

SVM-based Sketch Recognition: Which Hyperparameter Interval to Try?

K. T. Yesilbek^{1†}, C. Sen^{1†}, S. Cakmak^{1†}, T. M. Sezgin¹

¹Koc University, Turkey

Abstract

Hyperparameters are among the most crucial factors that affect the performance of machine learning algorithms. In general, there is no direct method for determining a set of satisfactory parameters, so hyperparameter search needs to be conducted each time a model is to be trained. In this work, we analyze how similar hyperparameters perform across various datasets from the sketch recognition domain. Results show that hyperparameter search space can be reduced to a subspace despite differences in characteristics of datasets.

Categories and Subject Descriptors (according to ACM CCS): I.5.0 [Pattern Recognition]: General—H.5.2 [Information Interfaces and Presentation]: User Interfaces—Input devices and strategies

1. Introduction

The selection of proper hyperparameters is one of the most significant factors that affect the accuracy of machine learning algorithms. The most widespread method used to determine the best hyperparameters is grid search [Sta03], [Nar07], [LXKJ13]. To find the optimum hyperparameters, this method requires an exhaustive search for all combinations of hyperparameters prior to the training phase. This exhaustive search requires time, machine power and labor due to its high computational cost. As there is no general set of hyperparameters that works across different databases and domains, grid search process is repeated for every individual study.

In this paper, we propose a confined space of hyperparameters to be used in the domain of sketch recognition. We examine five sketch datasets with different characteristics in our experiments. Considering their effectiveness in sketch recognition, we apply Support Vector Machines as the machine learning algorithm in this study [UYA12]. In order to identify the optimum hyperparameters shared among different datasets, we performed grid search with various hyperparameters in a wide range. Experimental results show that, hyperparameter search space can be confined to a specific region for sketch recognition and lower the computational cost of training recognizers for new datasets.

2. Related Work

The most popular method in the literature employed for the hyperparameter search problem involves defining a range for hyperparameters and performing grid search on this range [Sta03], [Nar07], [LXKJ13]. In this method, hyperparameters are varied using a predefined step size in a wide range and accuracies are recorded for each combination of parameters. Gradient based methods are other popular methods used for hyperparameter search [CVBM02], [GI05], [Kee02]. However, gradient based methods come with certain drawbacks. The most significant disadvantage of these methods is the necessity of the function being differentiable, making the method impractical for general usage. Therefore, we prefer grid search for optimizing hyperparameters.

3. Method

We conduct an experiment to find a common sub-range of hyperparameters on different datasets that produces optimum results. On these datasets, we apply the grid search method for determining the free hyperparameters of the Support Vector Machine (SVM) algorithm and observe a sub-range that gives the highest accuracies.

3.1. Datasets

For the purpose of finding a common sub-range of hyperparameters, we conducted experiments on five frequently used datasets in the sketch recognition domain, namely:

[†] Authors have equally contributed to this work.


























COAD	NicIcon	TrafficSign	ShortStraw	Eitz
				
				
				
				
				

Table 1: Sample sketches from datasets

1. COAD dataset [Man97], including symbols from military courses of action diagrams
2. NicIcon database [NWV08], that expresses the context of an emergency management application
3. Traffic Sign database [Oze13], which consists of traffic sign boards
4. ShortStraw database [XJ09], that is used in corner finding algorithms for sketches
5. Eitz database [EHA12], which consists of numerous objects from the daily life

Table 1 presents the identifying instances from 5 classes which are selected randomly just for visualization purposes. We do not have any constraint on the selected instances, so our argument about the subspace restriction would hold even if different instances had been chosen.

3.1.1. COAD Dataset

A complete list of objects in this dataset can be found in the US Army Field Manual 101-5-1 [Man97]. There are hundreds of symbols in this dataset, and we focus on a subset of 20 for practical reasons. Eight different users sketched the symbols from each of the 20 classes. In total 620 examples of different symbols were collected and the dataset contains approximately 27 symbols per class.

3.1.2. NicIcon Dataset

The NicIcon database includes multi-stroke symbols that are used in the context of an emergency management application (e.g., symbols for fire brigade, gas, roadblock, injury). Dataset includes 23641 symbols from 14 different classes in total. The symbols consist of an average of 5.2 strokes, and the average number of strokes for the individual categories ranges from 3.1 to 7.5. Dataset contains approximately 1860 symbols per class.

3.1.3. TrafficSign Dataset

TrafficSign dataset consists of 10320 symbols from 8 different traffic sign classes. Full dataset was constructed by 43 different users, and each user has 30 sketches per class. In

	Number of objects	Number of participants	Average number of strokes	Density of sketch points
COAD	20	8	3.64	0.0271
NicIcon	14	35	2.83	0.0052
TrafficSign	8	43	3.84	0.0112
ShortStraw	15	6	1.00	0.0067
Eitz	250	1350	17.55	0.0168

Table 2: Dataset characteristics

order to keep our experiments tractable, we used a smaller subset of 800 symbols, i.e. 100 randomly selected symbols for each class. Dataset contains approximately 1290 symbols per class.

3.1.4. ShortStraw Dataset

ShortStraw Dataset consists of 15 classes which are collected from 15 different users. Each user was asked to draw each shape four times. 1260 strokes were collected, but 14 were removed because they were very poorly written. Dataset contains 84 symbols per class.

3.1.5. Eitz Dataset

Eitz dataset consists of 20000 symbols from 250 different classes which are collected from 1350 unique users. Amazon Mechanical Turk (AMT) is used in order to collect the sketches [EHA12]. Dataset contains 80 symbols per class and a total of 351,060 strokes with each sketch containing a median number of 13 strokes.

3.2. Dataset Characteristics

The five datasets used in our experiments are quite different from each other. The characteristic features that differentiate them are number of objects, number of participants, number of strokes and the density of sketch points. The density of sketch points is determined by dividing the number of sketch points by the area of sketch. Related information about the characteristic features of datasets is shown in Table 2.

3.3. Features

In this study, we used IDM (Image Deformation Model) features considering the fact that it is one of the most successful feature extraction method in sketch recognition domain [OD09]. In this method, sketches are converted to low density feature images by using the directions and end points of the strokes. The free parameters for the IDM method are k (kernel size), r (resampling parameter) and σ (smoothing factor). We set $k = 25$, $r = 50$ and $\sigma = 10$ in order to get the highest prediction accuracies [TAS10]. The resulting IDM features have 720 entries for each sketch.

3.4. Support Vector Machines

In our experiments, we used C-SVM (a variant of SVM) with RBF kernel together with free parameters. We perform 5-fold cross validation to objectively assess the classifiers performance in our analyses. C-SVM with RBF kernel needs two hyperparameters to be specified. The hyperparameter C , controls the weight of the free parameters and the other hyperparameter γ defines the shape of the RBF kernel. We used LibSVM library for the implementation of the SVM algorithm [CL11].

3.5. Grid Search

We train a separate model for every combination of SVM free hyperparameters and record the cross validation accuracies. For each dataset we visually demonstrate the cross validation accuracies. Both for the training and test data, we use the best hyperparameter pairs obtained through grid search. Training and test accuracies are beyond the scope of this paper. Our purpose is to define a restricted hyperparameter range. Hence, we report only the cross validation accuracies.

3.6. Hyperparameter Range

We performed grid search in the range between 2^{-10} and 2^{20} for the C parameter. Step size is defined as 0.5 in exponential range of values. For the γ parameter, grid search was performed in the range between 2^{-20} and 2^{10} in such a way that the exponential values increase by 0.5. Thus, we scan 3721 hyperparameter combinations in total, 61 different hyperparameters for C and 61 different parameters for gamma specifically.

4. Experiment

In the grid search method, we need to retrain the SVM model for each pair of hyperparameters and make predictions based on this model. Therefore, it suffers from computational complexity. Computational time is dependent on the number of free hyperparameters, step size, and dataset size. When using the grid search method, researchers face the problem of computational time being intractable. In this study, we parallelized grid search in order to make it work on a multi-core processor with the purpose of decreasing the computational time. We divided this process into different executable files such that the accuracies of hyperparameter pairs are calculated by these files. We split the computations into 961 different executable files and ran our experiments on a high performance cluster with 150 distributed cores.

5. Results

We report the experiment results separately for each dataset in Figures 1 through 5. As the results for five datasets are combined, we find a common interval for SVM hyperparameters, inspite of the different characteristics of the datasets.

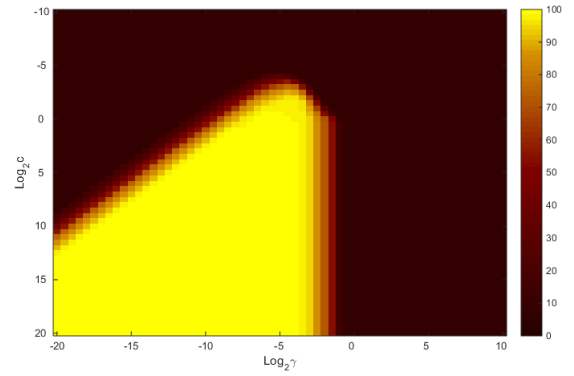


Figure 1: Cross validation accuracies for the COAD dataset

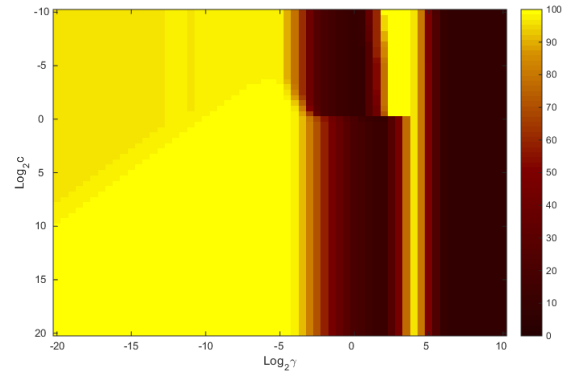


Figure 2: Cross validation accuracies for the NicIcon dataset

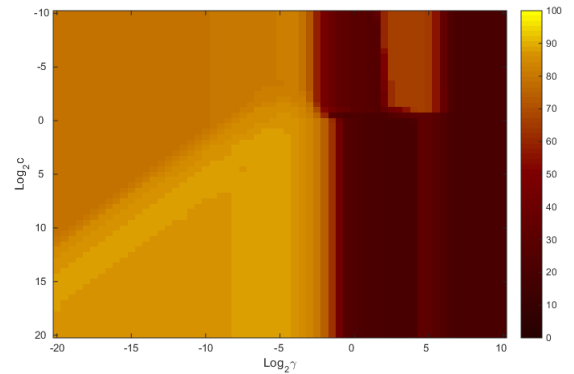


Figure 3: Cross validation accuracies for the TrafficSign dataset

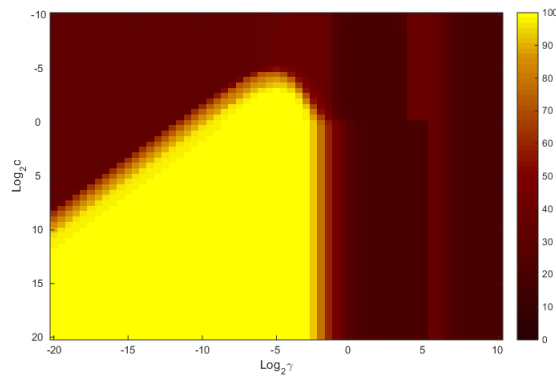


Figure 4: Cross validation accuracies for the ShortStraw dataset

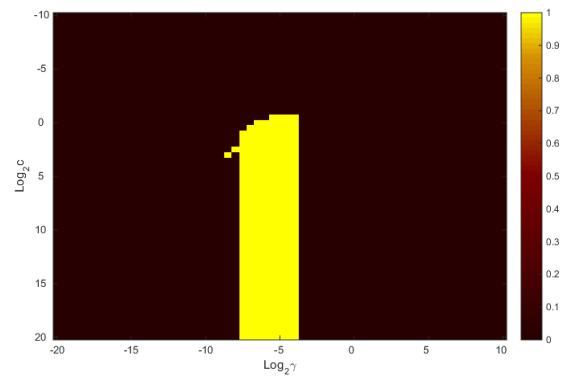


Figure 7: Intersection of the highest accuracy regions

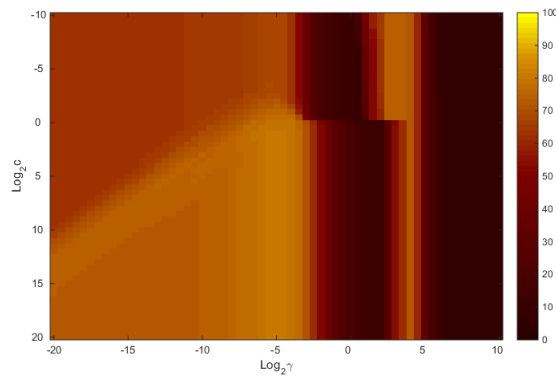


Figure 5: Cross validation accuracies for the Eitz datasets

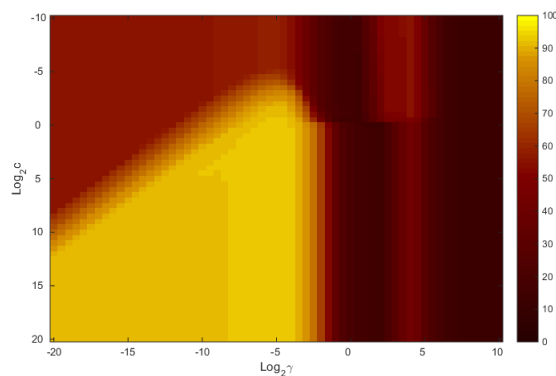


Figure 6: Cross validation accuracies for the COAD dataset

We acquire highest accuracies when we choose C hyperparameter from the interval $[2^{-2}, 2^{20}]$ and γ hyperparameter from the interval $[2^{-7.5}, 2^{-3}]$. We present average accuracies for all datasets in Figure 6.

Figure 7 shows how we reduce the hyperparameter search space. For all datasets, the regions where accuracies are within 95% of the maximum cross validation accuracies are intersected. Grid search is performed for 3721 hyperparameter pairs and 331 of these pairs have the highest accuracies in the intersected region. As a result, we reduce the hyperparameter search space by 91.1%. Searching only 8.9% of the overall search space, amount of time required to find the optimum hyperparameters decreases from tens of hours to minutes.

To evaluate our restricted hyperparameter selection strategy, we assess the results on an unseen dataset by excluding one dataset each time in a leave-one-out fashion. From five datasets, we exclude one dataset and determine the shared hyperparameter interval by intersecting the high-accuracy regions of remaining four datasets. In Figures 8 and 9, we present the mean accuracy and maximum accuracy results on unseen datasets both for entire hyperparameter space and subspace. Maximum accuracies are the upper limit that we can reach and we see that for an unseen dataset maximum accuracy is acquired when the subspace of hyperparameters are used. We can also see that, mean accuracies are much higher in the restricted hyperparameter space than they are in overall hyperparameter space.

Leave-one-out experiment results show that hyperparameter sets chosen from our restricted interval will include the parameters giving the highest accuracy for the dataset that is left out. Hence, using our restricted hyperparameter space leads higher accuracy on any future dataset.

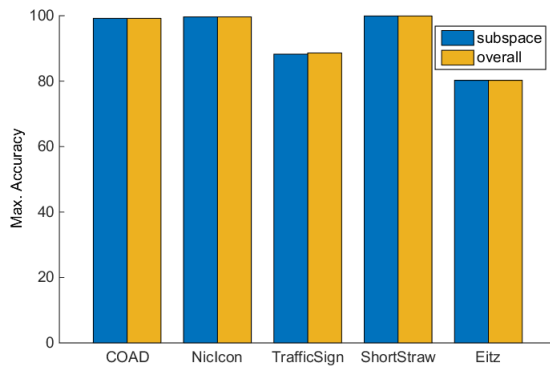


Figure 8: Maximum accuracy results on unseen datasets (shown on the x axis) for the whole hyperparameters and for the subspace of hyperparameters

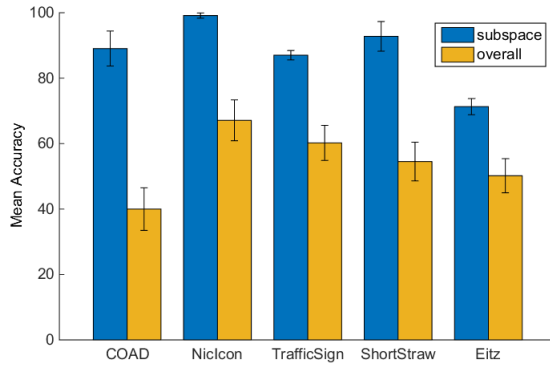


Figure 9: Mean accuracy results on unseen datasets (shown on the x axis) for the whole hyperparameters and for the subspace of hyperparameters

6. Discussion

Accuracy of machine learning algorithms is directly related to the selected hyperparameter values. Grid search is among the most commonly used methods for this purpose. In spite of being simple and reliable, grid search holds a high computational complexity that makes the usage of this method troublesome. In this study, we investigated how C-SVM algorithm with RBF kernel performs on different sketch datasets by scanning a wide range of hyperparameters. Our analysis shows that there is a common interval that delivers high accuracies for all datasets. According to our observations, hyperparameter search for C-SVM algorithm with RBF kernel is reduced so that C values vary between 2^{-2} and 2^{20} , while γ values vary between $2^{-7.5}$ and 2^{-3} .

Furthermore, we argue that techniques that work well across a sufficiently large set of sketch symbols should, in-

deed, handle new sketch symbol sets very well. We support our argument by the leave-one-out dataset study. In conclusion, the proposed hyperparameter intervals for cost and gamma are generalizable for new sketch datasets.

7. Acknowledgement

The authors gratefully acknowledge the support and funding of SAN-TEZ under grant number 01198.stz.2012-1 and TUBITAK under grant number 113E059.

References

- [CL11] CHANG C.-C., LIN C.-J.: Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2, 3 (2011), 27. [3](#)
- [CVBM02] CHAPPELLE O., VAPNIK V., BOUSQUET O., MUKHERJEE S.: Choosing multiple parameters for support vector machines. *Machine learning* 46, 1-3 (2002), 131–159. [1](#)
- [EHA12] EITZ M., HAYS J., ALEXA M.: How do humans sketch objects? *ACM Trans. Graph.* 31, 4 (2012), 44. [2](#)
- [GI05] GLASMACHERS T., IGEL C.: Gradient-based adaptation of general gaussian kernels. *Neural Computation* 17, 10 (2005), 2099–2105. [1](#)
- [Kee02] KEERTHI S. S.: Efficient tuning of svm hyperparameters using radius/margin bound and iterative algorithms. *IEEE Transactions on Neural Networks* 13, 5 (2002), 1225–1229. [1](#)
- [LXKJ13] LIN L., XIAOLONG Z., KAI Z., JUN L.: Bilinear grid search strategy based support vector machines learning method. [1](#)
- [Man97] MANUAL F.: 101-5-1, operational terms and graphics. Washington, DC, Department of the Army 30, 3 (1997). [2](#)
- [Nar07] NARZISI G.: An experimental multi-objective study of the svm model selection problem. *Courant Inst. Math. Sci* (2007). [1](#)
- [NWV08] NIELS R., WILLEMS D., VUURPIJL L.: The nicicon database of handwritten icons for crisis management. *Nijmegen Institute for Cognition and Information Radboud University Nijmegen, Nijmegen, The Netherlands* (2008). [2](#)
- [OD09] OUYANG T. Y., DAVIS R.: A visual approach to sketched symbol recognition. [2](#)
- [Oze13] OZEN B.: *A Simple Resource-Aware Approach to Sketch Recognizers via Style Identification*. Master's thesis, Koc University, Istanbul, Turkey, 2013. [2](#)
- [Sta03] STAELIN C.: Parameter selection for support vector machines. *Hewlett-Packard Company, Tech. Rep. HPL-2002-354R1* (2003). [1](#)
- [TAS10] TUMEN R. S., ACER M. E., SEZGIN T. M.: Feature extraction and classifier combination for image-based sketch recognition. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium* (2010), Eurographics Association, pp. 63–70. [2](#)
- [UYA12] ULAŞ A., YILDIZ O. T., ALPAYDIN E.: Cost-conscious comparison of supervised learning algorithms over multiple data sets. *Pattern Recognition* 45, 4 (2012), 1772–1781. [1](#)
- [XJ09] XIONG Y., JR J. J. L.: *Revisiting ShortStraw: improving corner finding in sketch-based interfaces*. Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling, 2009. [2](#)