

A PREDICTION OF RAINFALL DATA BASED ON SUPPORT VECTOR MACHINE WITH STOCHASTIC GRADIENT DESCENT

Dian Puspita Hapsari¹, Mohammad Imam Utoyo², Santi Wulan Purnami³

¹ Hapsari Institute of Technology Adhi Tama, Surabaya, Indonesia, dian.puspita@itats.ac.id

² Utoyo Airlangga University, Surabaya, Indonesia, m.i.utoyo@fst.unair.ac.id

³ Purnami Institute of Technology Sepuluh Nopember, Surabaya, Indonesia, santi_w@statistika.its.ac.id

ABSTRACT

Predicting rainfall is part of the weather forecast, an activity that is difficult to do because it depends on various variables. The value of the dependent variable such as wind speed and direction, temperature, humidity, varies from time to time and the weather calculation varies according to geographical location and atmosphere. An accurate prediction model is needed to predict rainfall using Support Vector Machine with Stochastic Gradient Descent (SGD-SVM) to replace the linear threshold used in traditional rainfall prediction activities. The selection of the right model parameters has an important impact on the accuracy of SVM predictions, and SGD is proposed to find the optimal parameters for SVM. The SGD-SVM algorithm is used in model training activities using historical data for rainfall prediction, which can be useful information and is used by governments and communities in making wise and smart decisions. The simulation shows that the proposed prediction model has an algorithm performance with much better accuracy than the traditional prediction model. On the other hand, the simulation results show the effectiveness of the SVM-SGD model used in machine learning and further promise scope for improvement because more and more relevant attributes can be used in predicting the dependent variable.

Key words: Support Vector Machine, Stochastic Gradient Descent, Predicting Rainfall.

1. INTRODUCTION

Climate change creating extreme weather events that are more frequent and more intense in certain locations around the globe. There is evidence heat waves have intensified, which has contributed to accelerating drought and extreme flood events. With few exceptions, the general phenomenon is that rainfall intensity has increased, but with a reduced number of wet days. Studies that associate rainfall and temperature are scarce, rainfall extremes have been studied more extensively than temperature extremes (Naveendrakumar *et al.*, 2019). In the last 20 years the use of machine learning in rainfall prediction activities has been carried out. It is applied in several areas and proven effective when solving those problems. Begins with clustering data model developed to overcome the problems of time, the speed of conversation used and for clustering large data sets (Hartono *et al.*, 2018). The selection of the classifier has a crucial impact on the accuracy and efficiency. Artificial neural network (ANN) can be used to predict the occurrence of rainfall in a short span of time. Models produced from ANN have peak performance in well-defined seasons, but lose their accuracy in the transition season (Esteves *et al.*, 2019).

Using the SVM algorithm for accurate rainfall-runoff modeling has been done on the precipitation forecast there is still some open research for improving the performance of the systems. The algorithm shows many unique advantages in solving nonlinear, high dimensional pattern recognitions and large scale data (Sehad *et al.*, 2017). The learning problem of SVM can be expressed as a convex optimization problem, so we can find the global minimum of the objective function by using the known effective algorithm (Young *et al.*, 2017). We propose a prediction model for rainfall forecasts based on Support Vector Machine with Stochastic gradient descent for optimization. Types of optimization algorithms for minimize a loss function we used First Order Optimization Algorithms - Gradient descent method is commonly used to train classifiers by minimizing the error function (Bottou, 2010). Bottou were introduce the classification optimization with the gradient method for large-scale data training and identify how optimization problems arise in machine learning and what makes them challenging.

We have looked for training algorithms that have a short training time property (linear scaling with training set size) but have high generalization accuracy. Support Vector Machines (SVMs) as one of the supervised learning technique algorithms that can be applied in many cases. That is the reason why we try to present most of our algorithms in general to facilitate the conception of derivation for large scale applications.

This study uses an online BMKG database for classification with 2160 lines, 9 attributes such as Tn: Minimum temperature (° C), Tx: Maximum temperature (° C), Tav: Average temperature (° C), RH_avg: Average humidity (%), RR: Rainfall (mm), ss: The duration of solar radiation (hours), ff_x: Maximum wind speed (m / s), ddd_x: Wind direction at maximum speed (°), ff_avg: Average wind speed (m / s), ddd_car: Most wind direction (°). This

data was collected from the Perak Maritime Meteorological Station II which conducted daily records. The binary label is based on the column the amount of rainfall measured, if there is a value there will be rain, if there is no value then there will be no rain.

In the literature, different variants of gradient descent (GD) and stochastic gradient descent (SGD) have been suggested with the aim of increasing performance in terms of accuracy and convergence speed (Wijnhoven and De With, 2010). In both methods, parameters are updated in an iterative manner to minimize the objective function (Sopyła and Drozda, 2015). To deal with large scale datasets and to factorize large scale matrices, different algorithms have been proposed. Stochastic gradient descent (SGD) algorithms much simple and efficient and are extensively used for matrix factorization (Sakr *et al.*, 2017).

This paper is organized as follows: in the following section, an introduction to SVM and SGD is presented. Section 2 presents the proposed SVM-SGD algorithm. In Section 3, we describe the experiments for prediction of rainfall dataset. Section 4 reports experiments or discussion of our model including comparative results between the traditional SVM-based prediction models and ours. Section 5, last section states our conclusions.

2. SVM BASED ON SGD

2.1 Support vector machines

Support vector machines are supervised learning models that analyzed data used for classification and regression analysis. The original SVM algorithm was invented by Vladimir N. Vapnik and Alexey Ya. Chervonenkis in 1963 (Chervonenkis, 2013). In 1992, Bernhard *et al.* (2004) suggested a way to create nonlinear classifiers by applying the kernel trick to maximum-margin hyperplanes. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall.

For linear classifier SVMs can efficiently perform a non-linear classification using kernel trick, mapping their inputs into high-dimensional feature spaces. SVM tries to find the hyperplane. We will call this the optimal hyperplane, and we will say that it is the one that best separates the data.

The SVM optimization problem is:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} \|w\|^2 \\ \text{subject to} \quad & y_i(w \cdot x_i) + b \geq 1, \quad i = 1 \dots m \end{aligned} \tag{1}$$

This function is convex in w . A function is convex if for every \mathbf{u}, \mathbf{v} in the domain, and for every $\lambda \in [0,1]$ we have $f(\lambda\mathbf{u} + (1 - \lambda)\mathbf{v}) \leq \lambda f(\mathbf{u}) + (1 - \lambda)f(\mathbf{v})$. For convex functions in general the function f is $\nabla f(x) = 0$ for x to be a minimum, the stationary condition tells us that the selected point must be a stationary point. It is a point where the function stops increasing or decreasing. When there is no constraint, the stationary condition is just the point where the gradient of the objective function is zero. When we have constraints, we use the gradient of the Lagrangian (Luo and Yu, 2006). Primal feasibility condition, looking at this condition, you should recognize the constraints of the primal problem. It makes sense that they must be enforced to find the minimum of the function under constraints, Dual feasibility condition. Similarly, this condition represents the constraints that must be respected for the dual problem.

This formulation of the SVM is called the hard margin SVM. It cannot work when the data is not linearly separable. There are several Support Vector Machines formulations. In the next chapter, we will consider another formulation called the soft margin SVM, which will be able to work when data is non-linearly separable because of outliers. Minimizing the norm of w is a convex optimization problem, which can be solved using the Lagrange multipliers method. Some researchers have discovered new heuristics to improve this algorithm, and popular libraries like LIBSVM use an SMO-like algorithm. Note that even if this is the standard way of solving the SVM problem, other methods exist, such as gradient descent and stochastic gradient descent (SGD) (Wijnhoven and De With, 2010), which is particularly used for online learning and dealing with huge datasets (Xu *et al.*, 2015).

2.2 Gradient Descent

Gradient Descent is a popular optimization technique in Machine Learning and Deep Learning and it can be used with most, if not all, of the learning algorithms. A gradient is slope of a function, the degree of change of a parameter with the amount of change in another parameter. Mathematically, it can be described as the partial derivatives of a set of parameters with respect to its inputs. Gradient Descent is a convex function (Wu *et al.*,

2011). Gradient Descent can be described as an iterative method which is used to find the values of the parameters of a function that minimizes the cost function as much as possible. The parameters are initially defined a particular value and from that, Gradient Descent is run in an iterative fashion to find the optimal values of the parameters, using calculus, to find the minimum possible value of the given cost function (Lin *et al.*, 2011).

Gradient descent is an iterative algorithm, the starts from a random point on a function and travels down its slope in steps until it reaches the lowest point of that function. This algorithm is useful in cases where the optimal points cannot be found by equating the slope of the function to 0. The general idea is to start with a random point (in our parabola example start with a random “x”) and find a way to update this point for each iteration such that we descend the slope. We are trying to minimize,

$$J(w) = \min_{w,b} \frac{1}{2} w^T w \quad (0,1 - y_i w^T x_i) \quad (2)$$

The steps of the algorithm are: First, find the slope of the objective function with respect to each parameter/feature. In other words, compute the gradient of the function. Second, pick a random initial value for the parameters. (To clarify, in the parabola example, differentiate “y” with respect to “x”. If we had more features like x1, x2 etc., we take the partial derivative of “y” with respect to each of the features.) Third, update the gradient function by plugging in the parameter values. And last calculate the step sizes for each feature as step size = gradient * learning rate. Calculate the new parameters as new params = old params -step size. The repeat steps 3 to 5 until gradient is almost 0. The “learning rate” mentioned above is a flexible parameter which heavily influences the convergence of the algorithm (Schraudolph *et al.*, 2007).

Larger learning rates make the algorithm take huge steps down the slope and it might jump across the minimum point thereby missing it. So, it is always good to stick to low learning rate such as 0.01. It can also be mathematically shown that gradient descent algorithm takes larger steps down the slope if the starting point is high above and takes baby steps as it reaches closer to the destination to be careful not to miss it and also be quick enough. There are a few downsides of the gradient descent algorithm (Zhang, 2004). We need to take a closer look at the amount of computation we make each iteration of the algorithm. We have 10,000 data points and 10 features. The sum of squared residuals consists of as many terms as there are data points, so 10000 terms in our case. We need to compute the derivative of this function with respect to each of the features, so in effect we will be doing $10000 * 10 = 100,000$ computations per iteration. It is common to take 1000 iterations; in effect we have $100,000 * 1000 = 100000000$ computations to complete the algorithm.

That is pretty much an overhead and hence gradient descent is slow on huge data. In gradient descent, we compute the gradient using the entire training set. A superficially simple (but in fact far-reaching) alteration of this is to find the gradient with respect to a single randomly chosen example. This technique is called stochastic gradient descent (SGD).

3. EXPERIMENT

3.1 Data Preprocessing

Data preprocessing is the initial stages of the machine learning process. Because only valid data will produce accurate output, data preprocessing is the key stage. For this study, we use the one-year ground-based meteorological data from Maritim Perak Station (ID WMO: 96937). The dataset contains atmospheric pressure, sea level pressure, wind direction, wind speed, relative humidity and precipitation. Data is collected every day in one year. We just consider related information and ignore the rest. The method of principal component analysis (PCA) is used to reduce the dimensionality of the data, thus reducing the data processing time and improving the efficiency of the algorithm. We performed data transformation on rainfall. By observing the original data set, we find the incorrect data in the data set, which does not correspond to the fact.

3.2 Train SVM-SGD

For the assessment of the algorithms results, cross-validation and external testing were carried out. The datasets were divided into two subsets, training and test, comprising 80% and 20% of the original samples, respectively. Training set take 80% of the samples randomly from the dataset as the training set. Test set we used other data remaining in the data set, which contained all the attributes except the rainfall data that the model is supposed to predict. The test set was never used for the training of any of the models.

Table 1. Original Meteorological-Rainfall Data.

Tn	Tx	Tavg	RH_avg	RR	ss	ff_x	ddd_x	ff_avg
26	28,8	25,8	92	10,6	2	3	340	2
24	33,6	28	79	18	0	8	330	4
25	33,7	29,1	78	0	5,3	8	340	4
26	32,4	28,7	79	4	7,3	5	300	2
27	34,6	30	77	2,5	0	9	340	4
.
.
25	33,6	28,9	79	26,5	6	6	340	3

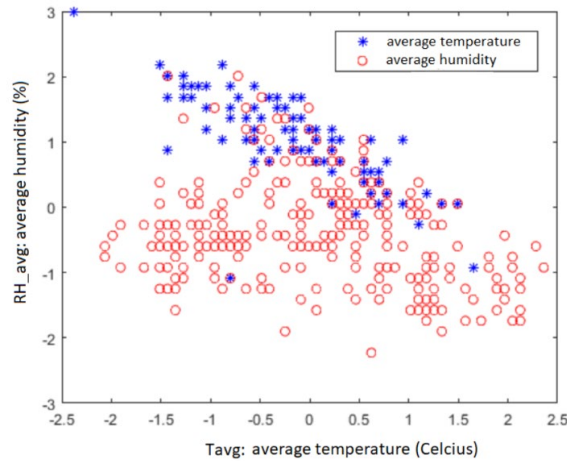


Fig. 1. Non Linear Rainfall dataset

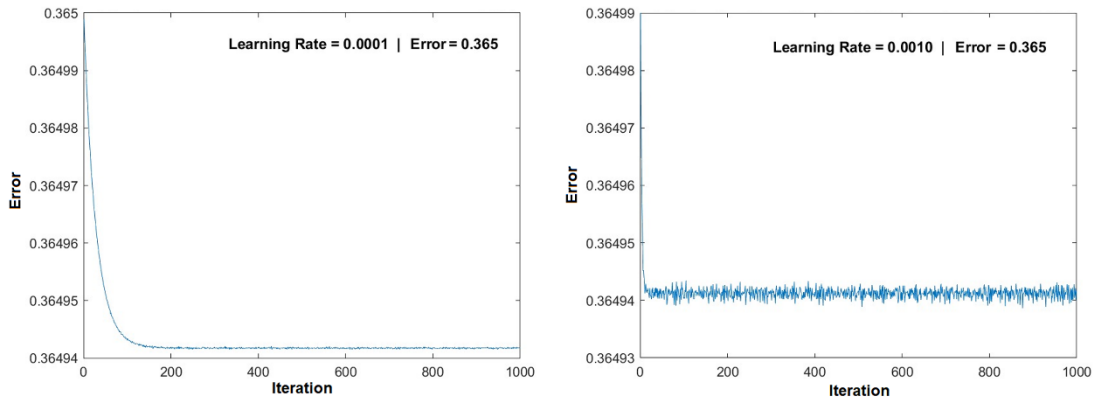


Fig. 2. (a) the value of error rate and learning rate = 0.0001 for first experiment (b) the value of error rate and learning rate = 0.001 for second experiment

4. DISCUSSION

The Data preprocessing first task is we need to remove abnormal values data before attempting to calculate, then normalize the data to eliminate the effects of sample spans, and smooth the training process. The data is subjected to a fivefold cross validation for a more stable data model. Due to the particularity of the precipitation datum, the regional rainfall datum are distributed unevenly in time and space. The days of precipitation are obviously shorter than that of the total sample.

To verify the convergence of the proposed Stochastic Gradient Descent -SVM algorithm, simulation has been done on the Rainfall dataset. The Rainfall dataset is collected from ground-based meteorological data from Maritim

Perak Station (ID WMO: 96937). There have 1825 record data for 1095 training samples and 730 testing samples data. To display the performances of differences of different Stochastic Gradient Descent -SVM, we employed different learning parameters: learning rates and error rate. For convenience, we graphed learning rates and error rate that showed in Fig.1 (a) with value of learning rate = 0.0001 error rate = 0.365. Fig 1. (b) with value of learning rate = 0.001 error rate = 0.364. Generally speaking, the error rate with larger learning rates are lowest error rate than those with smaller learning rates.

Although the datum are pretreated, they still cannot completely remove the datum on the impact of the forecast results, and the degree of volatility is greater when the number of samples is small. We analyze the results of the optimization parameters C and g with the four methods. The SVM-SGD method based on the Stochastic Gradient Descent Optimization algorithm has better experimental results, with a higher classification rate. This indicates that the Stochastic Gradient Descent Optimization algorithm has a strong ability to optimize the parameters of the Support Vector Machine C and g, and the results are more accurate.

5. CONCLUSIONS

The purpose of this study is to investigate the current state of the optimization method based on the gradient descent conducted on the Support Vector Machine classifier. Based on the findings of this study, it is possible to make an overview of the dominant optimization method based on gradient descent. We model the model parameters by using the SGD algorithm to optimize the model parameters. The SVM-SGD algorithm is proven to be an effective method for rainfall forecast decisions. The SVM-SGD algorithm is proven to be an effective method for rainfall forecast decisions. The SVM method is a type of machine learning method with a high level of nonlinear problems. This is a kind of smart learning method with a strong theoretical basis. In addition, there is no limit to the dimensions (number of vectors) of vectors due to the formation of the SVM model, which facilitates the handling of meteorological problems with time, space and various factors.

From the results of the trials conducted it is very important to improve accuracy and increase computational time for learning rainfall datasets that are included in large-scale datasets. For further research it is possible to apply optimization methods based on fractional derivatives in SVM classifiers.

Thus, early warning of weather conditions in an area can be informed to the government before a natural disaster occurs as a form of service using disaster prevention tools. These models provide a good example of the ability of the Vector Support Engine classifier for modeling weather forecasting with high precision and efficiency.

ACKNOWLEDGEMENT

The authors would like to express their gratitude to the editors and anonymous reviewers for their valuable comments and suggestions which improve the quality of this paper. Gratitude's for the supervisor of the doctoral program, the faculty of science and technology, Airlangga University.

REFERENCES

- Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent, In: *Proceedings of COMPSTAT 2010 - 19th International Conference on Computational Statistics, Keynote, Invited and Contributed Papers*. doi: 10.1007/978-3-7908-2604-3-16.
- Chervonenkis, A.Y. (2013). Early history of support vector machines, In: *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*. doi: 10.1007/978-3-642-41136-6_3.
- Esteves, J.T., de Souza Rolim, G., and Ferrauo, A.S. (2019). Rainfall prediction methodology with binary multilayer perceptron neural networks, *Climate Dynamics*. doi: 10.1007/s00382-018-4252-x.
- Hartono, O., Sitompul, O.S., Tulus, and Nababan, E.B. (2018). Optimization Model of K-Means Clustering Using Artificial Neural Networks to Handle Class Imbalance Problem, In: *IOP Conference Series: Materials Science and Engineering*., 288, 1-9. doi: 10.1088/1757-899X/288/1/012075.
- Lin, Y., et al. (2011). Large-scale image classification: Fast feature extraction and SVM training, In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. doi: 10.1109/CVPR.2011.5995477.
- Luo, Z.Q., and Yu, W. (2006). An introduction to convex optimization for communications and signal processing, *IEEE Journal on Selected Areas in Communications*. doi: 10.1109/JSAC.2006.879347.
- Skumar, G., et al. (2019). South Asian perspective on temperature and rainfall extremes: A review, *Atmospheric Research*. doi: 10.1016/j.atmosres.2019.03.021.

- Sakr, C. *et al.* (2017). Minimum precision requirements for the SVM-SGD learning algorithm, In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. doi: 10.1109/ICASSP.2017.7952334.
- Schraudolph, N.N., Yu, J., and Günter, S. (2007). A stochastic quasi-Newton method for online convex optimization, *Journal of Machine Learning Research*, 2, 436-443.
- Sehad, M., Lazri, M., and Ameur, S. (2017). Novel SVM-based technique to improve rainfall estimation over the Mediterranean region (north of Algeria) using the multispectral MSG SEVIRI imagery, *Advances in Space Research*. doi: 10.1016/j.asr.2016.11.042.
- Sopyła, K., and Drozda, P. (2015) Stochastic gradient descent with Barzilai-Borwein update step for SVM, *Information Sciences*. doi: 10.1016/j.ins.2015.03.073.
- Wijnhoven, R.G.J., and De With, P.H.N. (2010). Fast training of object detection using stochastic gradient descent, In: *Proceedings - International Conference on Pattern Recognition*. doi: 10.1109/ICPR.2010.112.
- Wu, W. *et al.* (2011). Convergence analysis of online gradient method for BP neural networks, *Neural Networks*. doi: 10.1016/j.neunet.2010.09.007.
- Xu, D., Zhang, H., and Mandic, D.P. (2015). Convergence analysis of an augmented algorithm for fully complex-valued neural networks, *Neural Networks*. doi: 10.1016/j.neunet.2015.05.003.
- Young, C.C., Liu, W.C., and Wu, M.C. (2017). A physically based and machine learning hybrid approach for accurate rainfall-runoff modeling during extreme typhoon events, *Applied Soft Computing Journal*. doi: 10.1016/j.asoc.2016.12.052.
- Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms, in *Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004*. doi: 10.1145/1015330.1015332.