Experimental Report on Method and Evaluation of SVM Implemented by SMO

Yuxiang Zhang

I. Methods

In this experiment, we try to evaluate the results performed by different models. The selected models contains SVM using SMO algorithm, which is implemented by us, and two well-developed toolboxes: sklearn and libsvm. This section includes two parts, in the first part, we will briefly introduce the formulation of SVM that implemented by us, such as choosing the kernel trick. In the second part, we will present the SMO using in SVM.

A. Formulation of SVM

For the convenience of the following description, the matrix of input sample features is called $\mathbf{X}$, the matrix of sample labels is called $\mathbf{Y}$, and a single sample with features is called $\mathbf{S}$.

Our very original goal is to find a classifier function $f(\mathbf{S}) = \mathbf{X}^T\beta + \beta_0$ to realize binary classification task.in order to avoid the possible two-type inseparable problem, the relaxation variable C is introduced to find a soft margin hyperplane. Then, the original task of SVM is transfer into minimize $E(\beta)$ subject to $\forall\ \varepsilon_i \geqslant 0,\ y_i(\mathbf{X_i}^T\beta + \beta_0) \geqslant 1 - \varepsilon_i$. By using Lagrange multipliers, we get the dual problem: maximize $E(\alpha)$ subject to $\sum_{i=1}^{m}\alpha_i y_i = 0,\ 0 \leq \alpha_i \leq C$. Then, we use SMO algorithm to solve the duel problem, getting the α_i, β and β_0. Finally, we can gain the classification decision function.

$$E(\beta) = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\varepsilon_i. \tag{1}$$

$$E(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \mathbf{X_i^T X_j}. \tag{2}$$

Additionally, we used three kernel functions containing linear kernel, polynomial kernel and radial basis kernel, to convert input sample features to Kernel matrix $\mathbf{K}$, which could help simplify the calculation of the inner product after the sample is mapped to high-dimensional space. ($\mathbf{K_i}$ represents the ith column of $\mathbf{K}$, $\mathbf{K_{ij}}$ represents the ith row first j column of $\mathbf{K}$). Using Equation(3), Equation(4) and Equation(5) to generate the Kernel martrix for linear, polynomial kernel and radio basis kernel, respectively.

$$\mathbf{K_i} = \mathbf{X}\mathbf{S}^T. \tag{3}$$

$$\mathbf{K_{ij}} = \mathbf{K_{ij}}^k \quad (s.t.\ \mathbf{K_i} = \mathbf{X}\mathbf{S}^T). \tag{4}$$

$$\mathbf{K_{ij}} = exp(-\frac{\mathbf{K_{ij}}}{\sigma^2}) \quad (s.t.\ \mathbf{K_i} = (\mathbf{X_i} - \mathbf{S})(\mathbf{X_i} - \mathbf{S})^T). \tag{5}$$

Then, the final solution of SVM can be written as:

$$f(\mathbf{S}) = \sum_{i=1}^{N}\alpha_i y_i \mathbf{K}(\mathbf{X}, \mathbf{S}) + \beta_0 \tag{6}$$

B. Algorithmic Description of SMO

In this section, we will introduce the SMO algorithm used in SVM. Firstly, we define a function: $smo\_pre(\mathbf{X}, \mathbf{Y}, C, toler, i)$ (the detail is in Algorithm1). By using the smo_{pre} function in Algorithm2, we can get the well-trained lagrange multipliers and the β_0.

Algorithm 1 $smo\_pre(\mathbf{X}, \mathbf{Y}, C, toler, i)$

Input : $\mathbf{X}$, $\mathbf{Y}$, C, Tolerance: $toler$, Maximal training times: $maxIter$, Old threshold: $\beta_\mathbf{0}$, Old lagrange multipliers for solution: α, Number of samples: m.

Calculate $E_i = f(X_i) - Y_i$ using Equation(6)
if (Y_iE_i<-toler & α_i<C)or (Y_iE_i>toler & α_i>0) then
  Select $j(j \neq i)$ that α_j does not fulfill the Karush-Kuhn-Tucker conditions with a maximal $E_(j)$.
  Calculate $E_i = f(X_i) - Y_i$ using Equation(6).
  Save old α_i and α_j; $_i$ and $_j$, $s = Y_i \times Y_j$.
  if $Y_i = Y_j$ then
    L=max(0,α_i+α_j-C); H=min(C,α_i+α_j).
  else
    L=max(0,α_j-α_i); H=min(C,α_j-α_i+C).
  end if
  if L=H then
    return 0
  end if
  Calculate Kernel matrix using Equation(3), (4) or (5).
  $\eta = \mathbf{K}_{ii} + \mathbf{K}_{jj} - 2 \times \mathbf{K}_{ij}$
  if $\eta < 0$ then
    return 0
  end if
  $\alpha_{jn} = \alpha_j + Y_j \times \frac{(E_i - E_j)}{\eta}$.
  Renew α_j using min(α_j,H) and α_j=max(α_j,L).
  if $|\alpha_{jn} - old\ \alpha_j|$<0.00001 then
    return 0
  end if
  Renew α_{in}, using $\alpha_i + s \times (old\ \alpha_j - \alpha_{jn})$, E_i and E_j.
  Calculate β_1, β_2 using Equation(7) and(8), respectively.
  Renew $\beta_0 = \frac{1}{2} \times (\beta_1 + \beta_2)$.
  return 1
else
  return 0
end if

$$\beta_1 = Y_i\mathbf{K}_{ii}(\alpha_i - \alpha_{in}) + Y_j\mathbf{K}_{ij}(\alpha_j - \alpha_{jn}) + \beta_0 - E_i \tag{7}$$

$$\beta_2 = Y_i\mathbf{K}_{ij}(\alpha_i - \alpha_{in}) + Y_j\mathbf{K}_{jj}(\alpha_j - \alpha_{jn}) + \beta_0 - E_j \tag{8}$$

II. Experimental settings

The training dataset gp96.csv and test dataset gp97.csv come from Series GSE3494, which express signature for breast cancer. We use the entire trainting dataset to train the model. Noticed that that the features of some samples in the data set are missing, we preprocess the data by deleting the sample with feature missing, gaining a training dataset with 222 samples. In order to improve the training efficiency of model, we set the maximum number of iterations and the model will stop training when the parameters change to a certain extent. After comparing the prediction results of different

Algorithm 2 SMO for SVM

Input : $\mathbf{X}$, $\mathbf{Y}$, C, $toler$, $maxIter$, $\beta_\mathbf{0}$, m.
Output : New β_0 and New α

```
Creat a m × 1 matrix and initialize α_i, β_0 to 0.
alpha changes times: alphachange to 0
while alpha changes or entireSet = 1 do
    Initialize alphachange to 0.
    ∀ i < m, i ∈ ℕ
    alphachange+smo_pre(X, Y, C, toler, i)
    ∀ Index of α = 0
    alphachange+smo_pre(X, Y, C, toler, index)
end while
```

kernel functions, we choose the kernel function with better performance for subsequent experiment. Using gradient search to select the best hyperparameters, we compare the model with these hyperparameters with sklearn and libsvm. The environment we use is python 3.9, numpy 1.20.3, pandas 1.3.4, scikit-learn 0.21.2 and libsvm-official 3.25.0.

III. Results & Discussion

A. Performance of Different Model On Test Dataset

Firstly, we employ our implementation, sklearn and libsvm on test dataset, using the ROC curve, AUC value and Accuracy as evaluation indexes. We find that the accuracy of SMO SVM with linear or polynomial kernel is obviously lower than the radio basis kernel (linear 0.73, poly 0.82 and radio basis 0.93), so we will mainly discuss the result performed by radio basis kernel. Then, the hyperparamaters of SMO SVM are C and σ, we used gradient search to find the best pair with the accuracy of 0.93: {C=1, $\sigma = 3$} (see Fig.1). Secondly, we use the SVM

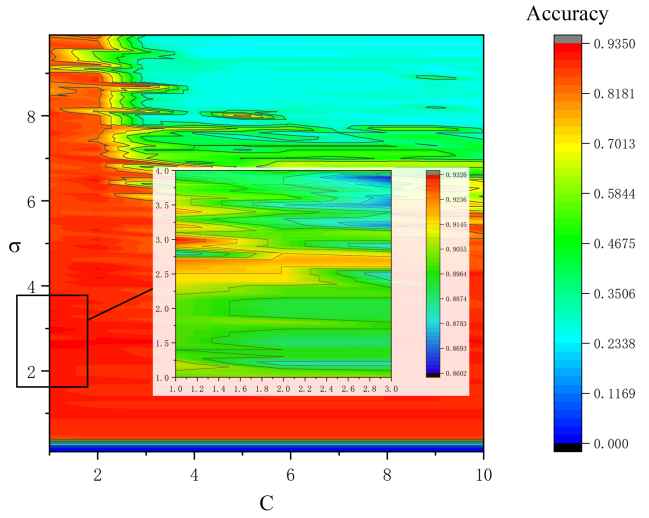

Fig. 1. The accuracy of SVM(SMO) using different C and $\boldsymbol{\sigma}$.

model in two well-developed toolboxes, sklearn and libsvm, employing on the same train and test dataset. The accuracy from largest to smallest is sklearn with radio basis kernel, libsvm with radio basis kernel, SMO with radio basis kernel, sklearn with linear kernel and libsvm with linear kernel. We also drew the ROC-cruve and compute the area under ROC-curve(AUC), the well-developed toolboxes both get a better performance comparing with our implementation(see Fig2).

Generally, we find that radio basis kernel is better than linear on the test dataset. Thus, we conclude that when the number

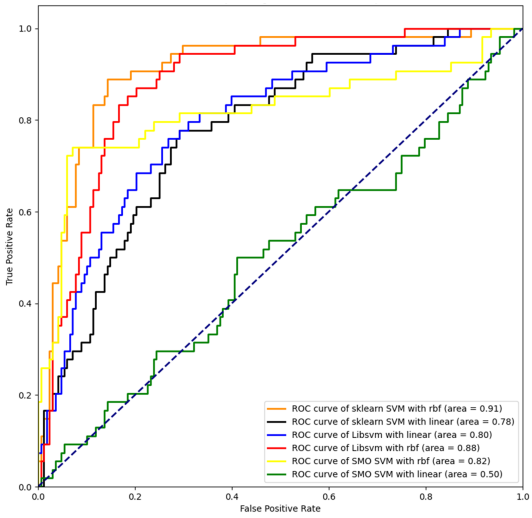

Fig. 2. ROC curve of different model perform on test dataset.

of samples is significantly larger than the number of features, a linear inseparable situation possibly occur, therefore, using a radio basis kernel may perform better at this circumstance.

B. Statistical Test of SMO SVM with Toolboxes

To compare whether there were significant differences between the predictive abilities of different models with radio basis kernel, we performed statistical tests for the two toolboxes and SMO SVM, respectively. Because of the use of the same dataset and the small number of models involved in the comparison, we used McNemar's test, which was tested with 95% confidence interval.

The null hypothesis H_0 is that "Two classifiers have similar error rate on the test dataset", while the alternative hypothesis H_1 is that "Two classifiers have different error rate on the test dataset".

	SMO T	SMO F
Sklearn T	**182**	**25**
Sklearn F	**10**	**182**

	SMO T	SMO F
Libsvm T	**171**	**36**
Libsvm F	**10**	**171**

Fig. 3. **2 × 2** contingency table of McNemar's test.

Using the 2 × 2 contingency table showing in Fig.3, the p-value of test between sklearn and SMO is 0.018; between Libsvm is 0.00023, both of them is small than 0.05, so we have sufficient evidence to reject H_0, the two well-developed toolboxes' predictive ability is significantly better than our implementation in 95% confidence interval.

References

[1] fengdu78, Support-vector-machine.ipynb, 2020, GitHub repository [Online]. Available: https://github.com/fengdu78/lihang-code/blob/master/7.support-vector-machine

[2] Coding Notes, The Simplified SMO Algorithm, 2018, Codeproject [Online]. Available: https://www.codeproject.com/Articles/1267445/An-Introduction-to-Support-Vector-Machine-SVM-and