6.S897/HST.956 Machine Learning for Healthcare

Lecture 5: Risk Stratification (cont.)

Instructors: David Sontag, Peter Szolovits

1 Outline

This lecture covers the following material:

- Deriving labels for data
- Evaluation of model performance
- Subtleties with machine learning-based risk stratification
- Survival Modeling

2 Deriving Labels

Labels are used to indicate whether or not a patient has a condition of interest. Labels are essential for any supervised learning technique. Labelling health-care data is not straight forward and has lot of subtleties that one has to be wary of. This section illustrates the general way process for obtaining labels. Broadly speaking, there are two approaches to generate labels

- Manually labelling by "Chart review"
- Writing an algorithm that could automatically label patients

We discuss how these two methods can be applied in labelling patients for the prediction of Type 2 Diabetes.

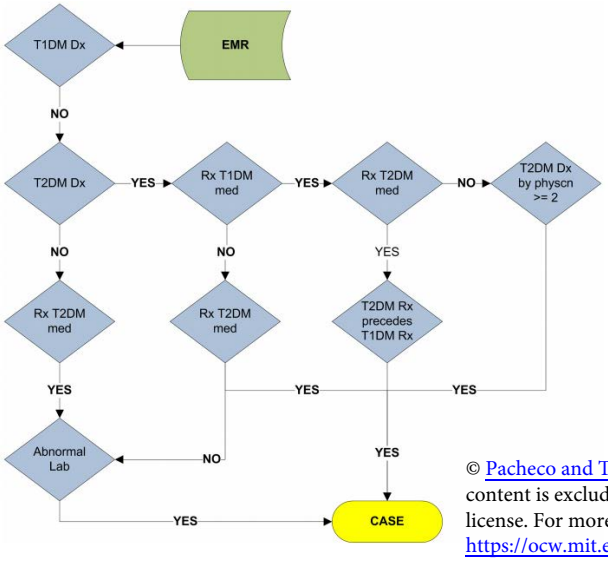
2.1 Manual Labelling

The most straight forward way is to carefully go through the patient's data and label the patient for the condition of interest. Visualizing the data of individual patient is an important part of chart review. This can be done in various ways, such as using straightforward histograms and plots, depending on the nature of the data. There are tools like *patient-viz* developed specifically to view and explore electronic health-care records.

In case of Type 2 Diabetes, one could look at the medication history of the patient and see when the patient started taking type 2 Diabetes medications like Metformin. However, this method can pose multiple problems. Certain medications can be used to treat multiple diseases. For example, Metformin can be used to treat Polycystic ovarian syndrome. There may be delay between official diagnosis and when the patient begins taking medication. Additionally, some patients might pay for the medicine themselves and hence this data could be absent from the insurance claims record. This method of labelling is also time consuming considering the amount of data used in modern machine learning approaches. This leads us to the next way of labelling i.e to write an algorithm that could automatically look through patients records and generate label.

2.2 Automatic labelling

Rule-based systems are a manual and deterministic approach for automatically labelling data. Also known as Phenotypes, these algorithms list a set of rules based on expert knowledge to label a patient for the condition of interest. Shown below is a Phenotype for Type 2 Diabetes labelling



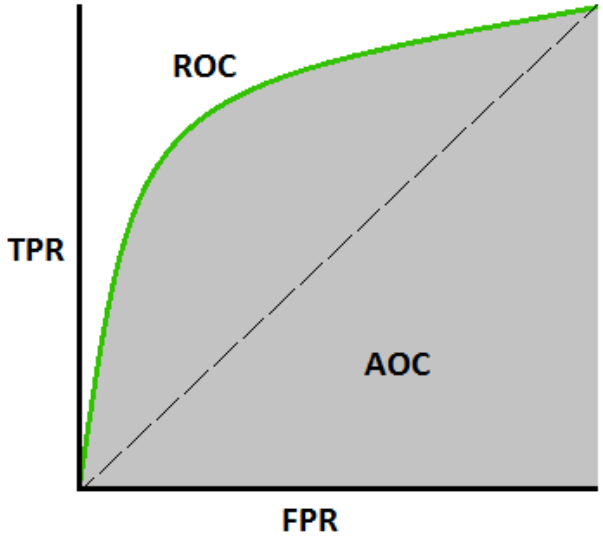
Examples of rule-based algorithms for various diseases can be found on PheKB. Rule-based systems solve the problem of handling big data but still suffer from the same set of problems described in the previous section. Additionally,these methods can become cumbersome if too many rules are used and tend to have high predictive value but poor recall. Often multiple phenotypes are available for a particular condition and they can lead to conflicting labels.

Recently, machine learning based labelling techniques are becoming popular. One could imagine setting up a supervised learning problem that can automatically learn rules for labelling a patient using only few manually labelled patients. Such techniques could use high dimensional data about the patients like lab results, medications, personal history to label the patient. Note that this learning problem is slightly different from the risk stratification problem where it uses the data even after the onset of diabetes to label the patient.

3 Evaluation

3.1 ROC Curve

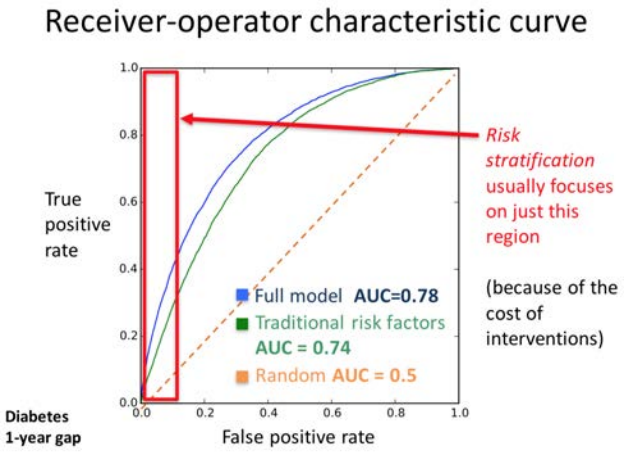
The Receiver Operating Characteristic (ROC) curve is a standard tool used for evaluating the performance of binary classification tasks. The curve maps the true positive vs false positive rates at different classification thresholds, as shown below.



3.2 Area Under (ROC) Curve (AUC)

The area underneath ROC curve or AUC is a performance metric that is derived from the ROC curve. In situations where discriminative power is our main concern, AUC is preferred over accuracy when comparing two models because AUC implicitly ensures that model doesn't overfit to a single class when the distribution of samples are skewed. To understand this, let's consider a skewed data set with 10% positive samples. A binary classifier that always predicts negative will have an accuracy of 0.9 however AUC for this model will be bad. This is particularly useful in healthcare data, where heavy class imbalances are common.

Examining this curve for general trends, we see that if all of our predictions are correct, then AUC would equal 1. If all of our predictions are random (ie. the model assigns a random probability/score of being in the positive class), then we expect the AUC to equal 0.5. AUC is also equal to the probability that a model assigns a score higher to a positive sample compared to a negative sample. More details about this probabilistic interpretation can be found here.



However, AUC has some limitations. Risk stratification involves identifying patients with high risk and making interventions and is focused only on the left part of the ROC curve. In this case, AUC might not

give a complete representation of the problem. One could get around this issue by using partial AUC that would integrate across a small portion of the ROC curve.

3.3 Calibration

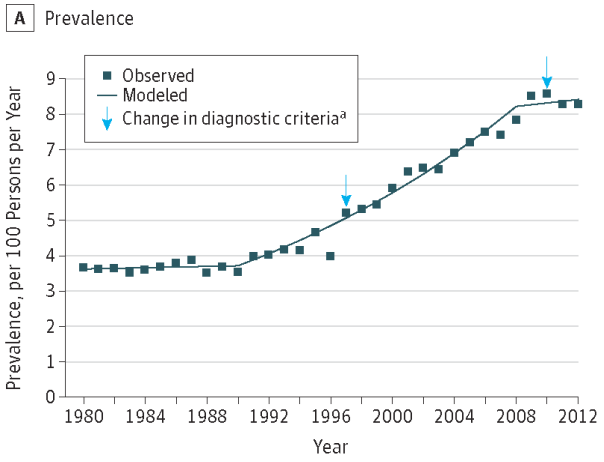
When running our model, we want some way to straightforwardly assess our performance and determine how to adjust our model. This can be done straightforwardly through visualization. One example discussed in class involved plotting the actual probability against the predicted probability. If predictions were made perfectly, this line would have a slope of 1. One recommendation is to plot the confidence interval alongside the probabilities. This is because we might not have enough data to make conclusive predictions, so giving a range allows for more flexibility.

4 Subtleties with ML-based risk stratification

In this section, we discuss some of the peculiarities of healthcare data.

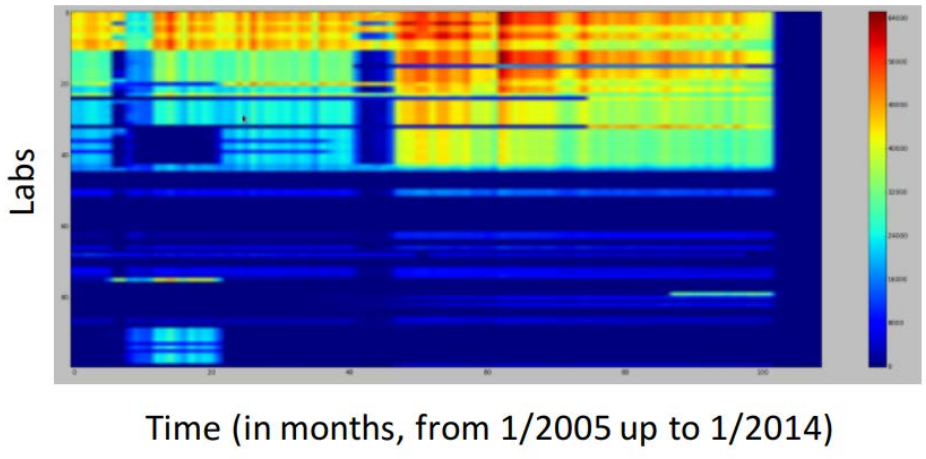
4.1 Non-stationarity of health-care data

Non-stationarity refers to the temporal changes in the distribution of data. It is common in health-care data because of the changes in lifestyle of people, invention of new technologies and changes in government policies. We can illustrate this by considering the prevalence of diabetes in United States over the course of three decades. Below, we see that there are upticks in the number of diagnosis.



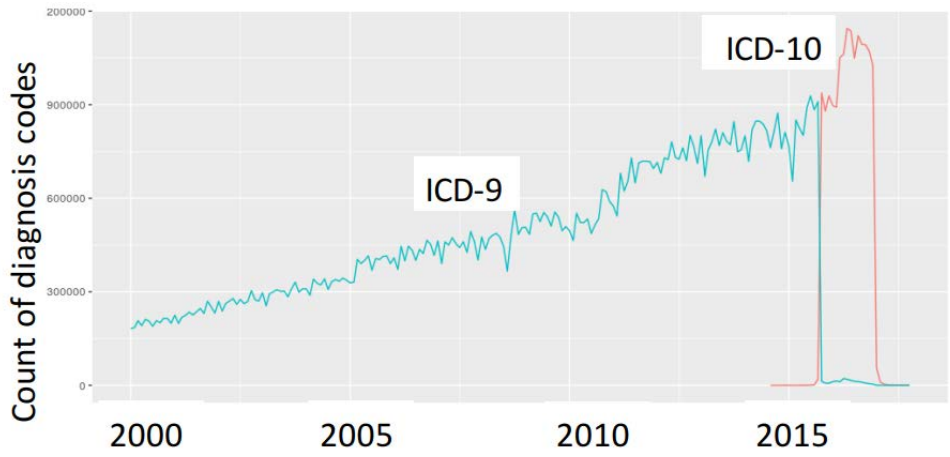
Does this mean that some sudden change happened in the population that caused people to have more diabetes? In reality, this was caused by a change in diagnostic criteria. Because diagnostic criteria are human-defined, this may occur.

We can consider another example. In the following heat-map, the x-axis represents time and the y-axis represents a particular test. The color represents the popularity of a particular test (blue = low, red = high). In certain areas, we notice a blue streak followed by a sudden change to bright orange. This indicates that a new test was created.

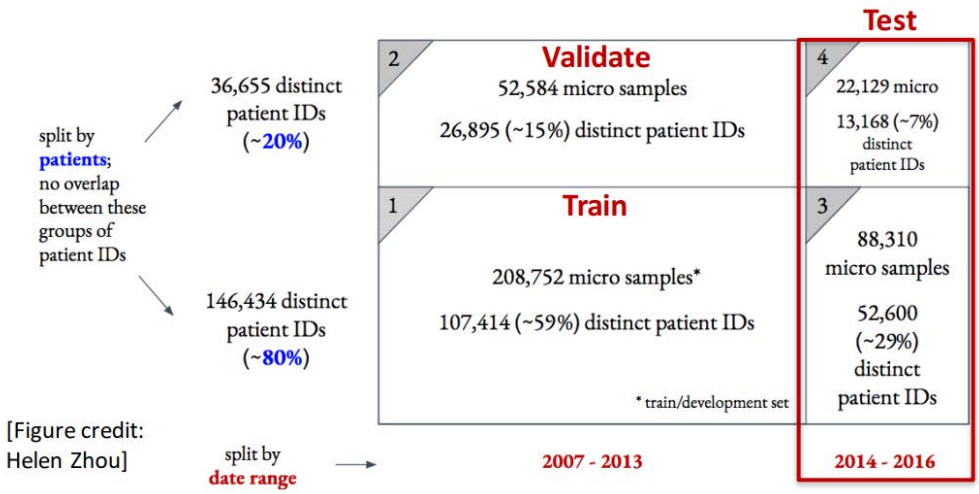


Similarly, we also notice that some tests transition from bright orange to dark blue, indicating that the test was not administered or available for some time period or the insurer stopped reimbursing that test. Few other tests show a gap with no instances in between, this could be because the EHR went offline and data was not recorded as a result.

Features can also lose relevance over time because of change in conventions. The following plot shows the frequency of ICD-9 and ICD-10 codes in medical records over the last few years.



These examples highlight the fact that data changes overtime, so it is not safe to assume that data available in the past will be available again at some point in the future. This is to also say that if a deep learning model predicts well in the present, the model may perform poorly if data changes in the future. One way to ensure that models don't lose relevance in the future due to non-stationarity is to report the test metrics of the model on data is in future compared to the training data. The following image demonstrates the above trick for prediction of diabetes onset which is a non-stationary data.

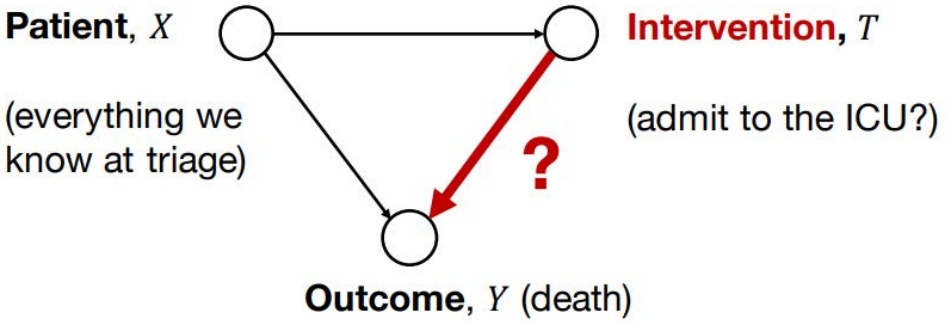


4.2 Intervention-tainted outcomes

Patients go through interventions in hospitals and ignoring this while developing the model can lead to false rules. This was demonstrated in Caruana et al. [CLG+15] where models learned that having Asthma lowered the risk of patient dying from Pneumonia. This is because patients with Asthma are given aggressive treatments for Pneumonia and hence the mortality rate is low but this doesn't imply that they are at low risk. There are a few ways by which one could get around this problem.

- If the model is interpretable, an expert could detect this anomaly and correct it.
- Redefine the classification problem by finding a pre-treatment surrogate.
- Consider patients with interventions to be right-censored by the treatment

One could think of scenarios where each of this hack could break down. More rigorous way to address this problem would be through the language of causality. In this model, the survival rate of a patient will also depend on the type of treatment strategy used. Causal model for the above discussed Pneumonia problem can be represented as shown below



4.3 Complex vs Simple Models

Simpler models might be able to generalize better if they are less sensitive to changes in the dataset. This was seen in a google paper, where the difference in performance results from a complex deep-learning model and from a simple regression model were not statistically significant. In other words, while the deep-learning model performed negligibly better at the time of publication, it is far less likely to perform as well as the simpler model in the future.

5 Survival Modeling

Survival modeling predicts the time until some future event. In class, we discussed two primary reasons for how this is different from general classification.

The first is that with general classification, if an individual does not have a definitive 0/1 label, then that individual cannot be used in the model. This is an issue with medical data, as many patients do not receive any sort of label whatsoever. For example, if someone lacks data on a diabetes diagnosis, we still cannot be certain that the individual did not develop diabetes.

The second reason is that classification tasks do not give you granularity. For example, if someone died at 1.1 years from present and your model predicted 1.0 years, then in the standard classification framework, your prediction is altogether wrong despite being close to the true value. Survival modeling allows us to address this issue.

References

[BSOM+14] David W Bates, Suchi Saria, Lucila Ohno-Machado, Anand Shah, and Gabriel Escobar. Big data in health care: using analytics to identify and manage high-risk and high-cost patients. *Health Affairs*, 33(7):1123–1131, 2014.

[CLG+15] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1721–1730, New York, NY, USA, 2015. ACM.

[WSW14] Xiang Wang, David Sontag, and Fei Wang. Unsupervised learning of disease progression models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 85–94. ACM, 2014.

6.S897 / HST.956 Machine Learning for Healthcare

Spring 2019