Improving Classification Performance of Support Vector Machines via Guided Custom Kernel Search

Kumar Ayush*
Media and Data Science Research, Adobe Inc.
kayush@adobe.com

Abhishek Sinha*
Media and Data Science Research, Adobe Inc.
abhsinha@adobe.com

ABSTRACT

Support Vector Machines (SVMs) deliver state-of-the-art performance in real-world applications and are established as one of the standard tools for machine learning and data mining. A key problem of these methods is how to choose an optimal kernel function. The real-world applications have also emphasized the need to adapt the kernel to the characteristics of heterogeneous data in order to boost the classification accuracy. Therefore, our goal is to automatically search a task specific kernel function. We use reinforcement learning based search mechanisms to discover custom kernel functions and verify the effectiveness of our approach by conducting an empirical evaluation with the discovered kernel function on MNIST classification. Our experiments show that the discovered kernel function shows significantly better classification performance than well-known classic kernels. Our solution will be very effective for resource constrained systems with low memory footprint which rely on traditional machine learning algorithms like SVMs for classification tasks.

KEYWORDS

Support Vector Machines, Kernel Function, Reinforcement Learning, Classification

ACM Reference Format:
Kumar Ayush* and Abhishek Sinha*. 2019. Improving Classification Performance of Support Vector Machines via Guided Custom Kernel Search. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3321923

1 INTRODUCTION

Support vector machines (SVMs) have shown great capability in producing high quality solutions for many types of real-world classification problems. SVMs classify the data by selecting hyperplanes that provide the maximum margin between classes. The expressive power of SVMs can be extended by mapping input data into high dimensional spaces. In the high dimensional spaces, the computational complexity of SVMs does not increase much, because all of the computation is done through the kernels - "the kernel trick". The maximal margin principle and the kernel trick make SVMs useful and efficient computational tools in machine learning. However, it was shown that any classical kernel achieves good enough performances for some classification problems [2, 3]. In real world problems, engineering an appropriate kernel becomes the major part of the modelling process. In this work, we propose to use automated search techniques to discover custom kernel functions. Using a reinforcement learning-based search approach, we find a custom kernel function (by using the data of a particular problem) that shows promising performance.

Deep learning models have performed remarkably well in several domains such as computer vision [5], natural language processing [4] and speech recognition [6]. These models are able to achieve high accuracy in various tasks and hence their recent popularity. Deep reinforcement learning based search mechanisms have been recently used for discovering neural optimization methods [1], neural activation functions [7] and neural architecture designs [8]. In this work, we build on the shoulder of these works and adapt it to our problem, allowing for the automatic discovery of a task specific custom SVM kernel function using the data for a particular problem. The numerical experiments show that such an approach is able to discover a befitting kernel function that are more efficient and achieves better classification performance and generalization on the considered dataset.

2 APPROACH

In this section, we explain our approach. As discussed, we built upon the existing RL based search mechanisms [1, 7, 8] with suitable modifications.

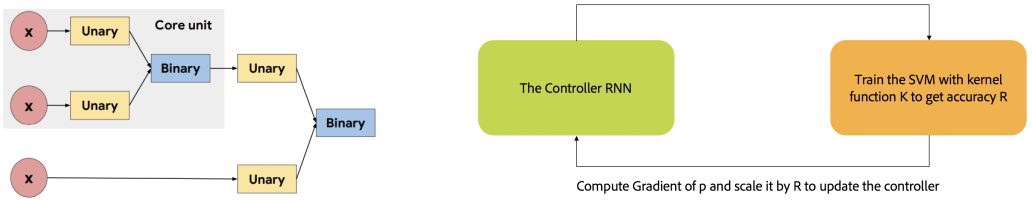
As shown in Figure 1 (a), the kernel function is constructed by repeatedly composing the the "core unit". A core unit first selects two operands ($op1$ and $op2$), then two unary functions ($u1$ and $u2$) to apply on the operands and finally a binary function b that combines the outputs of the two unary functions. The resulting $b(u1(op1), u2(op2))$ then becomes an operand that can be selected in the next group of predictions. Every prediction is carried out by a softmax classifier and then fed into the next time step as input.

Given the search space, the goal of the search algorithm is to find effective choices for the unary and binary functions. We use an RNN controller [8]. At each timestep, the controller predicts a single component of the kernel function. The prediction is fed back to the controller in the next timestep, and this process is repeated until every component of the kernel function is predicted. The predicted string is then used to construct the kernel function. Once a candidate kernel function has been generated by the search algorithm, a SVM model with the candidate kernel function is trained on some task, such as MNIST classification. After training,

*Equal Contribution.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3321923



(a) Example Kernel Function Structure (from [7])

(b) Overview of Kernel Function Search

Figure 1

the validation accuracy is recorded and used to update the search algorithm.

The RNN controller is trained with reinforcement learning to maximize the validation accuracy, where the validation accuracy serves as the reward. This training pushes the controller to generate kernel functions that have high validation accuracies.

3 SEARCH FINDINGS

We conduct our searches over MNIST classification dataset. The operands, unary functions and binary functions that are accessible to our controller are the following:

- Operands using the two vectors x and y: $|x-y|$, $(x+y)$, $(x-y)^2$, $||x-y||_1$, $||x-y||_2$, $x \cdot y$, $x * y$, γ[1]
- Unary functions: x, $-x$, x^2, $|x|$, x^3, $\sqrt{(|x|)}$, e^x, $\sin x$, $\cos x$, $\sinh x$, $\cosh x$, $\tanh x$, $||x||_1$, $||x||_2$, $\max(x,0)$, $\min(x,0)$, $\log_e(1+e^x)$, $\sigma(x)$
- Binary functions: x_1+x_2, x_1-x_2, $x_1 \cdot x_2$, $x_1 * x_2$, $\max(x_1,x_2)$, $\min(x_1,x_2)$, $e^{-|x_1-x_2|}$, x_1

The operands have been chosen in a way so as to keep the discovered kernel function symmetric. The RNN controller emits a kernel function for SVM. The SVM with the emitted kernel function is trained over 1000 MNIST training samples and the accuracy over a separate 500 validation samples is used as a reward signal for the RNN controller. The RNN controller is trained via the vanilla policy gradient algorithm. The final discovered kernel function is:

$$k(x,y) = ||\min(\sin(x*y),\ \sin(x \cdot y/\gamma))|| \quad (1)$$

The results in Table 1 show the accuracy over 10000 MNIST test samples when the SVM is trained with different kernel functions (classic kernels and discovered kernel function) over 1000 training samples.

Even though the RNN controller has been trained to learn a kernel function that best fits the 1000 training samples, the discovered kernel function works better than the classic kernel functions even when it is used to fit 2000 training samples. Table 2 shows the test accuracy results when different kernel functions are used to train SVM over 2000 samples.

4 CONCLUSION

In this paper, we use RL based search mechanisms to find task specific custom kernel functions. We show that the discovered kernel function provides better classification performance and generalization capabilities than well known classic kernels. Our work advocates future research towards such kernel discovery. We plan to perform extensive evaluations with multiple datasets.

[1] γ is a constant, that is equal to the number of features in the data

Table 1: Comparison with classic kernel functions on MNIST testset while being trained on 1000 training examples

Kernel Function	Validation Accuracy	Test Accuracy
Linear	89.0	87.96
RBF	86.4	82.76
Sigmoid	81.0	74.42
Discovered Kernel	90.2	91.01

Table 2: Comparison with classic kernel functions on MNIST testset while being trained on 2000 training examples.

Kernel Function	Test Accuracy
Linear	89.87
RBF	87.85
Sigmoid	84.94
Discovered Kernel	93.12

REFERENCES

[1] Irwan Bello, Barret Zoph, Vijay Vasudevan, and Quoc V Le. 2017. Neural optimizer search with reinforcement learning. *arXiv preprint arXiv:1709.07417* (2017).
[2] Bao Rong Chang and Hsiu-Fen Tsai. 2007. Composite of adaptive support vector regression and nonlinear conditional heteroscedasticity tuned by quantum minimization for forecasts. *Applied Intelligence* 27, 3 (2007), 277–289.
[3] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. 2002. Choosing multiple parameters for support vector machines. *Machine learning* 46, 1-3 (2002), 131–159.
[4] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. ACM, 160–167.
[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.
[6] Abdel-rahman Mohamed, George E Dahl, and Geoffrey Hinton. 2012. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing* 20, 1 (2012), 14–22.
[7] Prajit Ramachandran, Barret Zoph, and Quoc V Le. 2018. Searching for activation functions. (2018).
[8] Barret Zoph and Quoc V Le. 2016. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578* (2016).