2013 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 22–25, 2013, SOUTHAMPTON, UK

PARAMETER DESIGN TRADEOFF BETWEEN PREDICTION PERFORMANCE AND TRAINING TIME FOR RIDGE-SVM

Rasmus Rothe, Yinan Yu, S.Y. Kung

Princeton University
Dept. of Electrical Engineering
Princeton, NJ 08544

ABSTRACT

It is well known that the accuracy of classifiers strongly depends on the distribution of the data. Consequently, a versatile classifier with a broad range of design parameters is better able to cope with various scenarios encountered in real-world applications. Kung [1] [2] [3] presented such a classifier named Ridge-SVM which incorporates the advantages of both Kernel Ridge Regression and Support Vector Machines by combining their regularization mechanisms for enhancing robustness. In this paper this novel classifier was tested on four different datasets and an optimal combination of parameters was identified. Furthermore, the influence of the parameter choice on the training time was quantified and methods to efficiently tune the parameters are presented. This prior knowledge about how each parameter influences the training is especially important for big data applications where the training time becomes the bottleneck as well as for applications in which the algorithm is regularly trained on new data.

Index Terms— Ridge-SVM, unified model for supervised learning, training time, parameter tuning, weight-error-curve (WEC)

1. INTRODUCTION

In the case of supervised learning, a set of training examples of the form $[\mathcal{X}, \mathcal{Y}] = \{[\mathbf{x}_1, y_1], [\mathbf{x}_2, y_2], [\mathbf{x}_N, y_N]\}$ is given. N denotes the number of training examples and $\mathbf{x}_i \in \mathcal{R}^M$ are the feature vectors, where M is its dimensionality. $y_i \in \{-1, 1\}$ is the teacher for the training vector $\mathbf{x}_i$. The aim is to find a hyperplane which separates the points with $y_i = -1$ and $y_i = 1$ with a maximum margin.

Such a hyperplane can be written as $\mathbf{x}^T\mathbf{w} - b = 0$. In the case in which the data is linearly separable one can define two hyperplanes which choose the largest possible margin to separate the data, the so-called *separation margin*. Writing these two hyperplanes as $\mathbf{x}^T\mathbf{w} - b = -1$ and $\mathbf{x}^T\mathbf{w} - b = 1$, the margin can then be derived as $\frac{2}{||\mathbf{w}||}$ so that the aim is to minimize $||\mathbf{w}||$. Given the test vector $\mathbf{x}$, the optimal linear discriminant function is then

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{w} + b. \tag{1}$$

1.1. Kernel induced vector spaces

By restricting the *decision vector* $\mathbf{w}$ to the form

$$\mathbf{w} = \sum_i^N \mathbf{x}_i a_i = \mathbf{X}\mathbf{a}, \text{ where } \mathbf{a} \equiv [a_1 \dots a_N]^T, \tag{2}$$

it is ensured that the solution is unique and its vector norm is the smallest among the feasible solutions. In matrix notation this results in the *empirical kernel space*:

$$\mathbf{X}^T\mathbf{w} + \mathbf{e}b = \mathbf{X}^T\mathbf{X}\mathbf{a} + \mathbf{e}b = \mathbf{K}\mathbf{a} + \mathbf{e}b = \mathbf{y}. \tag{3}$$

Unfortunately, in practice the data is often not linearly separable. In these cases a nonlinear decision boundary needs to be adopted by defining a new distance metric. In the kernel approach, a kernel function is used to calculate a nonlinear inner-product which results in the new distance metric. This metric is defined over a kernel-based vector space as

$$K(\mathbf{x}, \mathbf{x}') = \vec{\phi}(\mathbf{x})^T\vec{\phi}(\mathbf{x}'). \tag{4}$$

$\vec{\phi}(\mathbf{x}) = [\phi^{(1)}(\mathbf{x}), \phi^{(2)}(\mathbf{x}), ..., \phi^{(J)}(\mathbf{x})]^T$ is the induced vector, where J is the number of basis functions which can be either finite or infinite. Note that $\min(J, N)$ is generally the rank of $\mathbf{K}$. Thus $\mathbf{K}$ is nonsingular when $J > N$. Kernel based discriminant analysis can be found in the literature, e.g. in [3] [4] and [5].

For all the experiments conducted in this paper a *Gaussian RBF Kernel* defined as

$$K(\mathbf{x}, \mathbf{x}') = exp\left\{-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2}\right\} \tag{5}$$

was used. It is one of the most popular kernel functions. This is partially due to the fact that it involves an infinite number of basis functions ($J \to \infty$), and therefore the rank of $\mathbf{K}$ will be N which in turn implies that it will be non-singular.

978-1-4799-1180-6/13/$31.00

1.2. Kernel Discriminant Analysis (KDA)

Eq. 3 can be extended to incorporate nonlinear kernel functions. Through a linear mapping this leads to the decision function similarly to Eq. 1 of the form

$$f(\mathbf{x}) = \mathbf{u}^T \vec{\phi}(\mathbf{x}) + b \tag{6}$$

$$= \sum_{i=1}^{N} a_i \vec{\phi}(\mathbf{x}_i)^T \vec{\phi}(\mathbf{x}) + b \tag{7}$$

$$= \mathbf{a}^T \vec{\mathbf{k}}(\mathbf{x}) + b \tag{8}$$

with $\vec{\mathbf{k}}(\mathbf{x}) = \begin{bmatrix} K(\mathbf{x}, \mathbf{x}_1) & K(\mathbf{x}, \mathbf{x}_2) & \dots & K(\mathbf{x}, \mathbf{x}_N) \end{bmatrix}^T$. The decision rule can then be expressed as

$$\text{sign}[f(\mathbf{x})] = \text{sign}\left[\sum_{i=1}^{N} K(\mathbf{x}, \mathbf{x}_i) a_i + b\right]. \tag{9}$$

By definition the decision-hyperplane must be orthogonal to the decision vector $\mathbf{w}$. If the data-hyperplane is represented by its normal vector $\mathbf{p}$, and thus $\mathbf{X}^T\mathbf{p} = \mathbf{e}$, then this orthogonality implies that $\mathbf{w}^T\mathbf{p} = 0$, or $\mathbf{a}^T\mathbf{e} = 0$ in the empirical kernel space. This is called the Orthogonal-Hyperplane Property (OHP).

Minimizing the margin and thus $||u||$ in the empirical kernel space, the OHP property and the restriction $\mathbf{w} = \mathbf{X}\mathbf{a}$ lead to the kernel-matrix-based optimization formulation

$$\max_{\mathbf{a}} L(\mathbf{a}) = \mathbf{a}^T\mathbf{y} - \frac{1}{2}\mathbf{a}^T\mathbf{K}\mathbf{a} \tag{10}$$

$$\text{subject to } \mathbf{a}^T\mathbf{e} = 0. \tag{11}$$

It can be shown that this optimization problem has the closed form solution $\mathbf{a} = \mathbf{K}^{-1}(\mathbf{y} - b\mathbf{e})$ where b can be derived as $b = \frac{\mathbf{y}^T\mathbf{K}^{-1}\mathbf{e}}{\mathbf{e}^T\mathbf{K}^{-1}\mathbf{e}}$ which is referred to as the *Kernel Discriminant Analysis* (KDA).

1.3. Kernel Ridge Regression (KRR)

In order to reduce the sensitivity to random noise of the classifier, the kernel matrix can be perturbed to $\mathbf{K} + \rho\mathbf{I}$ which is called *Perturbational Discriminant Analysis* (PDA) or *Kernel Ridge Regression* (KRR) [1]. Intuitively, by adding a constant term to the kernel matrix $\mathbf{K}$, the algorithm becomes less dependent on the specific training data mitigating the overfitting problem. KRR can be written as the solution of an optimization problem of the following form

$$\max_{\mathbf{a}} \left\{ \mathbf{a}^T\mathbf{y} - \frac{1}{2}\mathbf{a}^T[\mathbf{K} + \rho I]\mathbf{a} \right\} \tag{12}$$

subject to $\mathbf{a}^T\mathbf{e} = 0$ and thus similarly to KDA. The introduced penalty term – controlled by the ridge factor ρ – avoids the oversubscription of vulnerable and weak components in the spectral space as shown in [3]. This in turn can prevent over-fitting.

1.4. Support Vector Machines (SVM)

The objective of *Support Vector Machines* (SVM), first presented by Vapnik [6], is to find the optimal vector $\mathbf{a}$ for

$$\max_{\mathbf{a}} \left\{ \mathbf{a}^T\mathbf{y} - \frac{1}{2}\mathbf{a}^T\mathbf{K}\mathbf{a} \right\} \tag{13}$$

subject to the OHP constraint $\mathbf{a}^T\mathbf{e} = 0$ and $0 \leq \alpha_i \leq C$ with $\alpha_i = a_i y_i$. In this case the penalty factor C is the parameter allowing for a softer separation margin. More specifically, a small C value will increase the number of support vectors. Because the final decision boundary depends on a weighted combination of support vectors a higher number of support vectors makes the classifier more stable – though at a potential loss of accuracy.

2. RIDGE-SVM

Noting the similarity between the objective function of KRR (Eq. 12) and SVM (Eq. 13) Kung [1] [2] [3] presented a combined classifier *Ridge-SVM* (formerly *PDA-SVM*):

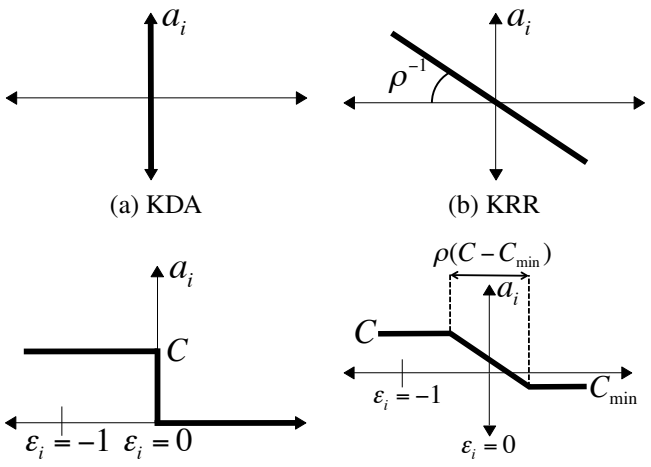
$$\mathbf{a} = \underset{\mathbf{a}}{\text{argmax}} \left(\mathbf{a}^T\mathbf{y} - \frac{1}{2}\mathbf{a}^T(\mathbf{K} + \rho\mathbf{I})\mathbf{a} \right) \tag{14}$$

subject to $\mathbf{a}^T\mathbf{e} = 0$, $C_{\text{min}} \leq \alpha_i \leq C$ with $\alpha_i = a_i y_i$. The discriminant function is $f(\mathbf{x}) = \sum_{i=1}^{N} a_i K(\mathbf{x}_i, \mathbf{x}) + b$.
b can be derived according to the KKT conditions. This hybrid classifier now combines the parameters from KRR and SVM so that C_{min}, C and ρ can be separately adjusted in order to improve the accuracy. Note that ρ in KRR and C in SVM complement each other since increasing ρ avoids overfitting, whereas decreasing C leads to over-fitting. Furthermore, the variance σ of the non-linear kernel needs to be adjusted so that in total four parameters need to be tuned at the same time. But while this allows for a better fitting to the structure of the data, one has to search a four dimensional space for the optimal set of parameters.

3. WEIGHT-ERROR-CURVE DESIGN FOR RIDGE-SVM

The error margin associated with the i-th training vector is denoted as $\epsilon_i = y_i - f(\mathbf{x}_i)$. Remembering that the classification reflects a voting on the i-th vector with weight $|a_i|$ according to Eq. 9, the *weight-error-curve* (WEC) then shows the relationship between ϵ_i and a_i. It can be derived that in the case of KRR the WEC is a straight line with a negative slope controlled by ρ as shown in Figure 1b. For the special case where $\rho = 0$, this results in KDA which is shown in Figure 1a. The main advantage of KRR lies in the smooth transition in the center region, in comparison to SVM as shown in Figure 1c where there is an abrupt drop in the WEC. However, the disadvantage of KRR lies in the two tail ends where so-called

anti-support vectors are assigned excessively large weights. SVM in comparison has constant weights in both tails. This again supports the argument for a hybrid classifier where the advantages of both classifiers are combined: Figure 1d shows the WECs for Ridge-SVM which is a combination of all three WECs.



(a) KDA (b) KRR

(c) SVM (d) Ridge-SVM: $C_{min} \leq 0$

Fig. 1. Weight-error-curves (WECs) for positive training vectors for various classifiers. Ridge-SVM incorporates the advantages of the WECs of both KRR and SVM.

3.1. Special cases of Ridge-SVM

Since Ridge-SVM is a reunification of KRR and SVM, these two can be viewed as special cases of Ridge-SVM as shown in Figure 2. More specifically, when $C_{min} = -\infty$ and $C = \infty$ the optimization formulation reduces to KRR. Additionally setting $\rho = 0$ further reduces KRR to KDA. In the same way, SVM can be seen as a special case of Ridge-SVM where $\rho = 0$ and $C_{min} = 0$. Note that these special cases can also be deduced visually from the WEC in Figure 1d by setting ρ, C and C_{min} accordingly.

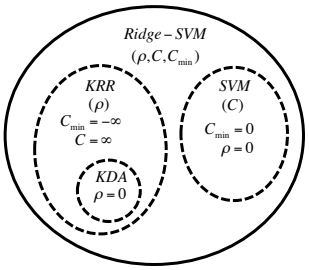


Fig. 2. The existing classifiers KRR, KDA and SVM can be seen as special cases of Ridge-SVM.

4. EXPERIMENTAL RESULTS

In order to show that the additional parameters in Ridge-SVM lead to an improved accuracy experiments on four UCI datasets [1] were conducted to find the optimal set of parameters for σ, C_{min}, C and ρ through exhaustive search. While tuning these parameters we noticed a significant change in training time. Thus we further investigated how, given optimal parameters, changing one of the parameters affects the training time and the prediction accuracy.

4.1. Optimal parameters

Kung et al. [2] conducted experiments on 6 datasets from the UCI machine learning repository and on microarray cancer diagnosis using Ridge-SVM and compared its performance against SVM, LSS (linear least squares), KDA and KRR. In these experiments the parameter σ for the RBF kernel was optimized for the conventional SVM and then applied with the same value to all other classifiers. The upper bound on α_i was set to 10, so that $C = 10$. Regarding the prediction performance, a direct consequence of using such a relatively small C is that only the most selective subset of training vectors can be retained which will in turn adversely affect the prediction capability. Using a grid search for $\rho \in \{0, 1, 2\}$ and $C_{min} \in \{\pm 0.1, \pm 0.5, \pm 1\}$ the best combination of parameters was found.

This very limited grid search on two parameters is extended in this work. Here all four parameters are varied extensively to find the optimal combination. Early experiments showed that for all the datasets increasing C beyond 10 yielded no additional improvement and lowering C generally worsened the accuracy. Thus, for all experiments C was fixed to 10 as before.

The accuracy of the parameters was measured using a leave-one-out cross validation to ensure consistency of the achieved accuracy when repeating the experiments. For datasets with more than two classes we performed a one-versus-all scheme as described in [3]: given a k-class problem, a one-versus-all classifier contains k binary classifiers, each trained to separate one of the k mutually exclusive classes from the rest. In other words, for the k-th binary classifier, patterns belonging to the k-th class are considered to be positive, while the rest is considered to be negative. The class which has the highest total score is identified as the most likely class for the test example.

The features for the wine and liver datasets had to be normalized as their features differed by several orders of magnitude which is generally problematic for kernel methods. In order to preserve the structure of the original data as much as possible, a $\log(X + 1)$ normalization was performed on the two datasets. We compared the results from the extensive parameter tuning of the Ridge-SVM algorithm with those

[1] http://archive.ics.uci.edu/ml/

from the previous experiments conducted in [2]. As shown in Table 1, extensive parameter tuning makes even more apparent the advantage of Ridge-SVM over SVM, LSS, KDA and KRR: for three out of four datasets the accuracy increased by 8.41%, 0.87% and 0.67% respectively compared to the previously chosen parameters presented in [2]. Relatively to the accuracy of SVM, the next best algorithm, Ridge-SVM, is better for all four datasets by 1.33%, 0.57%, 1.16%, 9.34% respectively.

Dataset	Iris	Wine	Liver	Glass
No. of samples (N)	150	178	345	214
No. of features (M)	4	13	6	9
No. of classes	3	3	2	6
Accuracy [%]				
SVM	96.00	98.31	73.04	64.02
LSS*	84.00	96.63	53.33	38.32
KDA*	84.00	92.70	53.33	38.32
KRR	95.33	91.57	72.17	56.07
Ridge-SVM				
Parameters from [2]	96.67	98.88	73.33	64.95
New parameters	**97.33**	**98.88**	**74.20**	**73.36**
Improvement over				
parameters from [2]	+0.67	+0	+0.87	+ 8.41
SVM	+1.33	+0.57	+1.16	+ 9.34
Improved parameters				
C_{min}	-0.5	-1	-1	-0.125
C	10	10	10	10
ρ	1.5	0.5	0.75	0.0625
σ	1	0.5	1	0.6

Table 1. Ridge-SVM consistently performs better than any of the existing algorithms. * Note that repeated examples were removed from the Iris dataset because they cause poor performance for LSS and KDA when computing the Kernel inverse. In order to keep the size of the dataset the same repeated examples were allowed in the test set for the cross-validation.

4.2. Tradeoff: prediction performance vs. training time

While tuning the parameters for the Ridge-SVM classifier, we noticed that the choice of the parameters significantly affects the training time. Especially for big data applications where the training process is very time consuming, prior knowledge about how each parameters influences the training time can significantly reduce the time spent on finding optimal parameters. Furthermore, in some applications when the algorithm has to be trained regularly on new incoming training data, a slightly worse prediction accuracy can be accepted in exchange for a much faster training time.

In this paper we study the relationship between the choice of parameters and the training time by running experiments using Matlab. The quadratic optimization problem was solved using the built in *quadprog* tool. All experiments were performed using the *trust-region-reflective* algorithm. Note that, in terms of the training time, the *interior-point-convex* and *active-set* algorithm yielded results in the same order.

Ridge-SVM can be tuned with three parameters C_{min}, C and ρ. Furthermore, σ, the variance of the kernel can be adjusted.

Varying σ: all experiments showed, that even though the choice of σ significantly affects the accuracy, it does not impact the training time. This is due to the fact that, although it introduces a new distance metric to allow nonlinear decision boundaries, it does not put tighter constraints on the optimization problem. For all 4 datasets, values for σ in the range of $[0.5\ 1]$ yielded the most accurate results. However, this strongly depends on the structure of the data and can vary greatly for other datasets.

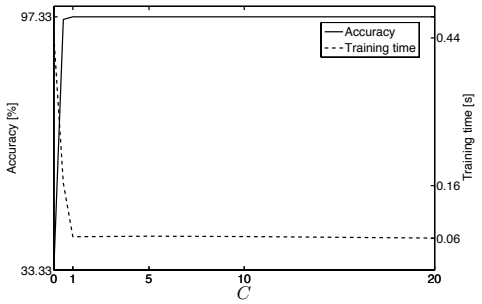
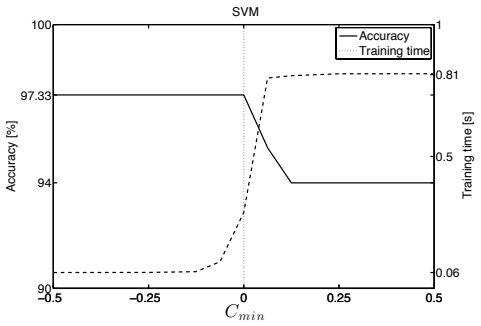


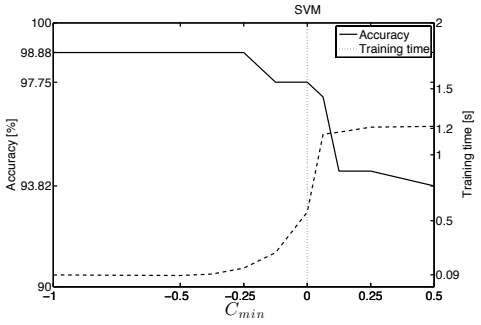
Fig. 3. Iris ($C_{min} = -0.5, \rho = 1.5, \sigma = 1$)

Varying C: somewhat surprisingly for the parameter C, reducing its value only increased the training time and decreased the accuracy in all cases as shown in Figure 3 for the iris dataset and thus was just kept constant at $C = 10$ for all experiments. The tuning of C_{min} and ρ are far from being so straightforward. In the two main experiments, ρ and C_{min} were individually tuned while all other parameters were fixed to the optimal value as presented in the previous section. We then compared learning speed and prediction performance as shown in Figure 4 for C_{min} and Figure 5 for ρ.

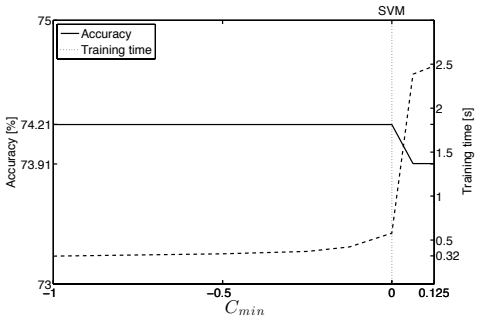
Varying $\mathbf{C_{min}}$: in the case of C_{min}, for three out of four datasets (i.e. except for the glass dataset), a lower value yielded a better prediction performance. For the iris and liver dataset $C_{min} = 0$ which equals ρ-adjusted SVM yielded the highest accuracy. For the glass dataset $C_{min} > 0$ gave the best accuracy whereas for the wine dataset $C_{min} < 0$ gave the best accuracy. Thus, C_{min} can take both positive and negative values depending on the structure of the data. In the same way when C_{min} is increased and approaches 0, training suddenly takes up to 10 times longer. Given the optimization formulation for Ridge-SVM, this is due to the following fact: for a fixed C, increasing C_{min} tightens the constraints for α_i (given by $C_{min} \leq \alpha_i \leq C$) and thus limits the choices for the α_i's which in turn makes the optimization problem more



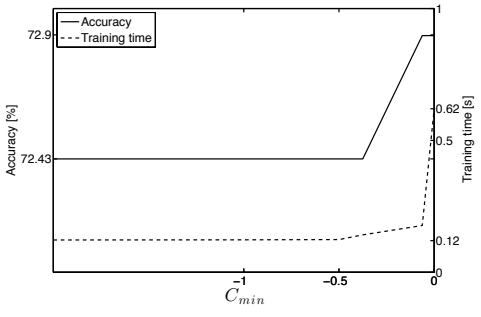
(a) Iris ($C = 10, \rho = 1.5, \sigma = 1$)



(b) Wine ($C = 10, \rho = 0.5, \sigma = 0.5$)

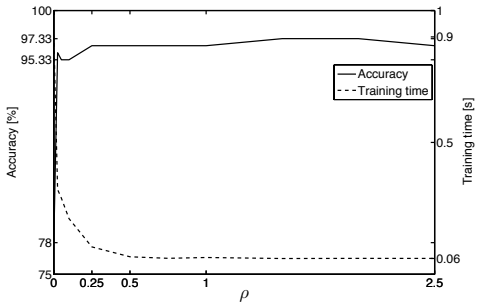


(c) Liver ($C = 10, \rho = 0.75, \sigma = 1$)

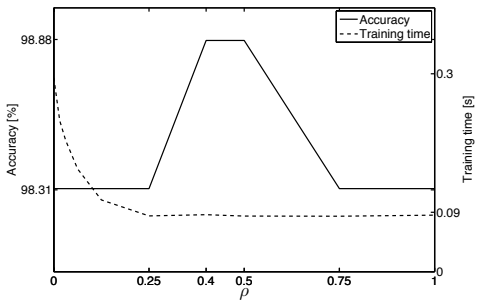


(d) Glass ($C = 10, \rho = 0.4, \sigma = 0.6$)

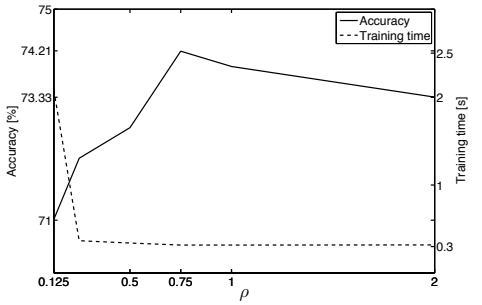
Fig. 4. As C approaches 0, the training time significantly increases. Except for the glass dataset, a lower C_{min} value yields a better accuracy. Thus, only in the case of the glass dataset, the user faces a tradeoff between accuracy and training time.



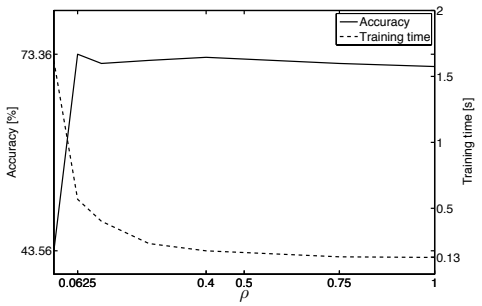
(a) Iris ($C_{min} = -0.5, C = 10, \sigma = 1$)



(b) Wine ($C_{min} = -1, C = 10, \sigma = 0.5$)



(c) Liver ($C_{min} = -1, C = 10, \sigma = 1$)



(d) Glass ($C_{min} = -0.125, C = 10, \sigma = 0.6$)

Fig. 5. For each dataset a distinct peak value for ρ yields the highest accuracy. The training time increases significantly when ρ approaches 0. For the glass dataset there is a tradeoff between training time and accuracy.

difficult. In order to explain why the training time suddenly increases as C_{min} increases above zero, we note that this disallows α_i from being negative. Recalling that $\alpha_i = a_i y_i$, in order to fulfill the OHP constraint that $\mathbf{a}^T\mathbf{e} = 0$, the signs of the a_i values are fixed based on the training labels. Thus it restricts the choices for α_i much more than in the case where α_i can be positive or negative. Except for the glass dataset, there is no tradeoff between training time and accuracy and it is advised to choose a negative value for C_{min} to keep the training time low. In order to be sure that there is no larger C_{min} value for which the accuracy increases C_{min} can be tuned by starting off with $C_{min} = -\infty$ and then increasing the value. If at some point the accuracy drops C_{min} should be kept at a low value. However, if the accuracy suddenly increases, as in the case of the glass dataset, one faces the tradeoff between accuracy and training time, especially when the training dataset is large or there is new training data arriving frequently and thus the algorithm has to re-run often. In the case of the glass dataset (Figure 4d) increasing C_{min} from -0.375 to -0.125 increases the accuracy by just 0.47% at the expense of a 25% longer training time. However, if C_{min} is now further increased to 0, the training time is 4 times as long as for the case where $C_{min} = -0.375$ at no additional accuracy. Thus one has to be careful when choosing C_{min} so that the training time does not significantly increase and preferably keep $C_{min} < 0$.

Varying ρ: in the case of ρ, for all four dataset, there exists a relatively small region of values for which Ridge-SVM yields the best prediction performance. If ρ is chosen too large, the accuracy drops which is due to the fact that the updated kernel matrix $\mathbf{K} + \rho\mathbf{I}$ is then dominated by ρ which results in under-fitting. Similarly, if ρ is set to 0, this can lead to over-fitting so that the accuracy drops. The experimental results suggest that the training time increases exponentially as ρ approaches 0 which was consistent across all four datasets. This is most likely due to the fact that as ρ increases, $\mathbf{K} + \rho\mathbf{I}$ is dominated by the identity matrix and thus the inverse can be calculated faster when solving the quadratic optimization problem. Thus, it is recommended to start with a relatively larger value of ρ in the order of $10 - 100$ and decrease until the accuracy stops improving. If ρ has to be decreased too much to reach the peak accuracy, it can lead to a tradeoff between accuracy and training time: in the case for the glass dataset, $\rho = 0.0625$ yielded the highest accuracy of 73.36% as opposed to 72.9% with $\rho = 0.4$ at the expense of a twice as long training time.

5. CONCLUSION

It is well known that the accuracy of classifiers strongly depends on the distribution of the data. Thus it cannot be known in advance which learning algorithm and what parameters yield the highest prediction accuracy. In this paper Ridge-SVM, a unified model for kernel-based supervised classification which allows extensive parameter tuning was applied to four UCI datasets. The additional parameters come at the expense of having to spend more time on the tuning. However, as shown in this paper, it yields a higher accuracy than SVM and KRR by 1.33 %, 0.57%, 1.16 %, 9.34% respectively. The influence of the parameter choice on the training time was quantified and methods to efficiently tune the parameters are presented. It was shown that when both ρ and C_{min} approach 0 the training time significantly increases and often the accuracy drops. However, in some cases this extra training time can result in a better accuracy of the classifier.

In future work, Ridge-SVM should be applied to a wider range of datasets in order to further demonstrate its flexibility through the range of parameters to yield a higher accuracy than SVM and KRR.

6. ACKNOWLEDGMENTS

This material is based on research by DARPA under agreement number FA8750-12-2-0126. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, the U.S. Government, or Princeton University.

7. REFERENCES

[1] S.Y. Kung, "Kernel approaches to unsupervised and supervised machine learning," in *Advances in Multimedia Information Processing-PCM 2009*, pp. 1–32. Springer, 2009.

[2] S.Y. Kung and Man-Wai Mak, "Pda-svm hybrid: A unified model for kernel-based supervised classification," *Journal of Signal Processing Systems*, vol. 65, no. 1, pp. 5–21, 2011.

[3] S.Y. Kung, *Kernel Methods and Machine Learning*, Cambridge University Press, 2014.

[4] Bernhard Schölkopf and Christopher JC Burges, *Advances in kernel methods: support vector learning*, The MIT press, 1999.

[5] K-R Muller, Sebastian Mika, Gunnar Ratsch, Koji Tsuda, and Bernhard Scholkopf, "An introduction to kernel-based learning algorithms," *Neural Networks, IEEE Transactions on*, vol. 12, no. 2, pp. 181–201, 2001.

[6] Vladimir Vapnik, *The nature of statistical learning theory*, springer, 1999.