Linear Classification

Caleb Thian Jia Le
Miin Wu School of Computing
National Cheng Kung University
Tainan
NN6114035@gs.ncku.edu.tw.

***Abstract*—This report is about the implementation of linear classifier with different approaches. Discussion and comparisons on each approaches is made**

Keywords—Linear classifier, SVM

I. Introduction

This report would like to discuss several kind of approaches of linear classifier. The datasets used is *crx.csv* and *data.csv* with 15 features and 30 features respectively after preprocessing. The approaches to be discussed is listed below:

1. Linear classifier from scratch
2. Linear classifier with least-squared manner
3. Voted perception
4. Minimizing $\|w\|^2$—SVM hard margin
5. Minimizing $\|w\|^2$ with slack variable—SVM soft margin
6. SVM by sklearn

These approaches are compared by comparing their performance on these 2 datasets. The performance is evaluated by accuracy. The code is on the github[1].

II. PROBLEM SETTING

A. AIM

The two datasets mentioned is mainly considered. The first dataset crx.csv has 690 records, with 15 attributes and 1 label column marked as "label". This dataset has several attributes are categorical features. The second dataset data.csv has 569 records, with 31 attributes and 1 label column marked as "Diagnosis". These datasets will be analyzed by the linear classification of different approaches mentioned. Analyzed is made to compare these linear classification approaches but not to achieve the best performance.

B. Terminology

X is used to represent the features for classification, *Y* will be the ground truth, and *pred* will be the prediction made by linear classifier. *Acc* is the accuracy of responding linear classifier on the dataset.

III. Methodological Framework

Before talking about the linear classifier, data preprocessing is required. Data preprocessing will first to be discussed, then will be the linear classification.

A. Data Preprocessings

The dataset is preprocessed by the following steps:

1. Missing value handling: Drop every row with '?' or nan
2. Remove unused column: Discard any column that cannot use for classification, for example, ID.
3. Label encoding: Encode all the categorical features by label encoding, for example, a column with *a* and *b* is represented by *0* and *1*. Note that label column is represented as *-1* and *1* instead of *0* and *1*.
4. Column casting: Cast all the column data types to float.

B. Linear classification

The pseudocode of linear classification is:

repeat until convergence (or for some # of iterations):
  randomize order or training examples
  for each training example $(f_1, f_2, \ldots, f_n, label)$:
    $prediction = b + \sum_{i=1}^{n} w_i f_i$
    if $prediction * label \leq 0$: //then don't agree
      for each w_i:
        $w_i = w_i + lr * (f_i * label)$
      $b = b + label$

Figure 1: Pseudocode of Linear Classifier

About the terminology, *w* represented the weights (same throughout this report), *b* represented the bias (same throughout this report), *lr* represented the learning rate, which is set as 0.01, *f* represented the features for each data, *label* represented the label of the data, and *prediction* represented the classification result. The maximum iteration number is set as 20.

C. Linear Classifier with Least-Square Manner

To solve $\min_W J = WX + b$, solve the partial derivative of W. After the mathematics, $\widehat{W} = (X^T X)^{-1} X^T Y$.

D. Voted Perception

In voted perception, when training, every time a mistake is made on a data, store the weights before changing for current data and store the number of data that set of weights got correct. Then in classification, calculate the prediction from all saved weights and multiply each prediction by the number it got correct and take the sum over all prediction. The weights among the voted perception also came from the linear classifier with settings mentioned in method B.

E. Linear Classifier with minimum $||w||^2$ (SVM hard margin)

To get maximized margin, solve quadratic optimization problem, i.e., $\min_{w,b} ||w||^2$ subject to $y_i(w \cdot x_i + b) \geq C \; \forall i$, here take C=1. The problem is transformed for easier

[1] https://github.com/CalebThian/classification

solving[1]. Margin is compared at the same time with method B.

F. *Linear Classifier with minimum* $||w||^2$ *and slack variables(SVM soft margin)*

Similar with method E but with slack variables and most effective weighting value C is wished to be found. The objective is $\min_{w,b}\left|\left|w\right|\right|^2 + C\sum_i \zeta_i \, subject\ to\ y_i(w \cdot x_i + b) \geq 1\text{-}\zeta_i \forall i, \zeta_i \geq 0$. The objective is also transformed for easier solution [1]. Optimized C is searched from 0 to 1, precision by 0.01.

G. *SVM from sklearn*

Using SVM classifier from package SVM. For fair, we use all the data to fit the classifier instead of using part of data.

IV. EXPERIMENTS RESULTS WITH DISCUSSION

A. *Accuracy*

The accuracy of each method is showed below:

	Linear Classifier	Least-Square	Voted Perception	SVM hard margin	SVM soft margin	SVM sklea
0	0.676876	0.344564	0.584992	0.130168	0.453292	0.9984
1	0.882250	0.627417	0.882250	1.000000	0.372583	1.0000

Figure 2: Accuracy of each approach on the datasets

Note that 0 and 1 in row indicates the *crx.csv* and *data.csv*. By observation, least-square manner did not lead to a better result. Voted perception and linear classifier have similar accuracy. SVM by self-implementation may not converges, thus the accuracy should not be referred. On the other hand, SVM by sklearn has the best accuracy, indicates that SVM should be the best classifier.

B. *Margin*

The margin comparison between conventional linear classier and SVM hard margin is showed below:

	Linear classifier	SVM hard margin
0	0.018260	0.870416
1	0.004121	0.000041

Figure 3: Comparison of margin

Note that 0 and 1 in row indicates the *crx.csv* and *data.csv*, which is same as above. By SVM hard margin, it leads to larger margin in first dataset, but in second dataset, because both margin is small, so it may just a special case.

C. *Effective C*

The best C in SVM soft margin for both datasets are 0.01 and 0.01, which are the same. By observation, C is better when closer to 0.

REFERENCES

[1] Kevin, “[Notes] Machine Learning Techniques - Support Vector Machine (SVM)([筆記]機器學習技法-支持向量機(Support Vector Machine , SVM)),” 04 06 2020. [Website]. Available: https://ithelp.ithome.com.tw/articles/10231614?sc=rss.qu.