5%
Genitourinary symptoms and ill-defined conditions (163)	1.384	61.5%	85.0%	62.6%
Thyroid disorders (48)	0.458	82.6%	85.4%	47.6%

Figure 7: A fictional patient's problem list as shown in the Inference Analyzer, with six conditions, evaluated by three NMH-trained HIV classifiers, showing what the scores would be if each problem were redacted. The circled scores indicate conditions in which the classifiers' score after redaction is not in ascending order along with the other conditions.

The problems in the patient's problem list are shown in descending order based on their correlation with HIV (as measured by relative risk in the data set). Notice how the Naïve Bayes scores associated with redactions are in *ascending* order (4.1, 28.0, 54.4, 54.3, 61.5, and 82.6): This demonstrates that redacting a more highly correlated condition will result in a lower Naïve Bayes score. But the Bayesian Network and Random Forest redactions are ***not*** always in ascending order. In the example above, redacting an endocrine disorder will result in a lower Bayesian Network score compared to redacting a retinal detachment, despite the fact that the endocrine disorder is less correlated with HIV in the data set.

Also in the example above, thyroid disorder is negatively correlated with HIV, and yet the HIV prediction of the Random Forest classifier actually decreases when it is redacted – the opposite behavior that one might expect based on the correlation.

In the Bayesian Network, one explanation for this behavior is that redaction of a single attribute can potentially impact the conditional probability-based computation of all of its child attributes in the network (see Figure 8 below).



Probability Distribution Table For 3006

5	9	657	7	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE	0.987	0.013
FALSE	FALSE	FALSE	TRUE	0.956	0.044
FALSE	FALSE	TRUE	FALSE	0.973	0.027
FALSE	FALSE	TRUE	TRUE	0.937	0.063
FALSE	TRUE	FALSE	FALSE	0.896	0.104
FALSE	TRUE	FALSE	TRUE	0.881	0.119
FALSE	TRUE	TRUE	FALSE	0.861	0.139
FALSE	TRUE	TRUE	TRUE	0.75	0.25
TRUE	FALSE	FALSE	FALSE	0.965	0.035
TRUE	FALSE	FALSE	TRUE	0.966	0.034
TRUE	FALSE	TRUE	FALSE	0.954	0.046
TRUE	FALSE	TRUE	TRUE	0.938	0.062
TRUE	TRUE	FALSE	FALSE	0.881	0.119
TRUE	TRUE	FALSE	TRUE	0.833	0.167
TRUE	TRUE	TRUE	FALSE	0.75	0.25
TRUE	TRUE	TRUE	TRUE	0.5	0.5

Figure 8: Subset of Bayesian Network utilized by NMH problem list classifier showing conditional probabilities for CCS condition 3006 (herpes simplex) and its parents in the network: CCS 5 (HIV), 9 (STDs), 657 (mood disorders), and 7 (viral infections).

Similarly, in decision tree-based classifiers such as the Random Forest, redaction can impact decisions made at relatively high levels in its trees that in turn affect the impact that other conditional attributes may have had on the overall decision.

A potential issue with redaction in the context of these types of predicates (that is, predicates based on classifiers with a dependence assumption) is that redaction could create instances of patient records that are “un-natural” – patients who have combinations of attributes that the classifiers have not been trained on because they do not occur in the real world. In this case it is conceivable that such a predicate, after a redaction, could return “FALSE,” but a human who is aware that the record may have been redacted could still infer the target condition.

A Bayesian network-based Predicate/Reducer that takes in to account the structure of the network could take steps to mitigate this issue, such as redacting all of an attribute’s children whenever an attribute is redacted.

5.2 Combining Deterministic and Probabilistic Predicates

We have demonstrated how the two types of predicates – deterministic predicates which leverage a knowledge-based approach, and probabilistic predicates which leverage machine learning approaches – have advantages and disadvantages when applied in a DS4P context. Deterministic predicates are predictable and can be defined to precisely implement written laws or policies; probabilistic predicates make it possible to predict inferences and may be less difficult to maintain.

Deterministic and probabilistic predicates can also be *combined* for fine-grained control over redaction thresholds and error rates.

Looking back at Results Table 1, we can see that to correctly predict HIV for 15% of the patients who have HIV diagnoses, using a Bayesian network predicate, we would need to accept false positives for 0.40% of the patient population. This means that 0.40% of patients for whom HIV should be redacted but who have no HIV diagnosis will have at least one clinical fact redacted from their record as it passes through the Predicate/Reducer. Whether or not this would be considered acceptable depends on a number of factors, including:

- Whether HIV redaction is being performed for all patients, or only the patients who have expressed a preference for redacting HIV
- Whether patients can see and approve redactions before they are performed
- Whether providers are aware that redactions may have been made
- Whether HIV redaction is being performed for all patients, or only the patients who actually have an HIV diagnosis
- The likelihood that HIV may have *already* been redacted or otherwise excluded from the patient record before being assessed by the predicate

Alternatively, instead of accepting a 0.40% false positive rate for the entire patient population, the classifier threshold could be dynamically adjusted depending on clinical factors. For example, a deterministic predicate could apply Level 1 rules and then call a probabilistic predicate with a threshold of 50% for patients without HIV diagnoses; but lower the threshold to 30% for patients with HIV diagnoses. This would result in more aggressive redaction for known HIV patients compared to others.

Another approach would be to apply Level 1, 2, and 3 deterministic rules to all patients but apply the probabilistic predicate only to patients that are known to have HIV as a “double check” to see if the deterministic rules caught enough.

5.3 Comparing Reducers

Using the Inference Analyzer, we simulated a deterministic-probabilistic reducer[27] and a deterministic-only reducer[28] against a random sample of 150 NMH problem lists with HIV, and obtained the following results:

- 78% of HIV patients had no redactions beyond Level 1 (*i.e.*, the HIV itself) with either reducer. These were patients with HIV and with other conditions not highly correlated with HIV and not among the AIDS-defining conditions: For example, a patient with HIV, depression, and high cholesterol.
- 7% of HIV patients had redactions (beyond Level 1) that were the same with both reducers. These were HIV patients with highly correlated comorbidities that were also among the official

[27] The deterministic-probabilistic reducer eliminated Level 1 HIV concepts (CCS category #5, HIV), and further eliminated concepts in order to satisfy a Bayesian network predicate set to a 7% score threshold (18% recall in the test data) for patients with HIV in the problem list, and a 50% score threshold (7% recall in the test data) for all other patients. For the simulation, the NMH data was split so that 66% was used for training and 34% was used in the Inference Analyzer.

[28] The deterministic reducer eliminated Level 1 HIV concepts (CCS category #5, HIV), and further eliminated Level 2 HIV concepts (AIDS-defining conditions, see Section 4.2) if a Level 1 HIV concept were present.

AIDS-defining conditions, such as Kaposi sarcoma, pneumocystis, or cytomegalovirus[29]; all of which were redacted by both reducers.

- 11% of HIV patients were more heavily redacted in the deterministic-probabilistic reducer. This was due to the redaction of one or more conditions moderately correlated with HIV but *not* one of the official AIDS-defining conditions. For example, a patient with HIV, leukoencephalopathy, hepatitis C, and haemophilia, would have had HIV and leukoencephalopathy redacted by both reducers (as leukoencephalopathy is highly correlated with HIV[30] and an AIDS-defining condition), but the deterministic-probabilistic predicate *also* required the redaction of at least one of the other two conditions, both of which were moderately correlated with HIV[31].
- 4% of HIV patients were more heavily redacted in the deterministic reducer. This occurred in cases where an AIDS-defining condition had a relatively low correlation with HIV, such as herpes simplex[32], and therefore was not redacted by the deterministic-probabilistic reducer but *was* redacted by the deterministic reducer based on its status as an AIDS-defining condition.

The deterministic-probabilistic reducer, with its score threshold set to 50%, would have also resulted in redaction for 0.05% of patients who did *not* have HIV in their problem list. These were patients without HIV in their problem list, but with correlated conditions, AIDS-defining conditions, or both; they could have been patients with undiagnosed HIV; or patients *with* HIV but for whom the HIV was not included in the problem list (either intentionally or unintentionally). Using the Inference Analyzer, we examined random patients among this “false positive” set, and found that, for example, patients with problem lists containing

1. hepatitis, haemophilia, and lymphadenitis or
2. cytomegalovirus and viral warts or
3. hepatitis, mycoses, nutritional deficiencies, and endocrine disorders

would have at least one condition redacted by the deterministic-probabilistic reducer. The deterministic reducer would redact none of these in the absence of HIV.

5.4 Additional Conditions, Sections, and Uses beyond DS4P

In previous sections we focused on the problem list section, with HIV as the target condition. However, the same approach can be applied to other sections of the medical record or other conditions.

Whether or not a section can be evaluated independently of other sections depends partly on the degree of co-dependence between sections.

[29] In the data set used to train the Bayesian network probabilistic predicate (66% of NMH Extended Problem List), relative risks for HIV given Kaposi sarcoma, pneumocystis and cytomegalovirus were 86.4, 70.8, and 45.3, respectively (i.e., problem lists with Kaposi sarcoma were 86.4 times as likely to have HIV compared to problem lists without Kaposi sarcoma).

[30] In the training set, relative risk for HIV given leukoencephalopathy was 129.3

[31] In the training set, relative risks for HIV given hepatitis and coagulation disorders were 8.5 and 5.3, respectively.

[32] In the training set, relative risk for HIV given herpes simplex was 2.3.

Medication	"May Treat" (converted to DS2-curated CCS category)
Combivir	HIV infection
Norvir	HIV infection
Procrit	Deficiency and other anemia
Azithromycin	Otitis media and related conditions, Urinary tract infections, Bacterial infections, Other lower respiratory disease
Fluconazole	Mycoses, Mycobacterium, Coccidioidomycosis, Candidiasis, Cryptococcosis

Medications, for example, are often prescribed to treat diagnosed conditions, so running a predicate independently on a diagnosis list and a medication list could result in a condition being redacted in one section, but the medication to treat it being *retained* in the other section. To build a Predicate/Reducer system that works with these two sections together, we utilized NDF-RT[33], a publicly available drug database. For medications in its database, NDF-RT provides lists of conditions – SNOMED CT concepts – that the medications are known to treat. We mapped these SNOMED CT concepts to the CCS categories used in DS2's problem list predicates (some examples are shown in the table above); and then experimented with two different Predicate/Reducer approaches: A deterministic approach that redacts any medication with a "May Treat" condition matching a condition redacted in the diagnosis list; and a probabilistic approach that combines problems and medications' "May Treat" conditions into a single attribute space for a machine learning classifier to work with.

Using the Inference Analyzer with a Naïve Bayes predicate, we also experimented with classifier performance in areas besides HIV. This was relatively easy to do because, as explained in Section 4.4, the Naïve Bayes predicate can be easily trained against ***all*** CCS categories as target conditions, as opposed to just selected targets such as HIV or mental health.

Hemorrhoids (0.877)
Acute and unspecified renal failure (0.877)
Respiratory failure; insufficiency; arrest (adult) (0.883)
Nervous system congenital anomalies (0.883)
Cancer of liver and intrahepatic bile duct (0.884)
Cancer of bronchus; lung (0.885)
Septicemia (except in labor) (0.885)
Alcohol-related disorders (0.885)
Cancer of pancreas (0.886)
Cancer of bone and connective tissue (0.886)
Aspiration pneumonitis; food/vomitus (0.888)
Attention-deficit, conduct, and disruptive behavior disorders (0.894)
Acute and chronic tonsillitis (0.904)
Paralysis (0.904)
Cystic fibrosis (0.905)
Cardiac arrest and ventricular fibrillation (0.906)
Nephritis; nephrosis; renal sclerosis (0.906)
Disorders usually diagnosed in infancy, childhood, or adolescence (0.908)
Peritonitis and intestinal abscess (0.909)
Infective arthritis & osteomyelitis (except caused by TB or STD) (0.911)
Appendicitis and other appendiceal conditions (0.913)
Medical examination/evaluation (0.913)
Malignant neoplasm without specification of site (0.913)
Intracranial injury (0.917)
Benign neoplasm of uterus (0.929)
Secondary malignancies (0.933)
Prolapse of female genital organs (0.937)
Impulse control disorders, NEC (0.937)
Gangrene (0.939)
Personality disorders (0.944)
Shock (0.955)
Menstrual disorders (0.957)
Chronic kidney disease (0.963)
Maintenance chemotherapy; radiotherapy (0.967)

Figure 9: CCS Diagnosis categories with highest Naïve Bayes Area under ROC Curve (shown in parenthesis), excluding injuries, poisoning, external causes of injury and poisoning, and labor & delivery (CDC Hospital discharge data 2010, 10-fold cross validation, all attributes).

[33] "May Treat" data is obtained from the National Drug File - Reference Terminology (NDF-RT), produced by the U.S. Department of Veterans Affairs, Veterans Health Administration (VHA), and made available to the general public via API at http://rxnav.nlm.nih.gov/NdfrtAPIs.html. Normalized names for clinical drugs and links to NDF-RT are provided by RxNorm via its public API at http://rxnav.nlm.nih.gov/RxNormAPIs.htm.

For target conditions with high classifier performance, Predicate/Reducer technology could potentially be used for privacy redaction or for other purposes such as:

- Providing a ***clinical summary evaluation tool*** to suggest possible "missing" diagnoses on patient records for quality review or for point-of-care clinical decision support.
- Providing an ***inverse reducer*** that applies a Predicate/Reducer to filter a record to retain *only* the conditions related to a target condition.

For example, a public health cancer registry, or a research study, may be interested in using an inverse reducer to obtain diagnoses and medications related to cancer but ***not*** other conditions. An inverse reducer might also be used to assist physicians when generating referrals to specialists in order to highlight subsets of the clinical summary most relevant to a particular specialty or referral reason.

5.5 Challenges and Opportunities

Applying Predicate/Reducer systems to test data sets illustrated a number of challenges and opportunities for future work. First, though the Predicate/Reducer architecture is capable of handling unstructured text, the prototype predicates and reducers used in the DS2 project only consider ***structured*** data in clinical summary documents, therefore requiring all narrative text to be redacted. And although clinical document standards such as CCD make it possible to link narrative text with corresponding clinical facts, such linking is not always utilized, nor does it constrain the text from introducing additional clinical facts. Natural Language Processing (NLP) applications have been developed to extract and classify clinical facts from narrative text[34] and could be applied to the DS2 model.

NLP approaches, like all probabilistic predicates, face the challenge that probability-based segmentation is inherently unpredictable from the point of view of the patient or provider using the system. Compared to deterministic approaches that are predictable and repeatable, probability-based approaches that redact based on context and training data may behave differently depending on context and may change over time. This kind of behavior illustrates the complexity of deploying data segmentation as a privacy-protective strategy in the context of electronic health information exchange. Moreover, it may prove challenging to explain such complex behaviors to patients, providers, and other stakeholders. For these reasons, we envision privacy-protective deployments of DS2—and DS4P more generally—as operating within communities of practice, organizations, and policy environments designed to foster patient trust and preserve the contextual integrity of their health information[35].

As described in Section 4.4, a challenge in applying machine learning classifiers to an HIE environment is that new provider participants, with unique data quality and correlation profiles, join the HIE on a routine basis, and may alter the effectiveness of a classifier trained on prior data; thereby requiring re-training. An additional challenge is that the ability of observers to infer specific conditions based on clinical context may vary across different types of providers, levels of training, and medical specialities; it

[34] There are many examples in the literature; one recent example is: Ye, Y., Tsui, F. R., Wagner, M., Espino, J. U., & Li, Q. (2014). Influenza detection from emergency department reports using natural language processing and Bayesian network classifiers. Journal of the American Medical Informatics Association.

[35] See our companion paper, French, M., et al. (2014). Decision Support for Data Segmentation (DS2): Contextual Integrity Considerations. Also see Barth, A., Datta, A., Mitchell, J. C., & Nissenbaum, H. (2006). Privacy and contextual integrity: Framework and applications. Proceedings of 27th IEEE Symposium on Security and Privacy, May, 2006.

may therefore may require that different predicates be applied to different domains. Furthermore, an observer may have access to additional context ***not*** evaluated by a predicate, such as a medical record obtained prior to, or after, the predicate evaluation.

Finally, data quality was mentioned in Section 3.3 as a challenge for Level 3 deterministic predicates, and it poses a challenge to probabilistic predicates as well. We found that, for example, the size and scope of medication lists varies widely by patient and that this affected our ability to train classifiers on that data.

6 Adding Predicates, Reducers, and Safety Checks to an HIE

The ILHIE-SHARPS Prototype is an implementation of the DS2 architecture as a collaborative effort with the Illinois HIE. It places the Predicate/Reducer into a larger proxy server-based system designed to enhance an existing HIE.

6.1 Prototype Architecture

When installed in front of an HIE's XDS Registry/Repository service[36], and configured to intercept its inbound XDS requests, the prototype queries for a patient consent directive in the repository, and sends it – along with the CCD returned by the HIE – to the Predicate/Reducer. The prototype then returns the redacted CCD to the requester, as illustrated in Figure 10.

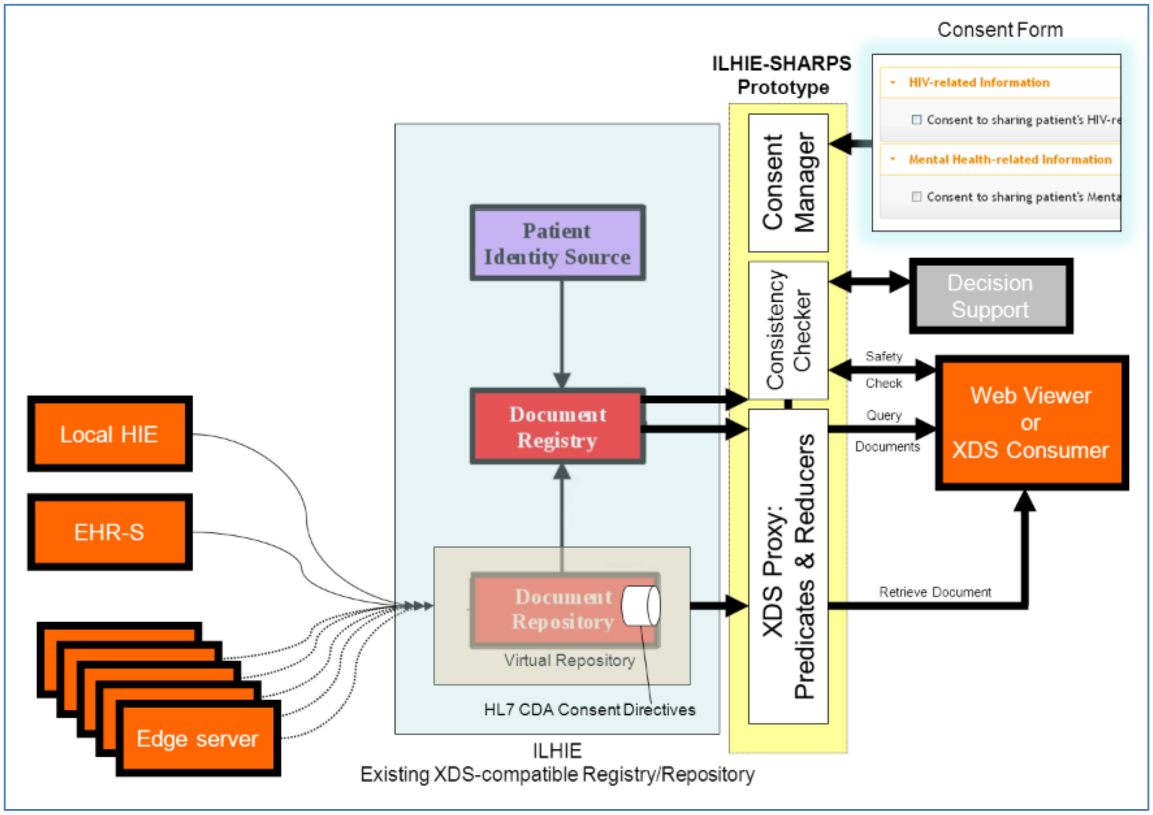


Figure 10: Prototype HIE Architecture

[36] XDS, or Cross-Enterprise Document Sharing, is an integration profile in the Integrating the Healthcare Enterprise (IHE) IT Infrastructure Technical Framework, http://www.ihe.net/Technical_Framework/index.cfm#IT.

The overall HIE architecture, which is common among state HIEs and implemented by a number of commercial vendors, involves provider EHRs connected to a central HIE registry, directly or via proprietary “edge servers.” The HIE maintains a document registry of patient CCDs, either stored in a central repository or in federated edge servers, and responds to requests for documents from “consumers” (typically clinicians who are participants in the HIE) running XDS client software in an EHR or using a web-based XDS viewer.

After the redacted CCD is returned by the prototype, the requester – knowing that the CCD may have been redacted – can initiate a safety check. Drug-drug interaction safety checking, part of a more generic idea called “Consistency Checking,” is designed to help mitigate the potential negative impact of redacted data on patient care. The prototype drug interaction safety checking service is a web-based application that responds (with a true or false result) to requests to compare a proposed medication list with all known interactions[37] with medications in the non-redacted CCD[38]. The prototype also includes a simple web application to create patient consent directive documents to be used by the predicate reducer.

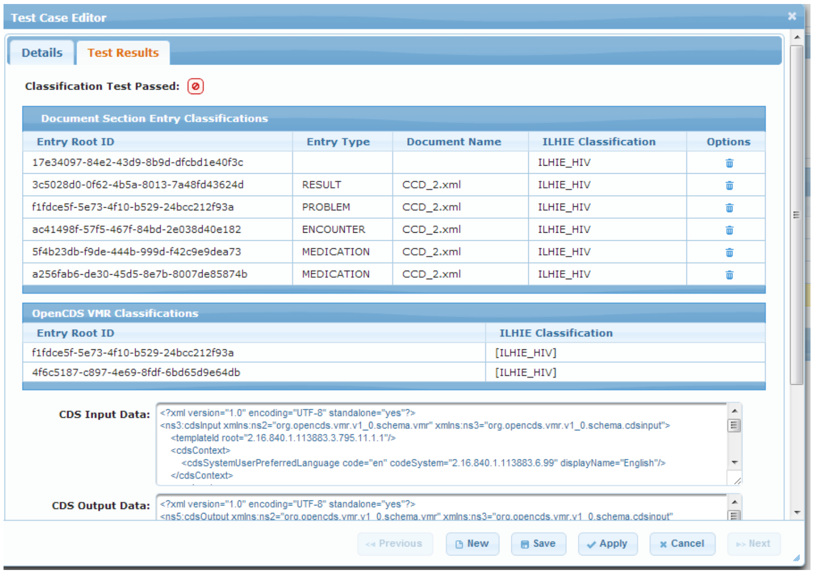
For each of the three prototype components – Predicate/Reducer proxy, Safety Checker, and Consent Form – the scope of the prototype is limited for demonstration purposes but the architecture is extensible so that the scope can be expanded in the future. For example, the safety checking is limited to medications but could potentially cover additional clinical domains beyond medications. Consent directives supported by the prototype are relatively simple and are limited to indicate a preference for disclosure or non-disclosure of specific conditions such as HIV and mental health information.

[37] Drug interactions are based on information from the National Drug File - Reference Terminology (NDF-RT), produced by the U.S. Department of Veterans Affairs, Veterans Health Administration (VHA), and made available to the general public via API at http://rxnav.nlm.nih.gov/NdfrtAPIs.html. Normalized names for clinical drugs and links to NDF-RT are provided by RxNorm via its public API at http://rxnav.nlm.nih.gov/RxNormAPIs.htm.

[38] Although it would be possible for a doctor to falsely propose medications to attempt to discover a redacted sensitive condition, the function still provides security against what cryptographers call “honest but curious attackers,” such as a doctor who might look at a sensitive portion of a record if he has it, but who will not do anything exceptional to find out about sensitive values.

6.2 Testing Architecture

As explained in Section 3.2, the *Test Manager* is a user interface that allows a tester to manage a set of test records and expected outcomes; run predicates and reducers against them; and analyze the results. While the Inference Analyzer takes the user on a deep dive into a narrow domain (via the ***SimpleProbabilisticPredicate*** API), the Test Manager is a higher-level tool that allows the user to manage entire CCDs and run predicates and reducers against them (leveraging the CCD-to-vMR converter). In addition to a test tool, the Test Manager is a template-based CDA editor that can create, edit, import, and export CCDs.



Classification Test Passed: ⊘

Document Section Entry Classifications

Entry Root ID	Entry Type	Document Name	ILHIE Classification	Options
17e34097-84e2-43d9-8b9d-dfcbd1e40f3c			ILHIE_HIV	🗑
3c5028d0-0f62-4b5a-8013-7a48fd43624d	RESULT	CCD_2.xml	ILHIE_HIV	🗑
f1fdce5f-5e73-4f10-b529-24bcc212f93a	PROBLEM	CCD_2.xml	ILHIE_HIV	🗑
ac41498f-57f5-467f-84bd-2e038d40e182	ENCOUNTER	CCD_2.xml	ILHIE_HIV	🗑
5f4b23db-f9de-444b-999d-f42c9e9dea73	MEDICATION	CCD_2.xml	ILHIE_HIV	🗑
a256fab6-de30-45d5-8e7b-8007de85874b	MEDICATION	CCD_2.xml	ILHIE_HIV	🗑

OpenCDS VMR Classifications

Entry Root ID	ILHIE Classification
f1fdce5f-5e73-4f10-b529-24bcc212f93a	[ILHIE_HIV]
4f6c5187-c897-4e69-8fdf-6bd65d9e64db	[ILHIE_HIV]

CDS Input Data:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:cdsInput xmlns:ns2="org.opencds.vmr.v1_0.schema.vmr" xmlns:ns3="org.opencds.vmr.v1_0.schema.cdsinput">
  <templateId root="2.16.840.1.113883.3.795.11.1.1"/>
  <cdsContext>
    <cdsSystemUserPreferredLanguage code="en" codeSystem="2.16.840.1.113883.6.99" displayName="English"/>
  </cdsContext>
```

CDS Output Data:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns5:cdsOutput xmlns:ns2="org.opencds.vmr.v1_0.schema.vmr" xmlns:ns3="org.opencds.vmr.v1_0.schema.cdsinput"
```

Figure 11: Test Manager

6.3 Open Source Software

All of the prototype software developed for the DS2 project is publicly available on the Internet under an open source license[39]. The source code is hosted in a Distributed Version Control System (DVCS) at https://bitbucket.org/sharps-ds2; and project documentation and related materials are hosted on a Wiki at https://sharps-ds2.atlassian.net[40].

It is important to note that DS2 is prototype software, intended to demonstrate the DS2 architecture. Visitors to the above web sites who are seeking pilot or production software related to Data Segmentation for Privacy are encouraged to visit the DS4P Pilot Projects[41]. As noted in the introduction, DS4P is a very broad topic; by contrast, DS2 focuses on a few niche areas that we hope will contribute to existing and future DS4P research and development, as well as other work related to HIE and CDS.

[39] DS2 prototype software is licensed under the "BSD 3-Clause License," available at http://opensource.org/licenses/BSD-3-Clause.

[40] Both SHARPS DS2 web sites are hosted by Atlassian, Inc. (http://www.atlassian.net); and as designated open source projects, they are hosted free of charge.

[41] DS4P Pilot Projects are available at http://wiki.siframework.org/Data+Segmentation+for+Privacy+RI+and+Pilots+Sub-Workgroup.

7 Conclusion and Future Work

In collaboration with the Illinois Health Information Exchange (ILHIE), this project defined a technical architecture and developed an open source prototype to leverage OpenCDS, a popular Clinical Decision Support (CDS) framework, for the identification and sequestration of certain types of sensitive information from patient records flowing through an HIE. The project adopted the name “DS2” – Decision Support for Data Segmentation – because of its unique focus on the ability to detect clinical facts that may imply a sensitive condition, in addition to detecting clinical facts that are directly related to the condition.

The project demonstrated that the redaction of a condition and its related clinical facts sometimes leaves residual facts that, through clinical inference, can still reveal the redacted condition. To address this challenge, OpenCDS deterministic rules were combined with Bayesian and other machine learning classifiers to redact targeted conditions along with certain co-occurrences and comorbidities.

Key contributions include the technical architecture and prototype, along with the following: A suite of related open source software tools for creating, manipulating, converting, and testing standards-based clinical documents (CCDs and vMRs); a methodology for developing and implementing deterministic and probabilistic predicates; test results on a variety of machine learning techniques; a web-based “inference analyzer” for visualizing the effectiveness and the impact of predicates and reducers; a “safety checker” for evaluating drug-drug interactions against non-redacted CCDs; and improvements to web-based software used with OpenCDS.

The scope of the prototype is limited for demonstration purposes but the architecture is extensible so that the scope can be expanded in the future. Potential areas for future work include human subject research to compare predicate reducer results with human perception (both patient and provider); reducer research to identify classifier-specific weaknesses in predicate-derived reducers; and expansion to other conditions, other sections of the medical record, and other uses beyond privacy such as public health, clinical research, and clinical decision making.

8 Appendix A: Additional Classifier Testing Results

Section 4.4 shows the classifier test results against NMH problem list data. Below are results against CDC hospital discharge data; as well as an annotated ROC curve (Figure 13).

Results table 2: Hospital Discharge diagnosis list HIV predicates using various machine learning classifiers

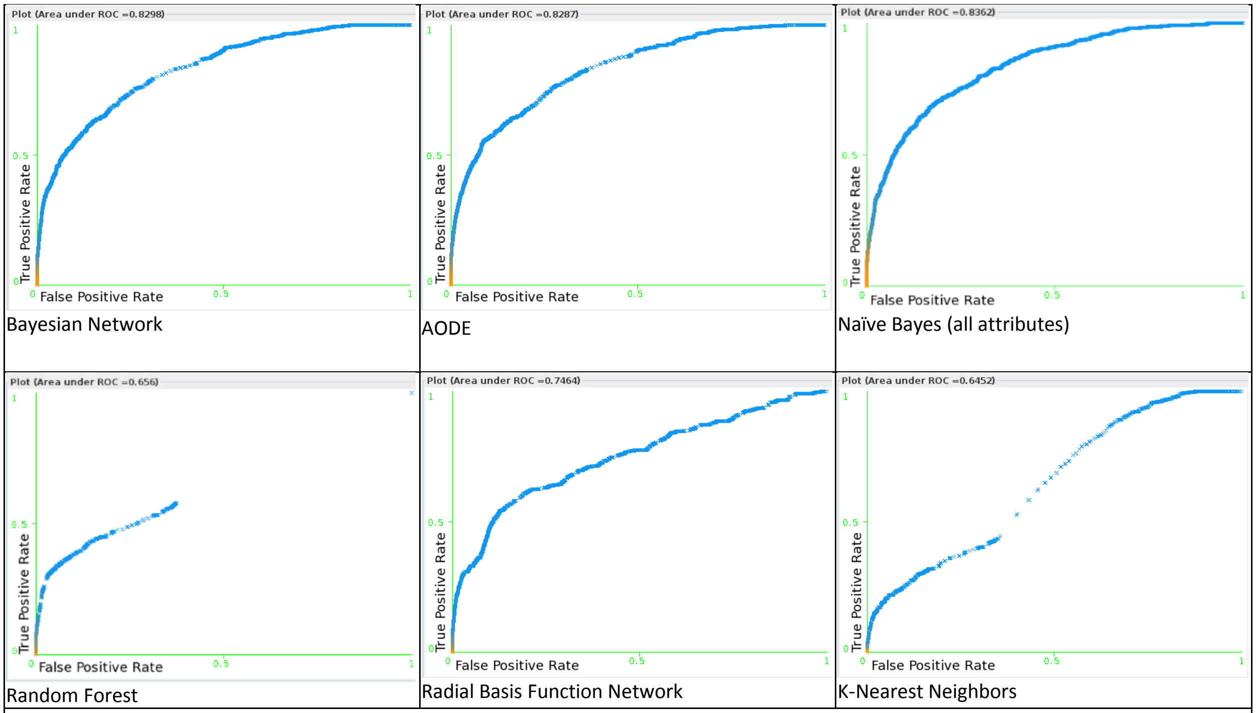
- **Dataset: 2010 CDC Hospital Discharge Survey Diagnoses; 151,551 discharges (unweighted sample), up to 15 diagnoses each**
- **Evaluation: 10-fold cross-validation; hidden attribute to predict: CCS attribute #5 (HIV)**
- **Classifiers: See Appendix B for classifier configuration details; except where “all attributes” is noted, each uses attribute selection of the top 60 custom-curated CCS attributes based on information gain for HIV, against each training fold; classifiers are listed in the table in order of execution time**

Classifier	Execution Time*	Area under ROC Curve (AUC)	False positive rate (%)	Precision (%)	Accuracy (%)	False positive rate (%)	Precision (%)	Accuracy (%)
			@ 30% recall			@ 15% recall		
Naïve Bayes	11s	0.826	2.51	5.73	97.15	0.54	12.47	99.04
AODE**	16s	0.829	1.75	7.27	97.71	0.34	18.20	99.23
Naïve Bayes (all attributes)	19s	0.836	2.22	6.42	97.44	0.58	11.57	98.99
AdaBoost	19s	0.690	4.40	3.31	95.27	1.09	6.49	98.49
Bayes Network	21s	0.830	1.58	8.82	98.08	0.44	14.71	99.13
Radial Basis Function Network	33s	0.746	3.10	4.69	96.56	0.58	11.62	98.99
Decision Tree	49s	0.520	20.01	0.60	79.70	40.01	0.54	59.90
AODE** (all attributes)	183s	0.837	1.61	8.66	98.05	0.33	18.61	99.24
Random Forest	284s	0.656	3.24	4.50	96.42	0.76	9.01	98.81
Support Vector Machine	2288s	0.523		N/A			N/A	
K-Nearest Neighbors***	4567s	0.645	14.28	1.06	85.44	2.16	3.41	97.42

* Combined time to build model (including attribute selection where used) and to test model once on full training set; see Appendix B for computing platform details

** AODE is sometimes referred to as “Not so Naïve Bayes”; see Webb, G. I., Boughton, J., Wang, Z. (2005). "Not So Naive Bayes: Aggregating One-Dependence Estimators". Machine Learning, 58(1), 5–24.

*** K-Nearest Neighbors is a “lazy learner,” where computation is deferred until classification time; so while training is fast, classification is significantly slower than other classifiers.



Bayesian Network

AODE

Naïve Bayes (all attributes)

Random Forest

Radial Basis Function Network

K-Nearest Neighbors

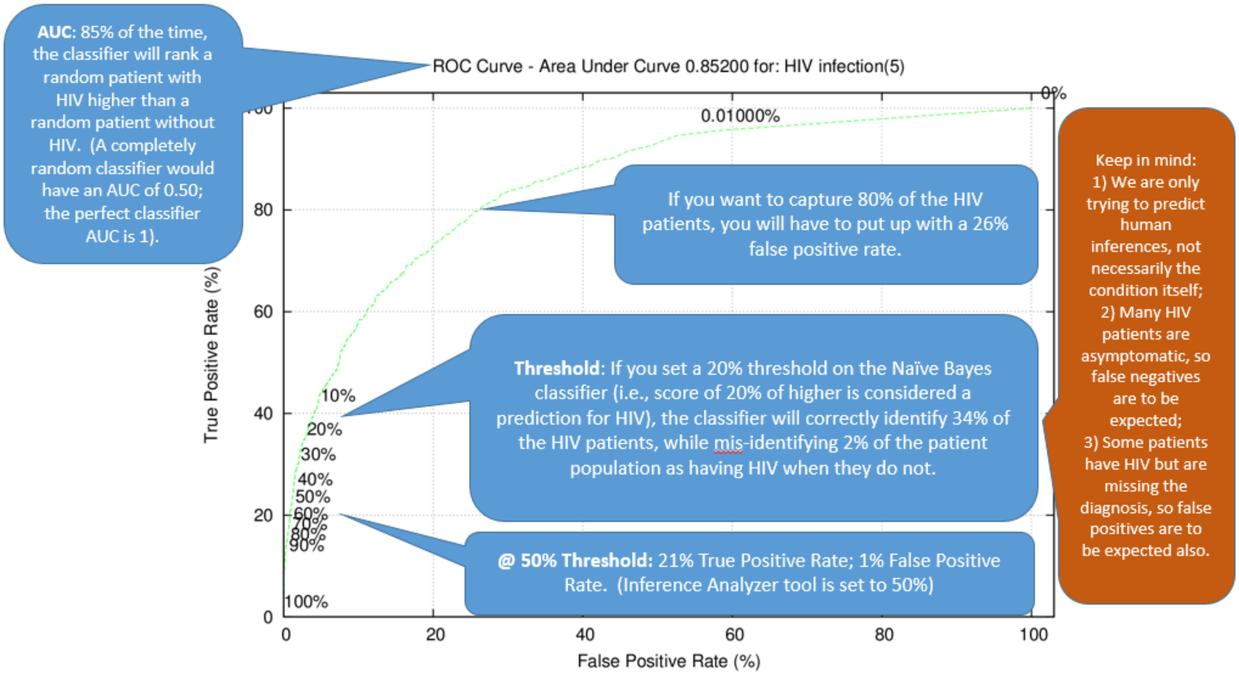
Figure 12: Receiver Operating Characteristic (ROC) curves for six classifiers as tested in Results Table 2.

Classifier threshold indicated by color: 0% 50% 100%

As mentioned in Section 4.3, the high-level observations made from predicate experiments performed on CDC hospital discharge data are similar to those made from the NMH problem list data. Indeed, the Bayesian classifiers with a dependence assumption performed best (though the Bayesian AODE classifier outperformed Bayesian Network at 15% recall in CDC data but not NMH data); followed by Naïve Bayes, and then the others.

Classifiers ranked by accuracy (at 15% and 30% recall) for prediction of a hidden HIV diagnosis in hospital discharges	
@ 30% recall	***@ 15% recall***
1. Bayesian Network 2. Bayesian Averaged One-Dependence Estimators 3. Naïve Bayes 4. Radial Basis Function Network 5. Random Forest 6. AdaBoost 7. K-Nearest Neighbors 8. Decision Tree	1. Bayesian Averaged One-Dependence Estimators 2. Bayesian Network 3. Naïve Bayes 4. Radial Basis Function Network 5. Random Forest 6. AdaBoost 7. K-Nearest Neighbors 8. Decision Tree

The two classifiers that were tested without attribute selection (Naïve Bayes and AODE) performed better without it against CDC data; but not against NMH data. One possible explanation for this is that hospital discharge diagnoses might have fewer irrelevant features that would be filtered out by attribute selection.



AUC: 85% of the time, the classifier will rank a random patient with HIV higher than a random patient without HIV. (A completely random classifier would have an AUC of 0.50; the perfect classifier AUC is 1).

If you want to capture 80% of the HIV patients, you will have to put up with a 26% false positive rate.

Threshold: If you set a 20% threshold on the Naïve Bayes classifier (i.e., score of 20% of higher is considered a prediction for HIV), the classifier will correctly identify 34% of the HIV patients, while mis-identifying 2% of the patient population as having HIV when they do not.

@ 50% Threshold: 21% True Positive Rate; 1% False Positive Rate. (Inference Analyzer tool is set to 50%)

Keep in mind:
1) We are only trying to predict human inferences, not necessarily the condition itself;
2) Many HIV patients are asymptomatic, so false negatives are to be expected;
3) Some patients have HIV but are missing the diagnosis, so false positives are to be expected also.

**Naïve Bayes classifier trained on CDC hospital discharges 1996-2010; tested on 2010 data set.*

Figure 13: Annotated ROC Curve for Naive Bayes classifier trained on 14 years of CDC data and tested against 2010 data.

9 Appendix B: Configuration Details

Classifier and system configuration details for both the NMH and CDC tests are listed below.

- Attribute Selection Configuration
  - **Information Gain** (`weka.attributeSelection.InfoGainAttributeEval`): 60 attributes; used in conjunction with classifiers and 10-fold cross validation via `weka.classifiers.meta.AttributeSelectedClassifier`.
- Classifier Configuration
  - **Naïve Bayes** (`weka.classifiers.bayes.NaiveBayes`): Default configuration
  - **Bayesian Network** (`weka.classifiers.bayes.BayesNet`): Limit of 5 parents per node in K2 search algorithm; use ADTree structure to increase speed; probability table estimation prior probability (SimpleEstimator alpha parameter) 0.4
  - **AODE** (`weka.classifiers.bayes.AODE`): Default configuration
  - **Random Forest** (`weka.classifiers.trees.RandomForest`): 30 trees, 6 random features per tree
  - **Radial Basis Function Network** (`weka.classifiers.functions.RBFClassifier`): Default configuration
  - **K-Nearest Neighbors** (`weka.classifiers.IBk`): Default configuration
  - **Decision Tree** (`weka.classifiers.trees.J48`): Default configuration
  - **Support Vector Machine** (`weka.classifiers.functions.SMO`): Default configuration
  - **AdaBoost** (`weka.classifiers.meta.AdaBoostM1`): Default configuration; it is used in conjunction with a 1-level Decision Tree (`weka.classifiers.trees.DecisionStump`)
- Computing Platform
  - Intel® Xeon® Processor E5-2680 v2 (Ivy Bridge), 2.80 GHz, Xen virtualized with 16GB RAM, 384GB SSD, 8 cores
  - Fedora Linux version 20, 64-bit
  - WEKA 3.6.10 on OpenJDK 8