Proceedings of Machine Learning Research 142: 35-38, 2021 2nd Workshop on Diversity in Artificial Intelligence (AIDBEI)

Working Set Selection to Accelerate SVR Training*

Pablo Rivas PABLO_RIVAS@BAYLOR.EDU
Department of Computer Science
Baylor University
Waco, TX 76798-97141, USA

Editor: Deepti Lamba and William H. Hsu

Abstract

With the increasing demand for robust and resilient machine learning models, support vector machines (SVMs) are regaining attention. One of the significant problems in SVMs is finding the support vectors as soon as possible during the optimization process. This paper describes a methodology to accelerate the training by making certain assumptions on the data and find the support vectors near the convex hull of every class group. Results suggest that the methodology can provide an advantage over traditional training for larger datasets with specific statistical properties. We focus on the particular case of support vector machines for regression.

Keywords: Support Vector Machines, Regression, Learning Theory

1. Background

Support vector machines for regression (SVRs) have wide range of applications (Zhong et al., 2019; Santamaría-Bonfil et al., 2016; Trzciński and Rokita, 2017). It is well known that traditional SVR formulations do not make assumptions about the probability distribution of the data (Vapnik et al., 1997; Joachims, 1998; Cherkassky and Ma, 2004). Nonetheless, each class ω_j, should have a conditional class distribution $p(\mathbf{x}|\omega_j)$, where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^M$ is defined as an M-dimensional random variable which could be be estimated if enough data points were available (Bennett and Bredensteiner, 2000; Joachims, 1998). Estimating a multidimensional probability density function (PDF) is difficult but we could make some basic assumptions. First, we could assume that the data has a uni-modal distribution, which implies that data-samples would cluster around the class mean, and that the further away a point is from its mean, the lower its probability is and, thus, it is expected to be localized near the convex hull of the data sample we are analyzing (Gu et al., 2018; Wang et al., 2013; Liu et al., 2009). A more strict assumption would be to consider that the $p(\mathbf{x}|\omega_j)$ are multivariate Gaussian distributed. Under this assumption each $p(\mathbf{x}|\omega_j)$ could be modeled using only the sample mean $\boldsymbol{\mu}_{\mathbf{x}|\omega_j}$ and a covariance matrix $\boldsymbol{\Sigma}_{\mathbf{x}|\omega_j}$, that is $p(\mathbf{x}|\omega_j) \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}|\omega_j}, \boldsymbol{\Sigma}_{\mathbf{x}|\omega_j})$. It is also well known that we can use the squared Mahalanobis distance (MD), $D(\mathbf{x}_i) = (\mathbf{x}_i - \boldsymbol{\mu}_{\mathbf{x}|\omega_j})^T \boldsymbol{\Sigma}^{-1}_{\mathbf{x}|\omega_j} (\mathbf{x}_i - \boldsymbol{\mu}_{\mathbf{x}|\omega_j})$, as a distance measure from a data point to its mean.

Based on these strict assumptions we propose a method for finding the SV candidates by computing the distances $D(\mathbf{x}_i)$ for all $i = \{1, 2, \ldots, N\}$. Once all training vectors are

0. A full version of this short paper was presented and published in (Rivas, 2020)

©2021 Pablo Rivas.

Algorithm 1 Mahalanobis-Based Working-Set Selection for LP-SVR Training Speed-Up

Require: Training set $\mathcal{T}_\phi = \{\mathbf{x}_i, d_i\}_{i=1}^N$.
Require: Desired num. samples per class v.
for $j = 1$ to $|D|$ **do**
  Estimate parameters $(\boldsymbol{\mu}_{\mathbf{x}|\omega_j}, \boldsymbol{\Sigma}_{\mathbf{x}|\omega_j})$
  for $i = 1$ to N **do**
    Get Mahalanobis distance $D(\mathbf{x}_i)_j$
  end for
  Get $\mathcal{Z}_j$ corresponding to sorted D_j
  for $i = 1$ to v **do**
    $\mathcal{Z}_{j,i} = \mathcal{Z}(i)_j$ ▷ $B_{\text{ini}} = k \equiv v \times |\mathcal{D}|$
  end for
end for
Ensure: Init. working set indices $\mathcal{B} \leftarrow \mathcal{Z}_{j,i}$
Ensure: Fixed set $\mathcal{M} \leftarrow \{1, \ldots, N\} \notin \mathcal{Z}_{j,i}$

sorted by their MD to their respective mean and saved into the sets $\mathcal{Z}_j$ for the j-th class, then we can form an initial working set $\mathcal{B}$ of size B_{ini} using the procedure described in Algorithm 1. We traverse elements of $\mathcal{Z}_{j,i}$ and add them into to $\mathcal{B}$ until B_{ini} elements have been inserted. In this manner, the SVR could be trained faster if the first working-set $\mathcal{B}$ contains those k samples, thereby, speeding up the training process. A similar approach to ours is given by Zhou, *et al.* (Zhou et al., 2010) in 2010; however, the authors' approach is based on class and subclass convex hulls, which makes it computationally expensive.

2. Within-Class Mahalanobis distance and Class-Convex Hull

To explain the ideas behind the procedure shown in Algorithm 1, consider the following definitions: Let $\mathcal{D} = \{\omega_1, \omega_2, \ldots, \omega_j\}$ be the set of classes where j is the total number of classes. Let $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_j\}$ denote a set of indices, where $\mathcal{C}_j$ contains the indices of all those samples associated with the j-th class, $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ for all $i \neq j$, and $\mathcal{C} \equiv \{1, 2, \ldots, N\}$. We will be considering the case of all the samples $\mathbf{x}_i$ belonging to the j-th class, that is, all $i \in \mathcal{C}_j$. The same principles will apply to all classes.

One of the first steps is to estimate the parameters $(\boldsymbol{\mu}_{\mathbf{x}|\omega_j}, \boldsymbol{\Sigma}_{\mathbf{x}|\omega_j})$, *i.e.*, from observed events. Then the within-class MD from the i-th feature vector $\mathbf{x}_i$ to the center of the j-th class $\boldsymbol{\mu}_{\mathbf{x}|\omega_j}$ is defined as $D(\mathbf{x}_i)$. Next, we define $\mathcal{Z}_j$ as the set of indices corresponding to the ordered Mahalanobis distance samples of the $j-$th class. The indices in $\mathcal{Z}_j$ correspond to ordered values in descending form, as shown in Figure 1. In our research, we argument that the MD $D(\mathbf{x}_i)$ is related to the SVs and the class convex hull (CCH), which is defined as follows: $\Theta(\omega_j) = \left\{ \sum_{i \in \mathcal{C}_j} \beta_i \mathbf{x}_i \ : \ i \in \mathcal{C}_j,\ \beta_i \in \mathbb{R},\ \beta_i \geq 0,\ \sum_{i \in \mathcal{C}_j} \beta_i = 1 \right\}$, where a number of $|\mathcal{C}_j|$ points in the form of $\sum_{i \in \mathcal{C}_j} \beta_i \mathbf{x}_i$ are the boundaries of the j-th class sample cloud. Then we can define the sets of indices corresponding to the convex hull of the j-th as $\mathcal{S} = \Theta(\omega_j)$. The algorithm that obtains the convex hull has complexity of $\mathcal{O}(N^{\frac{M}{2}})$, where M is the dimensionality of the feature vector. The complexity of the method proposed here has a complexity of $\mathcal{O}(L)$, where $L = \max\left[N \log N, \binom{M}{2}\right]$. This demonstrates that our model has lower complexity than those based on convex hulls. Now, we define a relationship between $\mathcal{Z}$, $\mathcal{S}$, and SV in Proposition 1.

Proposition 1 (SVs and Within-Class Distances) *Assume classes in $\mathcal{D}$ are linearly separable. Let $\mathcal{Z}_v = \{\mathcal{Z}(1), \mathcal{Z}(2), \ldots, \mathcal{Z}(v)\}$ denote the v maximum Mahalanobis distance*

indices. Let $\mathcal{Z}_{j,i} = \{\mathcal{Z}_{1,v}, \mathcal{Z}_{2,v}, \ldots, \mathcal{Z}_{j,v}\}$ *be the set of* v *maximum Mahalanobis distance indices of all classes. Then*

1. *the maximum Mahalanobis distance samples indices contain the convex hull:* $\mathcal{Z}_v \in \mathcal{S}$,

2. *the maximum Mahalanobis distance samples indices contain the SVs:* $\mathcal{Z}_{j,i} \in SV$,

where v *is an integer stating how many samples per class should be considered.*

Proposition 1 states that the first v ranked MD indices $\mathcal{Z}_v$ contain the class convex hull indices $\mathcal{S}$ and, thus, contain the support vector indices. The integer v is bounded, $|\mathcal{D}| \leq v \leq |\mathcal{S}|$. Therefore, if the initial working-set is fixed to the indices in $\mathcal{Z}_{j,i}$, the training process will converge faster. This is mainly because if the support vectors are found at the very first iterations, the problem will be solved faster. Since $\mathcal{Z}_{j,i}$ is more likely to contain support vector indices, one can conclude that the training will be faster. We have found that a good value for v is the quotient between the initial working set size B_{ini} and the total number of classes: $v = \left\lceil \frac{B_{\text{ini}}}{|\mathcal{D}|} \right\rceil$. This choice of v was found empirically using benchmark datasets. This value v is used as input in Algorithm 1. Using such benchmark datasets, the speed-up in terms of training time is an average of 30.6% ($\pm$7.7).

Acknowledgments

Thanks to the reviewers and organizers of the workshop for their valuable feedback.

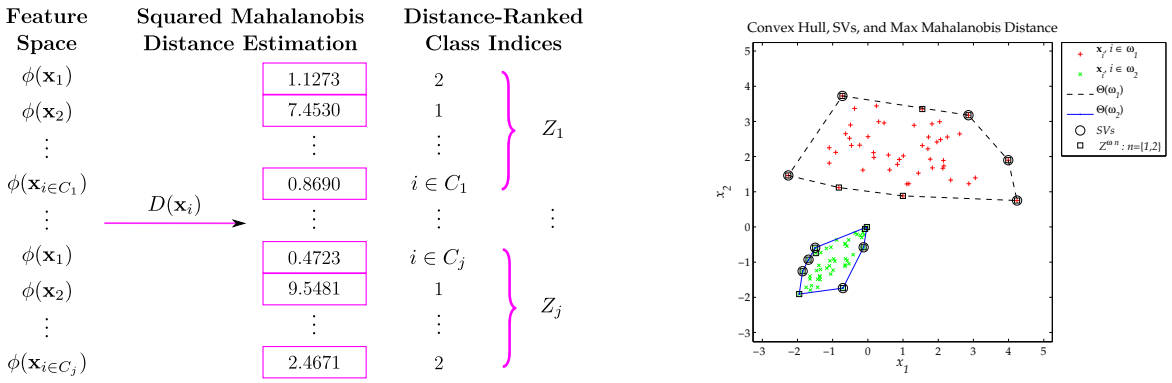

Figure 1: *Left:* Mahalanobis distance-ranking of class indices using feature vectors in either the input space or the kernel-induced feature space. *Right:* Relationship between convex hull and maximum Mahalanobis distance for a two class problem. Here it is shown a separable two class problem, class convex hull, support vectors, and k maximum Mahalanobis distance samples. Note that both SVs and Convex Hull match the k maximum Mahalanobis distance samples.

References

Kristin P Bennett and Erin J Bredensteiner. Duality and geometry in svm classifiers. In *ICML*, volume 2000, pages 57–64, 2000.

Vladimir Cherkassky and Yunqian Ma. Practical selection of svm parameters and noise estimation for svm regression. *Neural networks*, 17(1):113–126, 2004.

Xiaoqing Gu, Fu-lai Chung, and Shitong Wang. Fast convex-hull vector machine for training on large-scale ncrna data classification tasks. *Knowledge-Based Systems*, 151:149–164, 2018.

Thorsten Joachims. Making large-scale svm learning practical. Technical report, Technical Report, 1998.

Zhenbing Liu, JG Liu, Chao Pan, and Guoyou Wang. A novel geometric approach to binary classification based on scaled convex hulls. *IEEE transactions on neural networks*, 20(7): 1215–1220, 2009.

Pablo Rivas. Accelerating the training of an lp-svr over large datasets. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 123–136. Springer, 2020.

Guillermo Santamaría-Bonfil, A Reyes-Ballesteros, and C Gershenson. Wind speed forecasting for wind farms: A method based on support vector regression. *Renewable Energy*, 85:790–809, 2016.

Tomasz Trzciński and Przemysław Rokita. Predicting popularity of online videos using support vector regression. *IEEE Transactions on Multimedia*, 19(11):2561–2570, 2017.

V. Vapnik, S. Golowich, and A. Smola. Support vector method for function approximation, regression estimation, and signal processing. *Advances in neural information processing systems*, 9:281–287, 1997.

Di Wang, Hong Qiao, Bo Zhang, and Min Wang. Online support vector machine based on convex hull vertices selection. *IEEE Transactions on Neural Networks and Learning Systems*, 24(4):593–609, 2013.

Hai Zhong, Jiajun Wang, Hongjie Jia, Yunfei Mu, and Shilei Lv. Vector field-based svr for building energy consumption prediction. *Applied Energy*, 242:403–414, 2019.

Xiaofei Zhou, Wenhan Jiang, Yingjie Tian, and Yong Shi. Kernel subclass convex hull sample selection method for svm on face recognition. *Neurocomputing*, 73(10-12):2234–2246, 2010.