Machine Learning for Clinical Predictive Analytics

Wei-Hung Weng [1]

Learning Objectives

- Understand the basics of machine learning techniques and the reasons behind why they are useful for solving clinical prediction problems.
- Understand the intuition behind some machine learning models, including regression, decision trees, and support vector machines.
- Understand how to apply these models to clinical prediction problems using publicly available datasets via case studies.

1. Machine Learning for Healthcare

1.1. Introduction

In this chapter, we provide a brief overview of applying machine learning techniques for clinical prediction tasks. We begin with a quick introduction to the concepts of machine learning, and outline some of the most common machine learning algorithms. Next, we demonstrate how to apply the algorithms with appropriate toolkits to conduct machine learning experiments for clinical prediction tasks.

This chapter is composed of five sections. First, we will explain why machine learning techniques are helpful for researchers in solving clinical prediction problems (section 1). Understanding the motivations behind machine learning approaches in healthcare are essential, since precision and accuracy are often critical in healthcare problems, and everything from diagnostic decisions to predictive clinical analytics could dramatically benefit from data-based processes with improved efficiency and reliability. In the second section, we will introduce several important concepts in machine learning in a colloquial manner, such as learning scenarios, objective/target function, error and loss function and metrics, optimization and model validation, and finally a summary of model selection methods (section 2). These topics will help us utilize machine learning algorithms in an appropriate way. Following that, we will introduce some popular machine learning algorithms for prediction problems (section 3). For example, logistic regression, decision tree and support vector machine. Then, we will discuss some limitations and pitfalls of using the machine learning approach (section 4). Lastly, we will provide case studies using real intensive care unit (ICU) data from a publicly available dataset, PhysioNet Challenge 2012, as well as the breast tumor data from Breast Cancer Wisconsin (Diagnostic) Database, and summarize what we have mentioned in the chapter (section 5).

1.2. Why machine learning?

Machine learning is an interdisciplinary field which consists of computer science, mathematics, and statistics. It is also an approach toward building intelligent machines for artificial intelligence (AI). Different from rule-based symbolic AI, the idea of utilizing machine learning for AI is to learn from data (examples and experiences). Instead of explicitly programming hand-crafted rules, we construct a model for prediction by feeding data into a machine learning algorithm, and the algorithm will learn an optimized function based on the data and the specific task. Such data-driven methodology is now the state-of-the-art approach of various research domains, such as computer vision (Krizhevsky et al., 2012), natural language processing (NLP) (Yala et al., 2017), and speech to text translation (Wu et al., 2016; Chung et al., 2018; 2019), for many complex real-world applications.

Due to the increased popularity of the electronic health record (EHR) system in recent years, massive quantities of healthcare data have been generated (Henry et al., 2016). Machine learning for healthcare therefore becomes an emerging applied domain. Recently, researchers and clinicians have started applying machine learning algorithms to solve the problems of clinical outcome prediction (Ghassemi et al., 2014), diagnosis (Gulshan et al., 2016; Esteva et al., 2017; Liu et al., 2017; Chung & Weng, 2017; Nagpal et al., 2018), treatment and optimal decision making (Raghu et al., 2017; Weng et al., 2017a; Komorowski et al., 2018) using data in different modalities, such as structured lab measurements (Pivovarov et al., 2015), claims data (Doshi-Velez et al., 2014; Pivovarov et al., 2015; Choi et al., 2016), free texts (Pivovarov et al., 2015; Weng & Szolovits, 2018; Weng et al., 2019b), images (Gulshan et al., 2016; Esteva et al., 2017; Bejnordi et al., 2017; Chen et al., 2019a), physi-

[1] MIT CSAIL, Cambridge, MA, USA. Correspondence to: Wei-Hung Weng <ckbjimmy@mit.edu>.

ological signals (Lehman et al., 2018), and even cross-modal information (Hsu et al., 2018; Liu et al., 2019).

Instead of traditional ad-hoc healthcare data analytics, which usually requires expert-intensive efforts for collecting data and designing limited hand-crafted features, machine learning-based approaches help us recognize patterns inside the data and allow us to perform personalized clinical prediction with more generalizable prediction models (Gehrmann et al., 2018). They help us maximize the utilization of massive but complex EHR data. In this chapter, we will focus on how to tackle clinical prediction problems using a machine learning-based approach.

2. General Concepts of Learning

2.1. Learning scenario for clinical prediction

We start with how to frame your clinical problem into a machine learning prediction problem with a simple example. Assuming that you want to build a model for predicting the mortality of ICU patients with continuous renal replacement therapy and you have a large ICU database, which includes hundreds of variables such as vital signs, lab data, demographics, medications, and even clinical notes and reports, the clinical problem can be reframed as a task: “Given data with hundreds of input variables, I want to learn a model from the data that can correctly make a prediction given a new datapoint.” That is, the output of the function (model) should be as close as possible to the outcome of what exactly happened (the ground truth). Machine learning algorithm is here to help you to find the best function from a set of functions. This is a typical machine learning scenario, which is termed supervised learning. In such a case, you may do the following steps:

- Define the outcome of your task
- Consult with domain experts to identify important features/variables
- Select an appropriate algorithm (or design a new machine learning algorithm) with a suitable parameter selection
- Find an optimized model with a subset of data (training data) with the algorithm
- Evaluate the model with another subset of data (testing data) with appropriate metrics
- Deploy the prediction model on real-world data

At the end of the chapter, we will show an exercise notebook that will help you go through the concepts mentioned above.

2.2. Machine learning scenarios

There are many machine learning scenarios, such as supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, and transfer learning. We will only focus on the first two main categories, supervised learning and unsupervised learning. Both of the scenarios expect to learn from the underlying data distribution, or to put it simply, find patterns inside data. The difference between them is that you have annotated data under the supervised scenario but only unlabelled data under unsupervised learning scenario.

2.2.1. Supervised learning

Supervised learning is the most common scenario for practical machine learning tasks if the outcome is well-defined, or example, if you are predicting patient mortality, hospital length of stay, or drug response. In general, the supervised learning algorithm will try to learn how to build a classifier for predicting the outcome variable y given input x, which is a mapping function f where $y = f(x)$. The classifier will be built by an algorithm along with a set of data $\{x_1, ..., x_n\}$ with the corresponding outcome label $\{y_1, ..., y_n\}$.Supervised learning can be categorized by two criteria, either by type of prediction or by type of model. First, it can be separated into regression or classification problems. For predicting continuous outcomes, using regression methods such as linear regression is suitable. For class prediction, classification algorithms such as logistic regression, naive Bayes, decision trees or support vector machines (SVM) (Cortes & Vapnik, 1995) will be a better choice. For example, linear regression is suitable for children height prediction problem whereas SVM is better for binary mortality prediction.

Regarding the goal of the learning process, a discriminative model such as regression, trees and SVMs can learn the decision boundary within the data.However, a generative model like naive Bayes will learn the probability distributions of the data.

2.2.2. Unsupervised learning

Without corresponding output variables (y), the unsupervised learning algorithms discover latent structures and patterns directly from the given unlabeled data $\{x_1, ..., x_n\}$.

There is no ground truth in the unsupervised learning, therefore, the machine will only find associations or clusters inside the data.For example, we may discover hidden subtypes in a disease using an unsupervised approach (Ghassemi et al., 2014).

2.2.3. Other scenario

Other scenarios such as reinforcement learning (RL) frame a decision making problem into a computer agent interaction with a dynamic environment (Silver et al., 2016), in which the agent attempts to reach the best reward based on feedback when it navigates the state and action space. Using a clinical scenario as an example, the agent (the RL algorithm) will try to improve the model parameters based on iteratively simulating the state (patient condition) and action (giving fluid or vasopressor for hypotension), obtain the feedback reward (mortality or not), and eventually converge to a model that may yield optimal decisions (Raghu et al., 2017).

2.3. Find the best function

To estimate and find the best mapping function in the above scenarios, the process of optimization is needed. However, we do need to define some criteria to tell us how well the function (model) can predict the task. Therefore, we need a loss function and a cost function (objective function) for this purpose.

Loss function defines the difference between the output of model y and the real data value $\hat{y}$. Different machine learning algorithms may use different loss functions, for example, least squared error for linear regression, logistic loss for logistic regression, and hinge loss for SVM (Table 1). Cost function is the summation of loss functions of each training data point. Using loss functions, we can define the cost function to evaluate model performance. Through loss and cost functions, we can compute the performance of functions on the whole dataset.

In unsupervised learning setting, the algorithms have no real data value to compute the loss function. In such case, we can use the input itself as the output and compute the difference between input and output. For example, we use reconstruction loss for autoencoder, a kind of unsupervised learning algorithms, to evaluate whether the model can well reconstruct the input from hidden states inside the model.

There is a mathematical proof for this learning problem to explain why machine learning is feasible even if the function space is infinite. Since our goal is not to explain the mathematics and mechanism of machine learning, further details on why there is a finite bound on the generalization error are not mentioned here. For readers who are interested in the theory of machine learning, such as Hoeffding's inequality that gives a probability upper bound, VapnikChervonenkis (VC) dimension and VC generalization bound, please refer to the textbooks (Abu-Mostafa et al., 2012).

2.4. Metrics

Choosing an appropriate numeric evaluation metric for optimization is crucial. Different evaluation metrics are applied to different scenarios and problems.

2.4.1. Supervised learning

In classification problems, accuracy, precision/positive predictive value (PPV), recall/sensitivity, specificity, and the F1 score are usually used. We use a confusion matrix to show the relation between these metrics 2.

The area under receiver operating curve (AUROC) is a very common metric, which sums up the area under the curve in the plot with x-axis of false positive rate (FPR, also known as 1-specificity), and y-axis of true positive rate (TPR) 1. FPR and TPR values may change based on the threshold of your subjective choice.

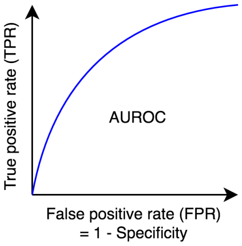

Figure 1. Example of AUROC.

In a regression problem, the adjusted R-squared value is commonly used for evaluation. The R-squared value, also known as the coefficient of determination, follows the equation and is defined by the total sum of squares (SStot) and the residual sum of squares (SSres). The detailed equations are as follows:

$$\mathrm{R}^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^{m}(y_i - f(x_i))^2}{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}$$

$$\text{Adjusted } \mathrm{R}^2 = 1 - \frac{(1 - \mathrm{R}^2)(m-1)}{m-n-1}$$

There are also other metrics for regression, such as Akaike information criterion (AIC) and Bayesian information criterion (BIC), for different study purposes.

2.4.2. Unsupervised learning

Since there are no ground truth labels for unsupervised scenarios, evaluation metrics of unsupervised learning settings are relatively difficult to define and usually depend on the algorithms in question. For example, the Calinski-Harabaz index and silhouette coefficient have been used to evaluate k-means clustering. Reconstruction error is used for autoen-

Task	Error type	Loss function	Note
Regression	Mean-squared error	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2$	Easy to learn but sensitive to outliers (MSE, L2 loss)
	Mean absolute error	$\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y_i}\rvert$	Robust to outliers but not differentiable (MAE, L1 loss)
Classification	Cross entropy = Log loss	$-\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y_i}) + (1-y_i)\log(1-\hat{y_i})]$ $= -\frac{1}{n}\sum_{i=1}^{n} p_i \log q_i$	Quantify the difference between two probability distributions
	Hinge loss	$\frac{1}{n}\sum_{i=1}^{n} max(0, 1 - y_i\hat{y_i})$	For support vector machine
	KL divergence	$D_{KL}(p\|\|q) = \sum_i p_i(\log\frac{p_i}{q_i})$	Quantify the difference between two probability distributions

Table 1. Examples of commonly-used loss functions in machine learning.

		Predicted		
		True	False	
Actual	True	True positive (TP)	False negative (FN) Type II error	Recall = Sensntivity = $\frac{\mathrm{TP}}{\mathrm{TP+FN}}$
	False	False positive (FP) Type I error	True negative (TN)	Specificity = $\frac{\mathrm{TN}}{\mathrm{TN+FP}}$

		Precision = $\frac{\mathrm{TP}}{\mathrm{TP+FP}}$		Accuracy = $\frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$
				F1 = $\frac{2\times\mathrm{Precision}\times\mathrm{Recall}}{\mathrm{Precision+Recall}}$

Table 2. Commonly-used metrics in machine learning.

coder, a kind of neural network architecture for learning data representation.

2.5. Model Validation

The next step after deciding the algorithm is to get your data ready for training a model for your task. In practice, we split the whole dataset into three pieces:

- Training set for model training. You will run the selected machine learning algorithm only on this subset.
- Development (a.k.a. dev, validation) set, also called hold-out, for parameter tuning and feature selection. This subset is only for optimization and model validation.
- Testing set for evaluating model performance. We only apply the model for prediction here, but wont change any content in the model at this moment.

There are a few things that we need to keep in mind:

- It is better to have your training, dev and testing sets all from the same data distribution instead of having them too different (e.g. training/dev on male patients but testing on female patients), otherwise you may face the problem of overfitting, in which your model will fit the data too well in training or dev sets but find it difficult to generalize to the test data. In this situation, the trained model will not be able to be applied to other cases.
- It is important to prevent using any data in the dev set or testing set for model training. Test data leakage, i.e. having part of testing data while training phase, may cause the overfitting of the model to your test data and erroneously gives you a high performance but a bad model.

There is no consensus on the relative proportions of the three subsets. However, people usually split out 20-30% of the whole dataset for their testing set. The proportion can be smaller if you have more data.

2.5.1. Cross-validation

The other commonly used approach for model validation is k-fold cross validation (CV). The goal of k-fold CV is to reduce the overfitting of the initial training set by further training several models with the same algorithm but with different training/dev set splitting.

In k-fold CV, we split the whole dataset into k folds and train the model k times. In each training, we iteratively leave one different fold out for validation, and train on the remaining $k-1$ folds. The final error is the average of errors over k times of training 2. In practice, we usually use k=5 or 10. The extreme case for n cases is n-fold CV, which is also called leave-one-out CV (LOOCV).

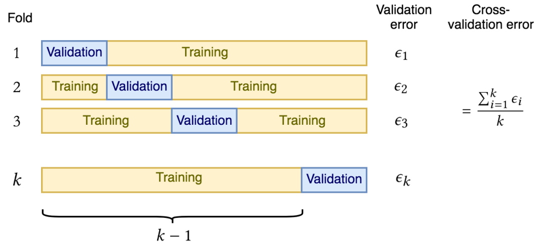

Figure 2. K-fold cross-validation.

Please keep in mind that the testing set is completely excluded from the process of CV. Only training and dev sets are involved in this process.

2.6. Diagnostics

After the first iteration of model training and evaluation, you may find that the trained model does not perform well on the unseen testing data. To address the issue of error in machine learning, we need to conduct some diagnostics regarding bias and variance in the model in order to achieve a model with low bias and low variance.

2.6.1. Bias and variance

The bias of a model is the difference between the expected prediction and the correct model that we try to predict for given data points. That is, it is the algorithm's error rate on training set. This is an underfitting problem, which the model can't capture the trend of the data well due to excessively simple model, and one potential solution is to make the model more complex, which can be done by reducing regularization (section 2.6.2), or configuring and adding more input features. For example, stacking more layers if you are using a deep learning approach. However, it is possible that the outcome of complex model is high variance.

The variance of a model is the variability of the model prediction for given data points. It is the model error rate difference between training and dev sets. Problems of high variance are usually related to the issue of overfitting. i.e. hard to generalize to unseen data. The possible solution is to simplify the model, such as using regularization, reducing the number of features, or add more training data. Yet the simpler model may also suffer from the issue of high bias.

High bias and high variance can happen simultaneously with very bad models. To achieve the optimal error rate, a.k.a. Bayes error rate, which is an unavoidable bias from the most optimized model, we need to do iterative experiments to find the optimal bias and variance tradeoff.

Finally, a good practice of investigating bias and variance is to plot the informative learning curve with training and validation errors. In Figure 3 and Table 3 we demonstrate a few cases of diagnostics as examples.

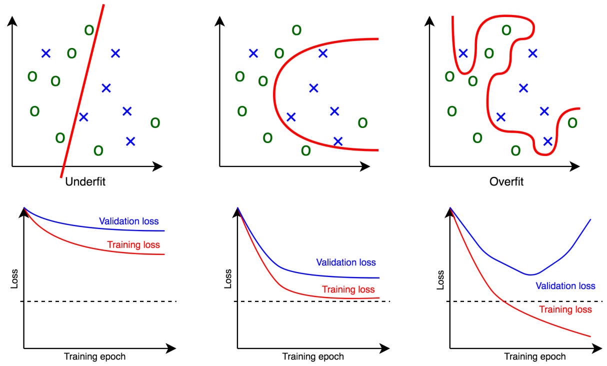

Figure 3. Bias and variance.

2.6.2. Regularization

The goal of regularization is to prevent model overfitting and high variance. The most common regularization techniques include Least absolute shrinkage and selection operator (LASSO regression, L1-regularization) (Tibshirani, 1996), ridge regression (L2-regression) (Hoerl & Kennard, 1970), and elastic net regression (a linear combination of L1 and L2 regularization) (Zou & Hastie, 2005).

In practice, we add a weighted penalty term λ to the cost function as a regularization. For L1-regularization, we add the absolute value of the magnitude of coefficient as penalty term, and in L2-regularization we add the squared value of magnitude instead (Table 4).

L1-regularization is also a good technique for feature selection since it can ”shrink” the coefficients of less important features to zero and remove them. In contrast, L2-regularization just makes the coefficients smaller, but not to zero.

2.7. Error analysis

It is an important practice to construct your first prediction pipeline as soon as possible and iteratively improve its performance by error analysis. Error analysis is a critical step to examine the performance between your model and the optimized one. To do the analysis, it is necessary to manually go through some erroneously predicted data from the dev set.

The error analysis can help you understand potential problems in the current algorithm setting. For example, the misclassified cases usually come from specific classes (e.g. patients with cardiovascular issues might get confused with those with renal problems since there are some shared patho-

	Training error	Validation error	Approach
High bias	High	Low	Increase complexity
High variance	Low	High	Decrease complexity Add more data

Table 3. The characteristic of high bias and high variance.

Regularization	Equation
L1 (LASSO)	$\sum_{i=1}^{m}(y_i - \sum_{j=1}^{n} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{n} \lvert\beta_j\rvert$
L2 (Ridge)	$\sum_{i=1}^{m}(y_i - \sum_{j=1}^{n} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{n} \beta_j^2$

Table 4. L1 and L2-regularized logistic regression.

logical features between two organ systems) or inputs with specific conditions (Weng et al., 2017b). Such misclassification can be prevented by changing to more complex model architecture (e.g. neural networks), or adding more features (e.g. combining word- and concept-level features), in order to help distinguish the classes.

2.8. Ablation analysis

Ablation analysis is a critical step for identifying important factors in the model. Once you obtain an ideal model, it is necessary to compare it with some simple but robust models, such as linear or logistic regression model. This step is also essential for research projects, since the readers of your work will want to know what factors and methods are related to the improvement of model performance. For example, the deep learning approach of clinical document deidentification outperforms traditional natural language processing approach. In the paper of using neural network for deidentification (Dernoncourt et al., 2017), the authors demonstrate that the character-level token embedding technique had the greatest effect on model performance, and this became the critical factor of their study.

3. Learning Algorithms

In this section, we briefly introduce the concepts of some algorithm families that can be used in the clinical prediction tasks. For supervised learning, we will discuss linear models, tree-based models and SVM. For unsupervised learning, we will discuss the concepts of clustering and dimensionality reduction algorithms. We will skip the neural network method in this chapter. Please refer to programming tutorial part 3 or deep learning textbook for further information (Goodfellow et al., 2016).

3.1. Supervised learning

3.1.1. Linear models

Linear models are commonly used not only in machine learning but also in statistical analysis. They are widely adopted in the clinical world and can usually be provided as baseline models for clinical machine learning tasks. In this class of algorithms, we usually use linear regression for regression problems and logistic regression for classification problems.

The pros of linear models include their interpretability, less computation, as well as less complexity comparing to other classical machine learning algorithms. The cons of them are their inferior performance. However, these are common trade-off features in model selection. It is still worthwhile to start from this simple but powerful family of algorithms.

3.1.2. Tree-based models

Tree-based models can be used for both regression and classification problems. Decision tree, also known as classification and regression trees (CART), is one of the most common tree-based models (Breiman, 2017). It follows the steps below to find the best tree:

- It looks across all possible thresholds across all possible features and picks the single feature split that best separates the data
- The data is split on that feature at a specific threshold that yields the highest performance
- It iteratively repeats the above two steps until reaching the maximal tree depth, or until all the leaves are pure

There are many parameters that should be considered while using the decision tree algorithm. The following are some important parameters:

- Splitting criteria: by Gini index or entropy

- Tree size: tree depth, tree pruning
- Number of samples: minimal samples in a leaf, or minimal sample to split a node

The biggest advantage of a decision tree is providing model interpretability and actionable decision. Since the tree is represented in a binary way, the trained tree model can be easily converted into a set of rules. For example, in the paper the authors utilized CART to create a series of clinical rules (Fonarow et al., 2005). However, decision trees may have the issue of high variance and yield an inferior performance.

Random forest is another tree-based algorithm that combines the idea of bagging and subsampling features (Breiman, 2001). In brief, it tries to ensemble the results and performances of a number of decision trees that were built by randomly selected sets of features. The algorithm can be explained as follows:

- Pick a random subset of features
- Create a bootstrap sample of data (randomly resample the data)
- Build a decision tree on this data
- Iteratively perform the above steps until termination

Random forest is a robust classifier that usually works well on most of the supervised learning problems, but a main concern is model interpretability. There are also other tree-based models such as adaptive boosting (Adaboost) and gradient boosting algorithms, which attempt to combine multiple weaker learners into a stronger model (Freund et al., 1999; Friedman, 2001).

3.1.3. Support vector machine (SVM)

SVM is a very powerful family of machine learning algorithms (Cortes & Vapnik, 1995). The goal of SVM is trying to find a hyperplane (e.g. a line in 2D, a plane in 3D, or a n-dimension structure in a $n+1$ dimensions space) to separate data points into two sides, and the hyperplane has to maximize the minimal distance from the sentinel data points, support vectors, to the hyperplane 4.

SVM also works for non-linear separable data. It uses a technique called "kernel trick" that linearly splits the data in another vector space, then converts the space back to the original one later 5. The commonly used kernels include linear kernel, radial basis function (RBF) kernel and polynomial kernel.

Regarding the optimization, we used hinge loss to train SVM. The pros of using SVM is its superior performance, yet the model's inferior interpretability limits its applications in the healthcare domain.

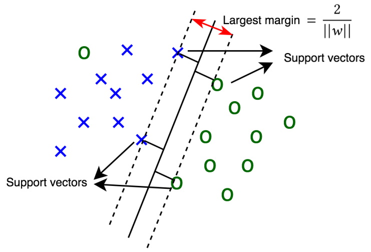

Figure 4. Hyperplane of SVM to linearly separate samples.

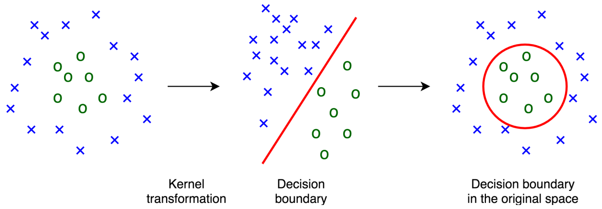

Figure 5. Kernel trick of SVM.

3.2. Unsupervised learning

In the previous section, we mentioned that the goal of unsupervised learning is to discover hidden patterns inside data. We can use clustering algorithms to aggregate data points into several clusters and investigate the characteristics of each cluster. We can also use dimensionality reduction algorithms to transform a high-dimensional into a smaller-dimensional vector space for further machine learning steps.

3.2.1. Clustering

K-means clustering, Expectation-Maximization (EM) algorithm, hierarchical clustering are all common clustering methods. In this section, we will just introduce k-means clustering. The goal of k-means clustering is to find latent groups in the data, with the number of groups represented by the variable k.

The simplified steps of k-means clustering are (Figure 6):

- Randomly initializing k points as the centroids of the k clusters
- Assigning data points to the nearest centroid and forming clusters
- Recomputing and updating centroids based on the mean value of data points in the cluster
- Repeating step 2 and 3 until convergence

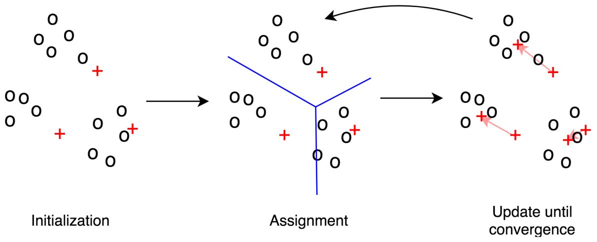

Figure 6. Steps of k-means clustering.

The k-means algorithm is guaranteed to converge to a final result. However, this converged state may be local optimum and therefore need to experiment several times to explore the variability of results.

The obtained final k centroids, as well as the cluster labels of data points can all serve as new features for further machine learning tasks, as well be shown in Section 9 of the "Applied Statistical Learning in Python" chapter. Regarding choosing the cluster number k, there are several techniques for k value validation. The most common methods include elbow method, silhouette coefficient, also Calinski-Harabaz index. However, it is very useful to decide k if you already have some clinical domain insights about potential cluster number.

3.2.2. Dimensionality reduction

While dealing with clinical data, it is possible that you need to face with a very high-dimensional but sparse dataset. Such characteristics may decrease the model performance even if you use top performing machine algorithms such as SVM, random forest or even deep learning due to the risk of overfitting. A potential solution is to utilize the power of dimensionality reduction algorithms to convert the dataset into lower dimensional vector space.Principal component analysis (PCA) is a method that finds the principal components of the data by transforming data points into a new coordinate system (Jolliffe, 2011). The first axis of the new coordinate system corresponds to the first principal component (PC1), which explains the most variance in the data and can serve as the most important feature of the dataset.

PCA is a linear algorithm and therefore it is hard to interpret the complex polynomial relationship between features. Also, PCA may not be able to represent similar data points of high-dimensional data that are close together since the linear algorithm does not consider non-linear manifolds.

The non-linear dimensionality reduction algorithm, t-Distributed Stochastic Neighbor Embedding (t-SNE), becomes an alternative when we want to explore or visualize the high-dimensional data (Maaten & Hinton, 2008). t-SNE considers probability distributions with random walk on neighborhood graphs on the curved manifold to find the patterns of data. Autoencoder is another dimensionality reduction algorithm based on a neural network architecture that aims for learning data representation by minimizing the difference between the input and output of the network (Rumelhart et al., 1988; Hinton & Salakhutdinov, 2006).

The dimensionality reduction algorithms are good at representing multi-dimensional data. Also, a smaller set of features learned from dimensionality reduction algorithms may not only reduce the complexity of the model, but also decrease model training time, as well as inference (classification/prediction) time.

4. Pitfalls and Limitations

Machine learning is a powerful technique for healthcare research. From a technical and algorithmic perspective, there are many directions that we can undertake to improve methodology, such as generalizability, less supervision, multimodal and multitask training (Weng et al., 2019a), or learning temporality and irregularity (Xiao et al., 2018).

However, there are some pitfalls and limitations about utilizing machine learning in healthcare that should be considered while model development (Chen et al., 2019b). For example, model biases and fairness is a critical issue since the training data we use are usually noisy and biased (Caruana et al., 2015; Ghassemi et al., 2018). We still need human expert to validate, interpret and adjust the models. Model interpretability is also an important topic from the aspects of (1) human-machine collaboration and (2) building a human-like intelligent machine for medicine (Girkar et al., 2018). Causality is usually not being addressed in most of the clinical machine learning research, yet it is a key of clinical decision making. We may need more complicated causal inference algorithms to answer clinical causal questions.

We also need to think more about how to deploy the developed machine learning models into clinical workflow. How to utilize them to improve workflow (Horng et al., 2017; Chen et al., 2019a), as well as integrate all information acquired by human and machine, to transform them into clinical actions and improve health outcomes are the most important things that we should consider for future clinician-machine collaboration.

5. Programming Exercise

We provide three tutorials for readers to have some hands-on exercises of learning basic machine learning concepts, algorithms and toolkits for clinical prediction tasks. They can be accessed through Google colab and Python Jupyter notebook with two real-world datasets:

- Breast Cancer Wisconsin (Diagnostic) Database

- Preprocessed ICU data from PhysioNet Challenge 2012 Database

The learning objectives of these tutorial include:

- Learn how to use Google colab / Jupyter notebook
- Learn how to build and diagnose machine learning models for clinical classification and clustering tasks

In part 1, we will go through the basic of machine learning concepts through classification problems. In part 2, we will go deeper into unsupervised learning methods for clustering and visualization. In part 3, we will discuss more about deep neural networks. Please check the link of tutorials in the Appendix.

6. Conclusion

In summary, machine learning is an important and powerful technique for healthcare research. In this chapter, we have shown readers how to reframe a clinical problem into appropriate machine learning tasks, select and adjust an algorithm for model training, perform model diagnostics and error analysis, as well as model results and interpretation. The concepts and tools described in this chapter aim to allow the researcher to better understand how to conduct a machine learning project for clinical predictive analytics.

Programming Tutorial Appendix

The tutorials mentioned in this chapter available in the GitHub repository: `https://github.com/ckbjimmy/2018_mlw.git`.

References

Abu-Mostafa, Y., Lin, H., and Magdon-Ismail, M. *Learning from data: a short course*. AMLbook, 2012.

Bejnordi, B. E., Lin, J., Glass, B., Mullooly, M., Gierach, G. L., Sherman, M. E., Karssemeijer, N., Van Der Laak, J., and Beck, A. H. Deep learning-based assessment of tumor-associated stroma for diagnosing breast cancer in histopathology images. *ISBI*, 2017.

Breiman, L. Random forests. *Machine learning*, 45(1): 5–32, 2001.

Breiman, L. *Classification and regression trees*. Routledge, 2017.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., and Elhadad, N. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. *KDD*, 2015.

Chen, P.-H. C., Gadepalli, K., MacDonald, R., Liu, Y., Kadowaki, S., Nagpal, K., Kohlberger, T., Dean, J., Corrado, G. S., Hipp, J. D., et al. An augmented reality microscope with real-time artificial intelligence integration for cancer diagnosis. *Nature Medicine*, 25(9):1453, 2019a.

Chen, P.-H. C., Liu, Y., and Peng, L. How to develop machine learning models for healthcare. *Nature Materials*, 18(5):410, 2019b.

Choi, Y., Chiu, C. Y.-I., and Sontag, D. Learning low-dimensional representations of medical concepts. *AMIA CRI*, 2016:41, 2016.

Chung, Y.-A. and Weng, W.-H. Learning deep representations of medical images using siamese cnns with application to content-based image retrieval. *Machine Learning for Health (ML4H) workshop at NIPS 2017*, 2017.

Chung, Y.-A., Weng, W.-H., Tong, S., and Glass, J. Unsupervised cross-modal alignment of speech and text embedding spaces. *NeurIPS*, 2018.

Chung, Y.-A., Weng, W.-H., Tong, S., and Glass, J. Towards unsupervised speech-to-text translation. *ICASSP*, 2019.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Dernoncourt, F., Lee, J. Y., Uzuner, O., and Szolovits, P. De-identification of patient notes with recurrent neural networks. *JAMIA*, 24(3):596–606, 2017.

Doshi-Velez, F., Ge, Y., and Kohane, I. Comorbidity clusters in autism spectrum disorders: an electronic health record time-series analysis. *Pediatrics*, 133(1):e54–e63, 2014.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115, 2017.

Fonarow, G. C., Adams, K. F., Abraham, W. T., Yancy, C. W., Boscardin, W. J., Committee, A. S. A., et al. Risk stratification for in-hospital mortality in acutely decompensated heart failure: classification and regression tree analysis. *JAMA*, 293(5):572–580, 2005.

Freund, Y., Schapire, R., and Abe, N. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pp. 1189–1232, 2001.

Gehrmann, S., Dernoncourt, F., Li, Y., Carlson, E. T., Wu, J. T., Welt, J., Foote Jr, J., Moseley, E. T., Grant, D. W., Tyler, P. D., et al. Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives. *PLoS one*, 13(2):e0192360, 2018.

Ghassemi, M., Naumann, T., Doshi-Velez, F., Brimmer, N., Joshi, R., Rumshisky, A., and Szolovits, P. Unfolding physiological state: Mortality modelling in intensive care units. *KDD*, pp. 75–84, 2014.

Ghassemi, M., Naumann, T., Schulam, P., Beam, A. L., and Ranganath, R. Opportunities in machine learning for healthcare. *arXiv preprint arXiv:1806.00388*, 2018.

Girkar, U. M., Uchimido, R., Lehman, L.-w. H., Szolovits, P., Celi, L., and Weng, W.-H. Predicting blood pressure response to fluid bolus therapy using attention-based neural networks for clinical interpretability. *Machine Learning for Health (ML4H) Workshop at NeurIPS 2018*, 2018.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316 (22):2402–2410, 2016.

Henry, J., Pylypchuk, Y., Searcy, T., and Patel, V. Adoption of electronic health record systems among us non-federal acute care hospitals: 2008-2015. *ONC data brief*, 35:1–9, 2016.

Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science*, 313 (5786):504–507, 2006.

Hoerl, A. E. and Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Horng, S., Sontag, D. A., Halpern, Y., Jernite, Y., Shapiro, N. I., and Nathanson, L. A. Creating an automated trigger for sepsis clinical decision support at emergency department triage using machine learning. *PLoS one*, 12(4): e0174708, 2017.

Hsu, T.-M. H., Weng, W.-H., Boag, W., McDermott, M., and Szolovits, P. Unsupervised multimodal representation learning across medical images and reports. *Machine Learning for Health (ML4H) Workshop at NeurIPS 2018*, 2018.

Jolliffe, I. *Principal component analysis*. Springer, 2011.

Komorowski, M., Celi, L. A., Badawi, O., Gordon, A. C., and Faisal, A. A. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature Medicine*, 24(11):1716, 2018.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *NIPS*, 2012.

Lehman, E. P., Krishnan, R. G., Zhao, X., Mark, R. G., and Li-wei, H. L. Representation learning approaches to detect false arrhythmia alarms from ecg dynamics. *MLHC*, 2018.

Liu, G., Hsu, T.-M. H., McDermott, M., Boag, W., Weng, W.-H., Szolovits, P., and Ghassemi, M. Clinically accurate chest x-ray report generation. *MLHC*, 2019.

Liu, Y., Gadepalli, K., Norouzi, M., Dahl, G. E., Kohlberger, T., Boyko, A., Venugopalan, S., Timofeev, A., Nelson, P. Q., Corrado, G. S., et al. Detecting cancer metastases on gigapixel pathology images. *arXiv preprint arXiv:1703.02442*, 2017.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-sne. *JMLR*, 9(Nov):2579–2605, 2008.

Nagpal, K., Foote, D., Liu, Y., Wulczyn, E., Tan, F., Olson, N., Smith, J. L., Mohtashamian, A., Wren, J. H., Corrado, G. S., et al. Development and validation of a deep learning algorithm for improving gleason scoring of prostate cancer. *arXiv preprint arXiv:1811.06497*, 2018.

Pivovarov, R., Perotte, A. J., Grave, E., Angiolillo, J., Wiggins, C. H., and Elhadad, N. Learning probabilistic phenotypes from heterogeneous ehr data. *JBI*, 58:156–165, 2015.

Raghu, A., Komorowski, M., Celi, L. A., Szolovits, P., and Ghassemi, M. Continuous state-space models for optimal sepsis treatment-a deep reinforcement learning approach. *MLHC*, 2017.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484, 2016.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Weng, W.-H. and Szolovits, P. Mapping unparalleled clinical professional and consumer languages with embedding alignment. *2018 KDD Workshop on Machine Learning for Medicine and Healthcare*, 2018.

Weng, W.-H., Gao, M., He, Z., Yan, S., and Szolovits, P. Representation and reinforcement learning for personalized glycemic control in septic patients. *Machine Learning for Health (ML4H) workshop at NIPS 2017*, 2017a.

Weng, W.-H., Wagholikar, K. B., McCray, A. T., Szolovits, P., and Chueh, H. C. Medical subdomain classification of clinical notes using a machine learning-based natural language processing approach. *BMC medical informatics and decision making*, 17(1):155, 2017b.

Weng, W.-H., Cai, Y., Lin, A., Tan, F., and Chen, P.-H. C. Multimodal multitask representation learning for pathology biobank metadata prediction. *arXiv preprint arXiv:1909.07846*, 2019a.

Weng, W.-H., Chung, Y.-A., and Szolovits, P. Unsupervised clinical language translation. *KDD*, 2019b.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Xiao, C., Choi, E., and Sun, J. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *JAMIA*, 25(10):1419–1428, 2018.

Yala, A., Barzilay, R., Salama, L., Griffin, M., Sollender, G., Bardia, A., Lehman, C., Buckley, J. M., Coopey, S. B., Polubriaginof, F., et al. Using machine learning to parse breast pathology reports. *Breast cancer research and treatment*, 161(2):203–211, 2017.

Zou, H. and Hastie, T. elasticnet: Elastic net regularization and variable selection. *R package version*, 2005.