BMEG3105: Data Analytics for Personalized Genomics and Precision Medicine
Lecture 10: Perf Evaluation
Autumn 2024
Lecturer: LI Yu
Scribe: 1155191391

Outline of Lecture 10:

- Review on Lecture 9 (Slide: 0-19)
- Performance Evaluation (Slide: 20-33)
- Cross-validation (Slide: 34-46)
- Multi-class Classification (Slide: 47-51)
- Clustering Evaluation (Slide: 51-65)

I. Review on Lecture 9 (Slide: 0-19)

1. The Problem of KNN
   - Need to store all the data
   - Need to calculate the distance matrix
   - Predicting is slow

Hence, we need to find a formula

2. Logistic Regression

   2.1. Logistic function

$$\frac{1}{1+e^{-(w_h H + w_w W + w_0)}} \geq 0.5$$

$$w_h H + w_w W + w_0 \geq 0.5$$

Training: Fit the training data (To find wh, ww and w0)

Make $\frac{1}{1+e^{-(w_h H + w_w W + w_0)}} \geq 0.5$ correct for the training data.

Testing: Run the formula

2.2. Loss function

$$(Y^{output} - Y)^2$$

$$Y^{output} = \frac{1}{1+e^{-(w_h H + w_w W + w_0)}}$$

Where,
Y: the true label we have for training data
Loss function that we would like to **minimize**

2.3. Gradient descent algorithm

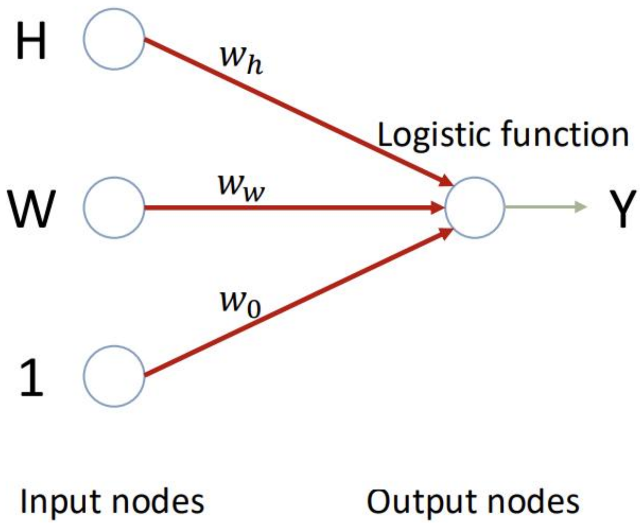
$$L = \sum_{P_1}^{P_4} (Y^{output} - Y)^2 \text{ is a function of } ws$$

We need to find a value to make the function value smallest
To get the formula:

- **Calculate the output Y^{output}**
- **Update weights**
- $w_i = w_i + \Delta w_i$
- $\Delta w_i = 2 * \alpha (Y - Y^{output}) \frac{\partial Y^{output}}{\partial w_i}$
- α **is a small constant**

Repeat the above steps until no more to update

2.4. Neural Network (NN)



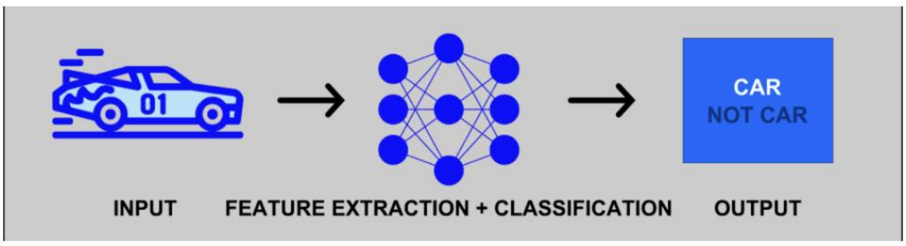
$$Y^{output} = \frac{1}{1 + e^{-(w_h H + w_w W + w_0)}}$$

2.5. From LR to NN

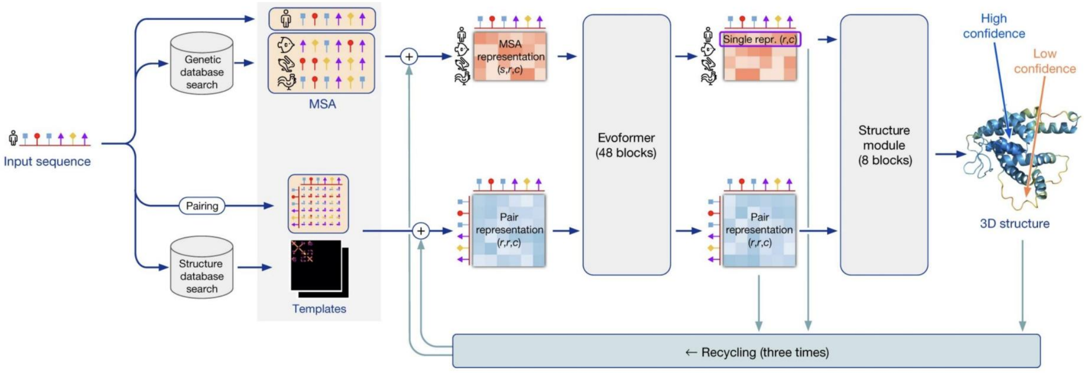
Pros: 1. Fast prediction
2. Successful in real-life problems
3. High tolerance to noisy data

Cons: 1. Long training time
2. Poor interpretability

2.6. From NN to Deep Learning



AlphaFold: the most successful deep learning application



II. Performance Evaluation (Slide: 20-33)

1. The purpose of model evaluation

Characterize the performance of a model:

- Pinpoint the strong points and weak points of a method
- Method selection/Model selection

2. Classification performance evaluation - Confusion matrix

		Predicted class	
		Class=Yes	Class=No
Actual class	Class=Yes	a(TP)	b(FN)
	Class=No	c(FP)	d(TN)

TP: True Positive
TN: True Negative
FP: False Positive
FN: False Negative

- **Most widely-used metric:**

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

- **Limitation: Maybe misleading for imbalanced data**

An example:

	Predicted class	
Actual class	**Class=Yes**	**Class=No**
Class=Yes	4949(TP)	0(FN)
Class=No	51(FP)	0(TN)

$$Accuracy = \frac{TP+TN}{TP+TN+\mathrm{FP}+TN} = \frac{4949}{4949+51} = 0.99$$

3. Classification performance evaluation - Precision, recall, and F1 score

	Predicted class	
Actual class	**Class=Yes**	**Class=No**
Class=Yes	a(TP)	b(FN)
Class=No	c(FP)	d(TN)

$$Precision = \frac{a}{a+c}$$

Among the predicted positive samples, how many of them are correct?

$$Recall = \frac{a}{a+b}$$

How many actual positive samples are predicted to be positive?

$$F1\ score = \frac{2*precision*recall}{presicion+recall}$$

The weighted average of precision and recall

- **Balanced Accuracy:**

	Predicted class	
Actual class		Class=Yes	Class=No
	Class=Yes	4949(TP)	0(FN)
	Class=No	51(FP)	0(TN)

$$Balanced\ accuracy = 0.5 * \left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) = 0.5$$

If already known it's an imbalanced dataset, look at the confusion matrix directly

4. Binary classification evaluation

- **Accuracy**
- **Precision**
- **Recall**
- **F1-score**
- **Balanced accuracy**
- **…**

III. Cross-validation (Slide: 34-46)

1. How to find a good K for KNN with the below data?

- **What is a good K?**

The K can give us good prediction accuracy

- **Problem:** we do not have the label for testing data

- **Solution:** use part of the training data as the testing data (Use each part one by one & Calculate the average over all the parts)

2. Cross-fold validation

   2.1. **Definition:** Cross-validation/rotation estimation, is a technique for assessing how the results of a machine learning analysis will generalize to an independent data set, i.e., A procedure to measure the performance of models

   2.2. **How the Cross-fold validation works?**
   One round of cross-validation involves partitioning a set of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the testing set)

2.3. Idea: train multiple times, leaving out a disjoint subset of data each time for validation. Average the validation set accuracy

2.4. Process:
Randomly partition data into n disjoint subsets
For i = 1 to n
1) Validation Data = i-th subset
2) h <- classifier trained on all data except for Validation Data
3) Accuracy(i) = accuracy of h on Validation Data

Final Accuracy = mean of the n recorded accuracies

2.5. Leave-one-out cross-validation
Idea: a special case of n-fold cross-validation, where n = N
Process:
Partition data into N disjoint subsets, each containing one data point
For i = 1 to N
1) Validation Data = i-th subset
2) h <- classifier trained on all data except for Validation Data
3) Accuracy(i) = accuracy of h on Validation Data

Final Accuracy = mean of the N recorded accuracies

IV. Multi-class Classification (Slide: 47-51)

1. For KNN: **Trivial**
   No need to change the algorithm

2. For logistic regression: need some change
   1) Build a logistic regression for each class
   2) When predicting, we assign class with highest value
   3) When training, we train 3*6=18 parameters

3. Multi-class evaluation: **Considering each class as a binary classification problem**
   Still using accuracy, precision, recall, F1 score and so on

V. Clustering Evaluation (Slide: 51-65)

1. Confusion Matrix:

	Predicted clusters		
Actual clusters		The same	Not the same
	The same	a(TP)	b(FN)
	Not the same	c(FP)	d(TN)

Where, a: the number of pairs are in the same cluster in the True clusters and also assigned to one cluster in the Predicted clusters

b: the number of pairs are in the same cluster in the True clusters and also assigned to different clusters in the Predicted clusters

c: the number of pairs are in different clusters in the True clusters and also assigned to one cluster in the Predicted clusters

d: the number of pairs are in different clusters in the True clusters and also assigned to different clusters in the Predicted clusters

2. Rand index, **R**

$$R = \frac{a+d}{a+b+c+d} = \frac{a+d}{Number\ of\ all\ the\ pair\ combinations}$$

$$Pairs = \binom{n}{2} = \frac{n*(n-1)}{2}$$

n: Total number of points