Abstract

Objectives: In machine learning based human activity monitoring, the algorithm needs to produce a prediction model with a high accuracy. Support vector machine is one of the leading machine learning algorithms. Methods/Statistical Analysis: We propose an optimization approach of support vector machines that optimizes its regularization parameter to further improve its prediction accuracy in a human activity recognition application. In order to implement an efficient support vector machines predictive model of a particular dataset that would generalize well and have a good prediction performance, a suitable regularization parameter has to be applied in the regularization part of the equation. Findings: In order to empirically evaluate the effectiveness of our proposed approach, we show the results of our implementation and discuss the results of our proposed approach explained in the previous section on support vector machines models. From our experiments, we can see that we got fabulous results when the regularization parameter is 1000. For the accuracy on train/test dataset pair, we got a sufficiently high percentage for regularization parameter values of 10, 100 and 1000. And, the best cross validation accuracy is 98.8575, which is corresponding to a regularization parameter value of 1000. Additionally, we can also notice that the relation between the classification accuracy and the cross validation accuracy is proportional, and that is obvious in the accuracies responding to the regularization parameter of 0.0001, because both accuracies are significantly low. Improvements/Applications: Our idea was to replace the parameter value with a vector of parameter values and compare their results. It shows more improved and promising performance improvement but if we can apply parallel programming.

Keywords: Support Vector Machines, Regularized Support Vector Machines, Wireless Sensor Network, Human Activity Recognition, and Incremental Parameter Tuning

1. Introduction

Support Vector Machines algorithm is one of the best and most efficient machine learning algorithms for data classification. It was first proposed by [1]. What makes SVMs different and more efficient than regression analysis and others most of the time is the use of kernel trick which maps the inputs into higher-dimensional feature. Suppose the training data shown above is described by $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(l)}, y^{(l)})$, $x \in \mathbb{R}^n$, $y \in \{+1, -1\}$. The maximum margin hyperplane that is separating the two classes of data points is shown in Figure 1.

The maximum margin hyperplane shown above is obtained by maximizing the distance from the classifiers to the closest data points, which are called support vectors. Mathematically, the hyperplane is represented by
Logarithmic Incremental Parameter Tuning of Support Vector Machines for Human Activity Recognition

Where • denotes the dot product of $w$, the normal vector to the hyperplane, to the data point $x^{(i)}$. The distance for a data point $x$ and the hyperplane $w \cdot x + b = 0$ can be calculated by

$$d(x) = \frac{|w \cdot x + b|}{\sqrt{||w||^2}} = \frac{|w \cdot x + b|}{\sqrt{\sum_{i=1}^{d}w_i^2}}$$

However, our concern is the nearest data points only because they are the only points that affect the hyperplane, the decision boundary, and where it should be. Hence, we consider the data points with the minimum distance

$$\text{margin} = \min_{x \in D} d(x) = \frac{\min_{x \in D} |w \cdot x + b|}{\sqrt{\sum_{i=1}^{d}w_i^2}}$$

The support vectors of the two categories are then in line with the two hyperplanes $w \cdot x + b = +1$ and $w \cdot x + b = -1$. Now, the maximizing argument is formalized

$$\arg \max_{w,b} \text{margin}(w,b,D) = \arg \max_{w,b} \min_{x \in D} d(x) = \frac{\min_{x \in D} |w \cdot x + b|}{\sqrt{\sum_{i=1}^{d}w_i^2}}$$

From the argument of maximizing the margin above it is equivalent to

$$\min ||w|| \text{ subject to } y^{(i)}(w \cdot x^{(i)} + b) \geq 1, i = 1, 2, ..., m$$

Since minimizing $||w||$ is equivalent to minimizing $\frac{1}{2}||w||$, we can reformulate it to

$$\min \frac{1}{2} ||w||^2 \text{ subject to } y^{(i)}(w \cdot x^{(i)} + b) \geq 1, i = 1, 2, ..., m$$

The use of this term enables us to perform Quadratic Programming (QP) optimization with the use of Lagrange multipliers $\alpha$ for the above minimization problem. So far, we assumed that the data we are dealing with linearly separable. However, any real data is susceptible to outliers. SVM has a regularization form to avoid producing a predictive model that won’t efficiently predict future data. Even when data is linearly separable, a single outlier can cause a dramatic swing to the decision boundary or hyperplane with a much narrower margin if regularization is not applied.

Regularization is applied to SVM to make it less sensitive to outliers. We introduce $C$ which is the number of train errors and it is then multiplied with the distances of error points to their right category. $C$ is basically the regularization parameter of SVM. Support vector machines algorithm uses $l_1$ regularization. The maximization of the hyperplane margin by minimizing $||w||$ is hence reformulated as follows and solved with QP as mentioned earlier for normal SVM with the use of Lagrange multipliers as well.

2. Regularization Parameter Tuning

Support vector machines algorithm is a workhorse method of machine learning and it is extensively used in classification but it always has an optimization problem to be solved. Several studies conducted to tackle this optimization problem as follows.

2.1 Bias–Variance Trade-off

First of all, we need to understand the problem to approach it. It is a modeling error occurs either when a predictive function is too closely fit to the limited training dataset or when it is so unfit and doesn’t represent the data in anyway. Hence, they are also called over-fitting and under-fitting. In the case of over-fitting occurrence, the model basically takes the form of making an overly complex model to closely represent the data under study. On the other hand, under-fitting is the total opposite which when the model has nothing to do with the data and doesn’t represent it in any way. In case of occurrence of any of these two cases, the model would be badly infected with substantial errors and severely harms its predictive power. This can also explain the importance of having a test set
that has never been seen by the algorithm during the process and test the accuracy of the model with that set. Obviously, what a predictive model needs to be is something in between. A model that is not so complex and powerful that it faithfully models the noise accompanied with this particular data sample but it is powerful enough to represent the underlying structure of the data under study.

One of the most effective and easiest methods of avoiding bias-variance trade off is by dividing the data we have into two sets; a training set which is used to train the model and a validation set which is used to test the performance of the model, the time that the model provides the lowest error rate with the validation set whereas it keeps getting better results on the training dataset would be the perfect time for the model to stop learning.

Another way that is proven to prevent over-fitting is to introduce a small amount of noise added each input of the training set with a mean value of zero. This may seem a rather stupid thing to do, to intentionally corrupt your own data but we can see that it is difficult now for the learning algorithm to pick up any specific data point too closely. Practically, training with added noise has indeed shown to reduce the occurrence of high variance and improving generalization sometimes.

Last and most important popular method of preventing under and over-fitting is regularization. It is a general concept that’s used in statistics. It is applied by introducing an additional penalty to the cost function for the purpose of smoothing the data examples considered by keeping all the features but reducing the magnitude or the values of the parameters of some features. It is used in several other machine learning algorithms too. Regularization, also called weight decay.

2.2 Cross Validation

The most popular used method of selecting the regularization parameter is cross validation (CV). It is basically to partition the dataset into two parts: training data and testing data. The dataset is split into K subsets. And the K subsets will be retrained as validation dataset to estimate the model fitness whiles the others, K-1; subsets are applied for fitting the model. This process is to be repeated for K times, whereas there will be no subset that is used more than once as a validation data. This process will produce different values of tuning parameter and obviously the parameter that produces the optimal performance of the predictive model will be preferred.

K-fold Cross Validation (K-FCV) is the most common type and 10-fold is the most commonly used in this type. Then comes the 2-fold cross validation which is the simplest variation of k-fold CV, it is also sometimes called the holdout method. It obviously divides the data into two sets equally but the data is usually shuffled before splitting it and the same process is carried out.

Another method of performing cross validation generalization technique is to split the data randomly and repeatedly hence comes the name repeated random subsampling validation. The only drawback of this method is since the splitting is performed randomly, some observations may never be selected to act as a validation subsample while some other observations may never be selected.

Leave-one-out cross validation (LOOCV) is the last method of performing cross validation which is to take each example or observation once and use it as a validation set and the rest to learn the model as a training set repeatedly over the whole set of observations. Basically, it is a k-fold cross validation with K being the number of the observations under study.

3. Proposed Optimization Approach

In order to implement an efficient support vector machines predictive model of a particular dataset that would generalize well and have a good prediction performance, a suitable regularization parameter has to be applied in the regularization part of the equation. Often, it is selected manually or using k-fold cross validation explained above and judging by the prediction accuracy of the generated hypothesis function on the test set.

We proposed two approaches; one of them is vectorizing the value of the regularization parameter fed to the regularized cost function and the other approach is to apply a vector that has preset values. The following two sections will explain them both.
3.1 Vectorized Regularization Parameter

\[ C = \left[ \begin{array}{c} C_{\text{start}} \\ C_{\text{start}+\text{step}} \\ \vdots \\ C_{\text{end}+\text{step}} \\ C_{\text{end}} \end{array} \right] \]

The implementer has to simply specify three mentioned values \( C_{\text{start}} \), \( C_{\text{end}} \) and \( C_{\text{step}} \). The first value, \( C_{\text{start}} \) to specify the first value of the vector which to be processed by the cost function that to be minimized; the second value, \( C_{\text{end}} \) is to specify the last value of the vector to be processed by the algorithm and \( C_{\text{step}} \) specifies the resolution or the difference between two adjacent regularization parameter values in the vector that is processed by

\[
\min \frac{1}{2} \|w\|^2 + \sum_{i=1}^{m} \varepsilon_i
\]

s.t. \( y_i^{(l)}(w^T \cdot x_i^{(l)} + b) \geq 1 - \varepsilon_i \), \( l = 1, 2, ..., m \)

\[ \varepsilon_i \geq 0, i = 1, ..., m \]

In the vectorized minimizing function above, it is shown how the cost values are calculated by substituting the vector of regularization parameter values instead of a single parameter value. And then, we would look for the cost with the least value. Let’s say it is the \( t \)-th element referred to as \( \|w\|_l \), we would then calculate the optimized \( \lambda \) by the simple equation \( C_{\text{opt}} = C_{\text{start}} + t C_{\text{step}} \). After finding \( C_{\text{opt}} \), we can build our optimized permanent prediction function with an optimized value of the regularization parameter over the range of the values applied, which satisfies \( C_{\text{opt}} = \arg \min \|w\|_l \).

3.2 Logarithmic Incremental Vectorizing

The other proposed approach is a customized way of the first approach proposed above. It eliminates the need of the implementer to specify the properties of the vector which will be applied to the regularized minimizing function. The starting parameter, the last parameter as well as the resolution are preset by the algorithm in a logarithmic scale to cover a very broad range; which is an advantage for the algorithm to be applicable and computationally effective with datasets that have large number of features. The idea of this approach is to do the same process of applying a vectorized value of the regularization parameter but in a logarithmic incremental scale. Here, the process of the first approach is to be repeated four times with a smaller size. Each time, the broad range goes narrower depending on the level of performance. The following Figure 2 explains how the values of the regularization parameter vector are pre-set and how it the range narrower range will be applied with each iteration.

![Figure 2. Logarithmic incremental technique.](image)

As we can see in the Figure 2 above, we will have four runs or iterations. The first iteration is the most important where it covers a very broad range of parameters where it starts with zero and increments in a logarithmic scale till 10,000, \( \{0, 0.01, 0.1, 1.0, 10, 100, 1000, 10000\} \). For instance, in the figure above, we show that the performance was optimized at the value of 100. It is important to mention that the performance is tested on observations from the cross validation set that have not been seen by the learning algorithm in training the model. After that, we will have the second iteration which will have ten values starting from half of the parameter value in the first iteration and increments by 50 until half of the value of the next element in the logarithmic scale as shown below.

First element, Last element

The second iteration will be processed with a vector limited from and to the values shown above. As shown in our example, the optimized choice of our second iteration is 150. The next third iteration will be processed with a vector of 9. The reason that the vector elements are reduced is because we don’t need to test the middle value of the intervals before and after the optimized parameter
as shown below because if it has the best performance of
the next iteration, we would have two optimized param-
eters in the second iteration and it would be obviously
the middle value and it would be finalized otherwise as
shown (Figure 3).

![Figure 3](image.jpg)

Finally, we will have a vector of five elements. It is kind
of obvious since it is the five values in the interval of the
previous iteration.

4. Experimental Results and Discussions

In order to empirically evaluate the effectiveness of
our proposed approach, we will show the results of our
implementation and discuss the results of our proposed
approach explained in the previous section on support
vector machines models. In the following subsection,
firstly, we will explain the public benchmark dataset we
have used to evaluate our approach. Then, we will pres-
ent the experimental results and further investigate the
implication of the results.

4.1 Smartphone Dataset Based Human
Activity Recognition (HAR)

For human activity recognition using smartphones, a a-
carried out experiments with a group of thirty volunteers
within an age range of 19-48 years. They constructed
Human Activity Recognition dataset from the record-
ings of the thirty subjects. The thirty subjects basically
had performed activities of daily living (ADL). And while
the subjects were performing ADL, they carried a waist-
mounted Smartphone. The waist-mounted Smartphone
included inertial sensors to record motion and activity
data. The Smartphone used is Samsung Galaxy S II.

We explain the data gathering stage in detail. Each
subject behaves six activities while wearing a Smartphone
on the waist. The six activities are SITTING, STANDING,
WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, and LAYING. Using the Smartphone’s
embedded accelerometer and gyroscope at a constant rate
of 50Hz, a a-collected 3-axial angular velocity and 3-axial lin-
ear acceleration. They video-recorded their experiments
and then label the data manually. They partitioned the
obtained dataset randomly partitioned into two sets. The
seventy percent of the subjects were selected for training
data generation, and thirty percent of the subjects were
selected for test data generation.

For dataset generation, a a-pre-processed sensor
signals from accelerometer and gyroscope. This data pre-
processing was performed by applying noise filters. After
that they sampled the data using a fixed-length sliding
window of 2.56 sec, allowing fifty percent overlap (i.e. 128
readings/window).

The collected sensor acceleration signal includes
body motion components and gravitational com-
ponents. A Butterworth low-pass filter was applied
to separate the signal into gravity and acceleration. In
assumed that the gravitational force had low fre-
quency component, therefore they applied a 0.3 Hz
cutoff frequency filter. For each window, a feature
vector was generated by measuring variables from the
frequency and time domain.

In each record of the final benchmark dataset, the fol-
lowing attributes were incorporated:

- Activity label.
- Triaxial acceleration and the estimated body accelera-
tion from the accelerometer.
- Triaxial Angular velocity from the gyroscope.
- A 561-feature vector containing frequency and time
domain variables.
- A subject identifier to denote who performed the
experiment.

Now, we would like to apply our classification algo-
rithm to this data and optimize the regularization
parameter to be used to achieve the highest performance
possible.

Table 1 shows the results of our proposed opti-
mization approach. We can see that we got fabulous
results when the regularization parameter is 1,000. For
the accuracy on train/test dataset pair, we got a suf-
ciently high percentage for regularization parameter
values of 10, 100 and 1000. And, the best cross valida-
tion accuracy is 98.8575, which is corresponding to a
regularization parameter value of 1000. Additionally,
we can also notice that the relation between the
classification accuracy and the cross validation
accuracy is proportional, and that is obvious in the
accuracies responding to the regularization parameter of 0.0001, because both accuracies are significantly low.

Table 1. Results on Human activity recognition using Smartphone’s dataset

| Regularization Parameter | Train/Test Accuracy % | Cross Validation Accuracy % |
|---------------------------|------------------------|-----------------------------|
| 0.0001                    | 18.2219                | 19.1376±1.18                |
| 0.001                     | 18.2219                | 19.1376±1.18                |
| 0.01                      | 47.7435                | 37.7040±1.46                |
| 0.1                       | 86.8341                | 88.5201±0.96                |
| 1                         | 94.0278                | 94.9810±0.66                |
| 10                        | 95.4530                | 98.2046±0.40                |
| 100                       | 96.4710                | 98.7894±0.33                |
| 1000                      | 96.6067                | 98.8575±0.32                |

Note that there are train/test pair of datasets provided in the Human Activity Recognition dataset. The number of instances in the training dataset is 7,352 and the number of instances in the test dataset is 2,947. Also, in the table 1, for the cross-validation accuracy, we performed 10-fold cross validation and calculated confidence interval with 95% confidence level.

In8 from Stanford University proposed an optimization approach in their paper ‘Efficient L1 Regularized Logistic Regression’ in 20068. They proposed a reformulation of Newton’s method as a weighted ordinary least squares problem considering the problem of finding the maximum likelihood estimate (MLE) of the parameters \( \theta \). First of all, their approach is for logistic regression, while our approach is for SVM. And although their approach is mathematically well established, it cannot be applied in parallel, as our approach. We will explain the parallel approach in more detail in the future work.

In7 carried out experiments for Human Activity Recognition Using Smartphone’s. They used multi-class hardware-friendly support vector machines. Their method adapted the standard Support Vector Machine (SVM) and exploited fixed-point arithmetic for computational cost reduction. In our proposed approach, we directly try to optimize the regularization parameters.

In9 performed sensor-based activity recognition for OPPORTUNITY Activity Recognition Dataset10. They basically presented a baseline activity recognition system on the OPPORTUNITY dataset. In this paper, we performed experiments on Human Activity Recognition Using Smartphone’s Dataset by2. We applied optimization of support vector machines algorithm by optimizing its regularization parameters.

5. Conclusion and Future Work

In this paper, we focused on optimizing one of the most efficient and increasingly popular machine learning algorithms, support vector machines. We conclude this paper with a summary of its contribution and we will talk about suggested future direction on regularization parameter optimization.

No one can deny that machine learning is attracting more and more attention from researchers as well as industry. In this paper, we studied regularization on support vector machines. We explained regularization and its role in support vector machines. After that, we discussed few approached used for optimizing these predictive models. Then, we explained our proposed approach in optimizing the regularization parameters used. Lastly, we showed our implementation results and discussed them.

We proposed an optimizing approach to determine the best regularization parameter that would result the highest performance accuracy possible. The idea was to replace the parameter value with a vector of parameter values and compare their results. The only drawback of this method is that it might be computationally expensive, definitely more expensive to process a vector than a single value. Although, it shows promising performance improvement but if we can apply parallel programming, we can reduce its execution time. We show how in the Figure 4 based on the example discussed in Figure 2.

Figure 4. Logarithmic incremental technique.
As we can see Figure 4, applying this method can theoretically reduce our execution time by 40%. This would be the next future step of this research.

6. Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (no. NRF-2015R1D1A1A01061328).

7. References

1. Vapnik V. An Overview of Statistical Learning Theory. IEEE Transactions On Neural Networks. 1999; 10(5):988-99.
2. Fletcher T. Support Vector Machines Explained. Tutorial paper-PhD. 2008; p. 1-19.
3. Kearns M, Vazirani U. Park Plaza, Boston, MA: PWS publishing Company: An Introduction to Computational Theory. 1997; p. 410.
4. Tibshirani R. Regression Shrinkage and Selection via the LASSO. Journal of the Royal Statistics Society, Series b (Methodological). 1996; 58(1):267-88.
5. Draper N, Smith H. New York: Wiley: Applied Regression Analysis, 3rd (edn). 1998; p. 736.
6. Refaeilzadeh P, Tang L, Liu H. Indian Institute of Technology: Statistical Machine Translation. Course Material. 2008.
7. Anguita D, Ghio A, Oneto L, Parra X, Reyes-Ortiz J. Human Activity Recognition on Smartphone’s using a Multiclass Hardware-Friendly Support Vector Machine. Spain, Vitoria-Gasteiz: International Workshop of Ambient Assisted Living (IWAAL 2012). 2012 Dec; p. 1-8.
8. Lee S, Lee H, Abbeel P, Ng A. Efficient L1 Regularized Logistic Regression. Massachusetts, USA: the proceeding of the national conference on artificial intelligence. 2006; p. 401-08.
9. Chavarriaga R, Sagha H, Calatroni A, Digumarti S, Troster G, Millan J, Roggen D. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. Pattern Recognition Letters. 2013; 34(15):2033-42.
10. Roggen D, Calatroni A, Rossi M, Holleczek T, Troster G, Lukowicz P, Pirkl G, Bannach D, Ferscha A, Doppler J, Holzmann C, Kurz M, Holl G, Chavarriaga R, Sagha H, Bayati H, Millan J. Collecting complex activity data sets in highly rich networked sensor environments. Kassel, Germany: Seventh International Conference on Networked Sensing Systems (INSS’10). 2010; p. 233-40.