Local Adaptive SVM for Object Recognition

Nayyar A.Zaidi, David McG. Squire
Clayton School of Information Technolgoy, Monash University, Clayton VIC 3800, Australia
Email: {nayyar.zaidi, david.squire}@infotech.monash.edu.au

***Abstract*—The Support Vector Machine (SVM) is an effective classification tool. Though extremely effective, SVMs are not a panacea. SVM training and testing is computationally expensive. Also, tuning the kernel parameters is a complicated procedure. On the other hand, the Nearest Neighbor (KNN) classifier is computationally efficient. In order to achieve the classification efficiency of an SVM and the computational efficiency of a KNN classifier, it has been shown previously that, rather than training a single global SVM, a separate SVM can be trained for the neighbourhood of each query point. In this work, we have extended this Local SVM (LSVM) formulation. Our Local Adaptive SVM (LASVM) formulation trains a local SVM in a modified neighborhood space of a query point. The main contributions of the paper are twofold: First, we present a novel LASVM algorithm to train a local SVM. Second, we discuss in detail the motivations behind the LSVM and LASVM formulations and its possible impacts on tuning the kernel parameters of an SVM. We found that training an SVM in a local adaptive neighborhood can result in significant classification performance gain. Experiments have been conducted on a selection of the UCIML, face, object, and digit databases.**

***Keywords*-local methods, metric learning, nearest neighbor classifier, support vector machine classification, object recognition**

I. Introduction

Support Vector Machine (SVM) classifiers, based on the principle of structured risk minimization, are a popular tool for classification. They have been shown to give state of the art performance on a wide range of classification data sets. An SVM finds an optimal hyperplane to separate data into two classes. Given a training set $S = \{(x_1, y_1), ...(x_n, y_n)\} \in \mathrm{R}^n \times \{-1, 1\}$, the decision function is found by solving the following convex optimization problem:

$$\begin{array}{cl} \max_\alpha f(\alpha) & = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j k(x_i, x_j) \\ \text{subject to} & 0 \leq \alpha_i \leq C \quad \text{and} \quad \sum_{i=1}^n \alpha_i y_i = 0 \\ \text{where} & k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \end{array} \quad (1)$$

where α are the Lagrange multipliers, C controls the misclassification penalty and $k(.,.)$ is the kernel function.

Though extremely popular and efficient, there are at least two issues that need to be addressed when training SVMs. First, SVMs are designed for binary classification problems. For multi-class classification, one has to resort to one-versus-all or one-versus-one classification strategies. This means that systems become more and more complicated as the number of classes increases. Also, any time a class is added, all classifiers have to be retrained. This results in very long training and testing times. The second issue is the tuning of the SVM parameters, for example C and kernel's length scale parameter σ in equation 1. These parameters are typically tuned through cross-validation schemes that are computationally expensive. Also, for sheer simplicity, the kernel in equation 1 is usually considered to be isotropic, that is $\sigma_1 = \sigma_2 = ... = \sigma_d$ for all d dimensions, since tuning multiple scale parameters is not feasible through cross-validation. Recently, however, Chapelle et al. [1] have shown that tuning multiple scale parameters for the kernel improves SVM classification performance.

Local methods, such as local logistic regression and K-Nearest Neighbor (KNN), provide an alternative strategy to training a complex global model. KNN in particular has proven to be very effective for classification problems [2], [3]. Not only are KNN classifiers computationally efficient, but it has been shown that, with the right distance measure, they can out-perform far more sophisticated alternatives [4]–[6]. Still, the findamental strength of KNN stems from its simplicity. Unlike SVM, the KNN classifier deals with multi-class problems effortlessly. The K neighbors of the query point x_0 vote for the class label $\hat{y}(x)$, as shown in the following equation: $\hat{y}(x) = \frac{\sum_{j=1}^K y(x_j)k(x_j,x_0)}{\sum_{j=1}^K k(x_j,x_0)}$. The vote decays smoothly as the distance of the neighbor from the query point increases (controlled by kernel $k(x_j, x_0)$). As for SVM, the tunable parameters of KNN, such as the size of neighborhood and the kernel itself ($k(x_j, x_0)$), are not easy to find and are the subject of much research [6]. The success of KNN methods, however, has taught two valuable lessons: First, good recognition performance is achieved by defining the right similarity measure to the prototypes, which corroborates early human perception studies [7]. Secondly, most of the information required to make decision about the label of a query is present in its local neighborhood.

Building on these insights from local methods, and motivated by [8], in this paper we propose to use SVMs locally. That is, rather than training a global SVM classifier, we train an SVM classifier in the neighborhood of each query point. LSVM in some ways alleviates the problems (described above) associated with SVM. Some motivations behind local SVM formulations are described below:

- A major motivation for using LSVM as reported in [8] is the gain in classification performance. As we will see in section IV, LSVM results in performance gain in some databases, but performance deteriorats in most databases. LSVM, however, has an advantage over standard SVM as it involves no training and extension to greater numbers of classes is more feasible. The testing time is likely to increase, as an SVM classifier has to be trained for every query point. Despite this, in most databases having M classes, a query point is surrounded by instances of only m classes, and usually $m \ll M$. This can expedite the recognition phase rather than slowing it down.
- Another motivation (which is not reported in [8]) for training SVMs locally has to do with the kernel's multiple scaling parameters. It is hoped that, at least locally, class conditional probabilities vary similarly across all features and using an isotropic kernel may not hurt much. This avoids the need to tune multiple kernel parameters in the SVM.

Typically in computer vision research [9], [10], to avoid expensive cross-valiation, the value of the length scale parameter (σ) of the kernel is set to be the average value of the squared distance between all data points, that is: $\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} d_i^2$. This strategy not only ensures that the kernel's numeric range is within a bounded interval but also results in achieving an optimal trade-off between bias and variance of the classifier. But using σ as an averaged value of distance between entire training data may not be optimal. Since LSVM is trained only locally, using σ as the average distance of all points in the neighborhood will be locally optimal. Thats why LSVM can result in better classification performance in some case as compared to standard SVM. As will be shown, the performance of LSVM is dependent on the neighborhood size in which it is trained. One can truly expect that LSVM performance will be better for a larger neighborhood size. But increasing neighborhood size will result in a decrease in computational efficiency. Therefore, a neighborhood size has to be tuned such that an optimal trade-off is achieved between computational and classification efficiency. An alternative to increase in classification performance but keeping the size of the neighborhood small is to train a local SVM in the adaptive neighborhood of the query point. Such formulation is called local adaptive support vector machine (LASVM). Another advantage of LASVM formulation is the alleviation of the effects of the Curse-of-Dimensionality (COD) on SVM. It has been shown that SVM classifiers are not immune to COD [11, section 12.3.4]. LASVM employs the strategy of modifying the distance metric and hence the kernel to give appropriate weights to each feature. This results in some dimensions to be completely ignored (hence feature selection), thereby reducing the dimensionality. Also, LASVM provides a framework for incorporating non-stationary kernels in the SVM framework.

The rest of the paper is organized as follows: related work is discussed in section II, we discuss our proposed algorithms LSVM and LASVM in section III. We discuss in detail our results on UCIML, face, digit, and object databases in section IV. We conclude in section V.

II. Related Work

The idea of local SVMs has been explored in some detail in [12] and [8]. In [12], the authors proposed a local SVM called a 'Profile SVM' (PSVM). The data is first clustered and a separate SVM is trained for every cluster. A query point is first assigned to its nearest cluster and the corresponding SVM is used to predict its label. Though interesting, a major drawback of the approach is how to determine the number of clusters before training. Also, results may vary depending on the clustering scheme and the number of clusters. Though results are reported on UCIML databases where PSVM is shown to outperform standard SVM, the performance of PSVM on object recognition databases is not reported. As discussed, our work is inspired by and closer to [8]. Just like [8], we train a separate SVM for each query. We keep the size of the neighborhood constant across all databases in order to study LSVM performance systematically, whereas in [8] the size of the neighborhood is optimized through cross-validation for each data-set. In section IV, we will discuss the motivation behind this strategy. We study the effects of variation in neighborhood size on LSVM performance which are not discussed and reported in [8]. This adds novelty to our results regarding LSVM. Also unlike [8], we propose Local Adaptive SVM (LASVM) which are inspired by recent successes in metric learning for k-Nearest Neighbor (KNN) methods. Learning a metric results in a linear transformation of the data. LASVM trains an LSVM in the local space transformed by the learnt metric. Though local SVM has been investigated as mentioned before, to the best of our knowledge the local adaptive SVM formulation is novel.

III. Approach

For any query point x_0, our local version of SVM (LSVM) works in the following way:

- Find the K nearest neighbors of x_0. If $N =$ cardinality of training dataset then, $K = k \times \log2(N)$ [1] , typically $k = [1, 2, 3, 5, 10]$.
- If the labels of all K points are same, x_0 belongs to the corresponding class and the procedure exits.
- If the labels are different, a one-versus-all SVM is trained for each class present, and x_0 is labeled accordingly.

[1]Any function altering neighborhood size can be used

As described, LASVM extends LSVM by training a local SVM in a transformed space, where the transformation is parameterized by a matrix A learnt using a metric learning algorithm. In this work we have used the MEGM algorithm for learning the transformation matrix A. Metric learning algorithms aim to find a linear transformation of the data such that, in the transformed space, a KNN classifier performs better. If we denote such a transformation by a matrix A, we are effectively learning a metric induced by A^TA such that $d^2(x,y) = (x-y)^TA^TA(x-y) = (Ax-Ay)^T(Ax-Ay)$. The Mean-Square-Error (MSE) Gradient Minimization (MEGM) is a simple gradient-based metric learning algorithm that has been shown to perform well on major object recognition databases [13]. MEGM is based on the minimization of an MSE objective function using a gradient descent algorithm in the KNN framework.

A. *Locally Adaptive SVM (LASVM)*

An outline of the LASVM algorithm is given in algorithm 1. Though our formulation of LASVM incorporates MEGM, other metric learning algorithms, for example NCA [14], LMNN [15] etc., can be used. A lot of work has also been done in locally adaptive metric learning algorithms, [6]. A local SVM can be trained on the transformed neighborhood that results from the local adaptive metric. We have used MEGM, as we got better results with MEGM than with NCA [14], LMNN [15] and DANN [6]. The comparison of LASVM performance with other metric learning algorithms and with different adaptive metric learning algorithms has been left as future work.

Algorithm 1 LASVM: Train an SVM classifier in the adapted neighborhood of a query point.

```
Require:
  - Testing data: $x_0$
  - Training data: $\{x_n, y_n\}_{n=1}^N$, where $x$ is a $p$ dimensional
    feature vector, $N$ is the number of training data, $y$ is
    training label $y = \{1, 2, ...M\}$, where $M$ is the number
    of classes.
  - $K = k \times \log2(N)$, typically $k = [1, 2, 3, 5, 10]$.

  - Learn a data-dependent distance matrix $L$ such that $L =
    A^TA$ using the MEGM metric learning algorithm.
  - Transform both training and testing data using matrix
    $A$.

  - Apply the LSVM procedure on the transformed training
    and testing data.
```

IV. Experimental Results

In this section we present our results on various databases. The Faces, USPS, Isolet, and Coil100 databases were preprocessed for efficiency. Pre-processing images using PCA is a common approach in object recognition experiments to reduce dimensionality. This can result in vastly reduced computational cost. In our experiments, the results are obtained by reducing the dimensionality of images by projecting data on the first few eigenfaces. The number of eigenfaces used for each database is given in table I.

The performance of LSVM and LASVM is compared with the following techniques, note that all SVM formulations are trained with a Gaussian kernel and a one-versus-all strategy is employed.

- **KNN:** 1 Nearest neighbor classification.
- **SVM:** The C parameter for the SVM is tuned through cross-validataion (searched from the set: $\{1, 10, 100, 1000\}$). The value of σ is set to the average distance of k nearest neighbor as discussed in section I (we will call this formulation 'standard SVM' in the subsequent results)
- **MEGM:** 1 Nearest neighbor classification in the transformed space. The transformation is parameterized by matrix learned using MEGM algorithm. This is used to compare our results with method in [13] whose metric learning strategy we have employed.
- **OSVM:** In the case of the UCIML databases, for a better comparison of LSVM and LASVM with standard SVM, both C and σ parameters are optimized for SVM using cross validation (we call this formulation 'optimized SVM (OSVM)'). An SVM giving the best performance from $C = \{1, 10, 100, 1000\}$ and $\sigma = \{0.1, 0.5, 1, 2, 3, 5\}$ is chosen.

For faces, USPS, Isolet and Coil100, each experiment is repeated 10 times and the mean results and standard deviations are reported. Note that no optimization is done for LSVM and LASVM. C parameter is set equal to 10 and σ is set equal to 1 for all experiments in all LSVM and LASVM formulations .

Database	#Data	#Features	#Classes	PCA	#Train/Class	#Test/Class
yalefaces	165	77760	15	50	4	7
yalefacesB	5850	307200	10	20	10	20
caltechfaces	435	47500	29	30	5	10
caltechfacesB	435	47500	2	30	50	100
AT&Tfaces	400	10304	40	50	5	5
Coil100	7200	16384	100	50	10	10
USPS	9298	256	10	50	20	10
Isolet	6238	617	26	30	20	10

Table I
Details of Faces, USPS, Isolet and Coil100 Databases used for classification

A. *Faces, USPS, Isolet and Coil Databases*

Yalefaces, YalefacesB, AT&T and Caltechfaces were used to study local adaptive SVM performance. Details of these databases are given in table I. The Yalefaces, YalefacesB and AT&T databases are well-known in face recognition research. Caltechfaces and CaltechfacesB constitute images from the face category in the Caltech-101 object database.

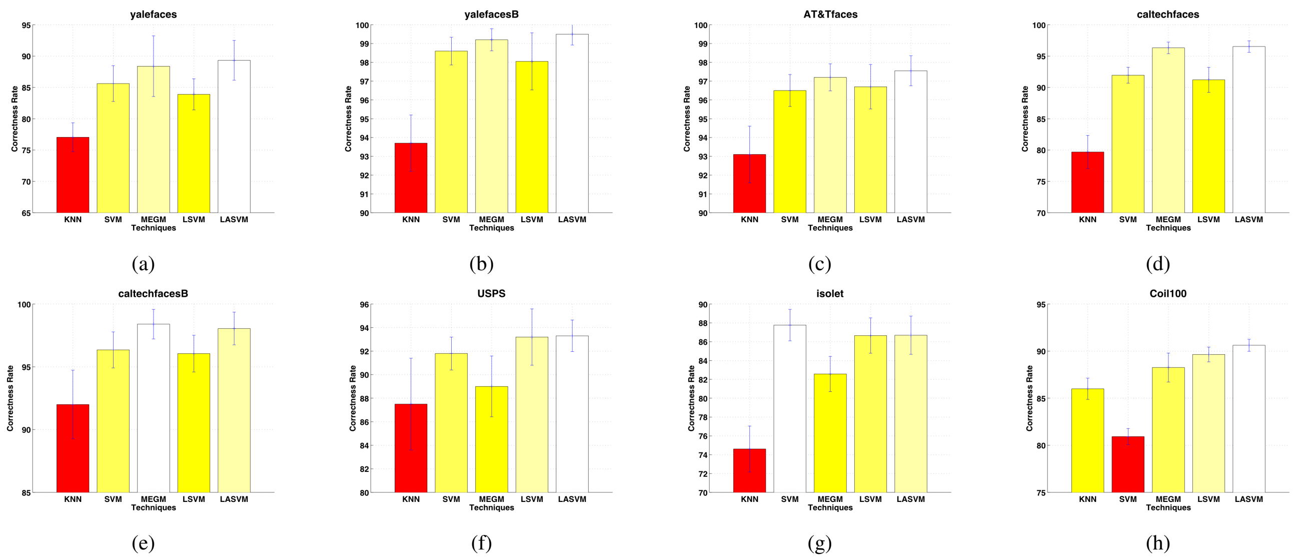

Figure 1. Comparison of the correctness rates of LSVM and LASVM method with KNN, SVM, MEGM methods on faces, USPS, Isolet and Coil100 database. LSVM and LASVM results with the best neighborhood size from figure 2 are reported for comparison.

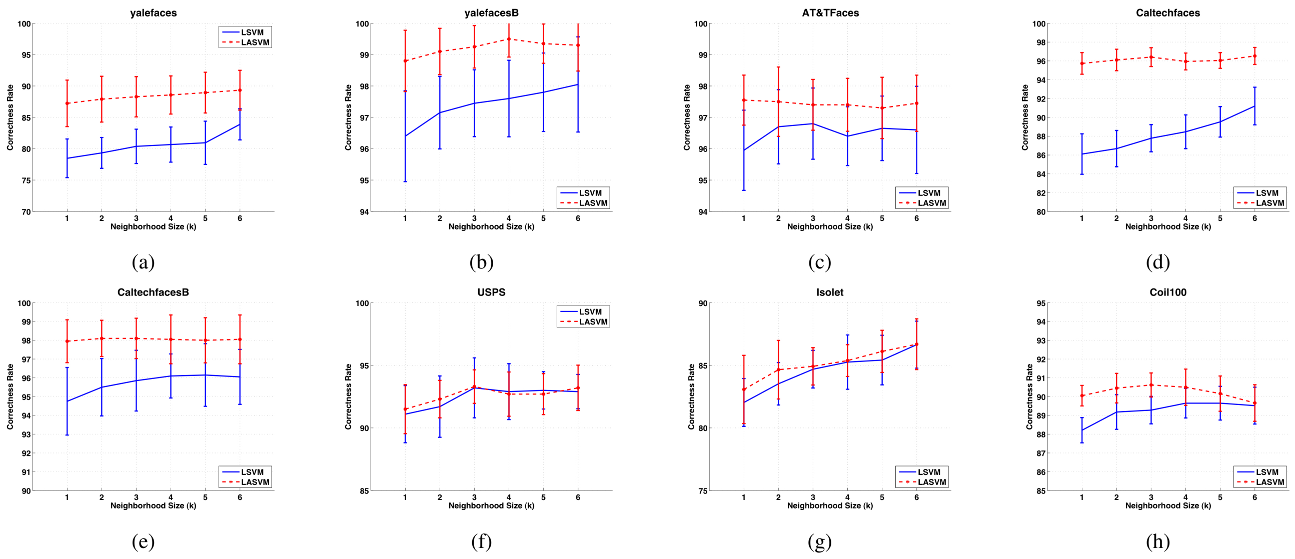

Figure 2. Comparison of LSVM and LASVM performance on faces, USPS, Isolet and Coil100 database by varying neighborhood size k.

The Caltech-101 face category has 435 images of around 20 people. The Caltechfaces database in table I is based on splitting the Caltech-101 face category into 20 categories, each belonging to a different person. On the other hand, CaltechfacesB in table I is based on splitting the Caltech-101 face category into two classes only: male and female.

LSVM and LASVM performance along with with KNN, SVM and MEGM classification performance for different databases is shown in figure 1. Our proposed LASVM results not only in the improvement in standard SVM, but also performed better or equally as compared to competing MEGM algorithm. LASVM performed best on 6 out of 8 databases whereas MEGM performed best on 1 database. On the other hand, LSVM improved standard SVM performance in the case of USPS database (figure 1(f)) and Coil100 database (figure 1(h)). In all other database, it did not result in much improvement over standard SVM. This suggests that LSVM though computationally efficient, may not be the best choice when classification efficiency is required. Figure 2 compares LSVM and LASVM performance with varying neighborhood size (k). An interesting pattern can be seen. LSVM's performance increases as k increases, whereas LASVM's performance in most cases is not affected by k. The dependence of LSVM on the neighborhood size may have motivated Zhang et al. in [8] to optimize the neighborhood size through cross-validation. LASVM's less proneness to neighborhood size as compared to LSVM suggests the importance of the right neighbors for prediction. Since in original space, neighborhood of the query point does not hold enough discriminatory information for prediction, a larger neighborhood is required. But since LASVM finds a local boundary in the transformed space, its performance is as good in bigger neighborhood as it is in the smaller one. This highlights the efficacy of LASVM formulation.

B. UCI-ML Repository Databases

The correctness rate of each method for several UCIML databases is shown in figure 3. The number of data, features and classes for each database is reported in the title. The correctness rate of each method is obtained using 40 rounds of 2-fold cross-validation. Prior to training, features were normalized to have a zero mean and a unit variance.

As mentioned before, apart from comparing LSVM and LASVM results with KNN, SVM and MEGM, results are also compared with OSVM (case where both C and σ parameters are optimized). It can be seen from figure 3, LASVM and MEGM performed equally well on most of the databases. Also LSVM and LASVM's performance followed the same trend from databases in section IV-A. That is, LASVM performed better than LSVM on most databases. Second, LSVM performance improved as neighborhood size is increased. And third, LASVM performed better than LSVM on most databases and its performance is not hugely impacted by the neighborhood size (k).

There are three things that needs to be mentioned. First, as described, no value is optimized in the case of LSVM and LASVM. Therefore our results are encouraging for not only LASVM but also for LSVM as LSVM's performance is comparable in some cases to standard SVM and OSVM. For example, in the case of Coil100 database, LSVM results in huge performance gain over standard SVM. Second, due to space constraints we have not reported computational time in this work. In our experiments we found that training an LSVM and LASVM is far more efficient in term of computational efficiency as compared to standard SVM provided we keep the neighborhood size reasonably small. Therefore, training a LSVM and LASVM can be attractive from computational efficiency point of view. Third, as depicted in our results, LSVM though improve SVM performance in some case, can deteriorates SVM performance in others. This suggests that, LSVM algorithm proposed in [8] can not be used with Euclidean distance. Note Zhang et al. in [8] have proposed to use LSVM with a specifically designed distance measure. Our locally adaptive formulation LASVM resulted in far more superior performance as compared to LSVM.

V. Conclusion

In this paper we proposed to use SVM in an adaptive local neighborhood settings. Our algorithm LASVM resulted in significant improvement in the performance of not only LSVM, but also SVM, KNN, and MEGM classifiers on faces, object, digit, and UCIML databases. We found that, unlike LSVM, LASVM performance is not affected by the neighborhood size. These results are promising and point to an interesting direction of further research. We also highlighted advantages of LSVM methods and described how they can help alleviate kernel parameter tuning problems. Our results on LSVM suggest that it should not be viewed as a replacement for SVM, but as a compromise between SVM and KNN. LSVM is especially useful in cases where the number of classes is very large, as LSVM is faster than standard SVM and has better performance than KNN. Our LASVM results, on the other hand, are encouraging and needs to be investigated further with other metric learning algorithms.

References

[1] O. Chapelle, V. Vladimir, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Journal of Machine Learning Research*, 2002.

[2] A. Frome, Y. Singer, and J. Malik, "Image retrieval and classification using local distance functions," in *Proceedings of Neural Inforamtion and Processing Systems*, 2006.

[3] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1447–1454.

[4] A. Berg, T. Berg, and J. Malik, "Shape matching and recognition using low distortion correspondence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[5] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Recognition*, 2005.

[6] T. Hastie and R. Tibshirani, "Discriminative adaptive nearest neighbor classification," *IEEE transactions on Pattern Analysis and Machine Intelligence*, 1996.

[7] E. Rosch, "Natural categories," *Cognitive Psychology*, 1973.

[8] H. Zhang, A. Berg, M. Maire, and J. Malik, "Svm-knn: Discriminative nearest neighbor classification for visual category recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[9] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in *Proceedings of the International Conference on Computer Vision*, 2007.

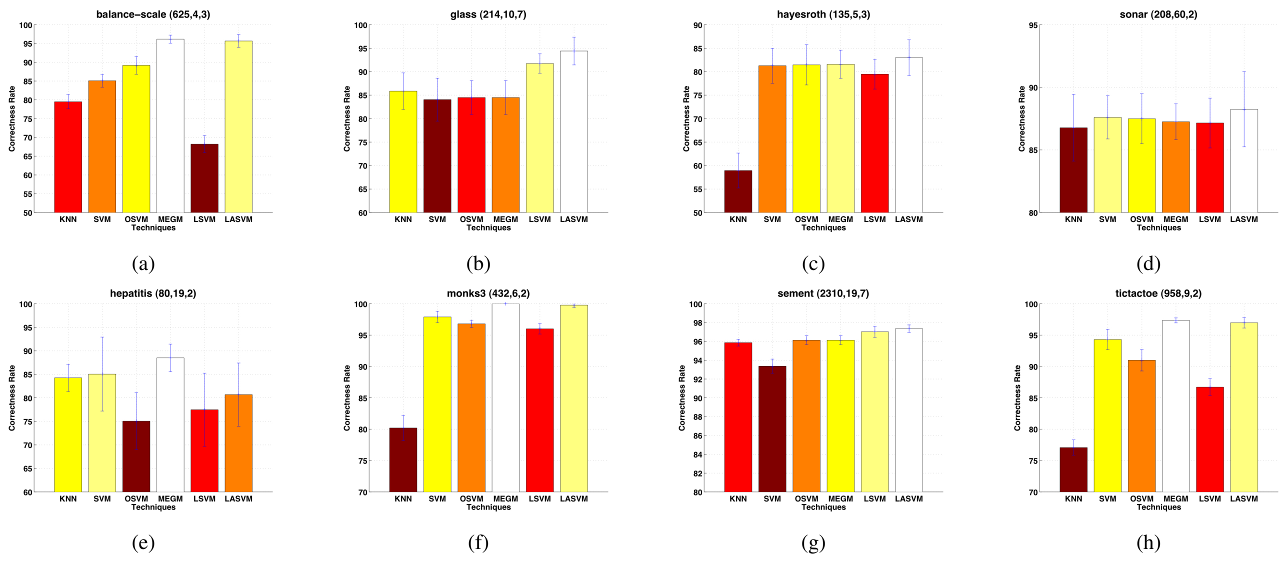

Figure 3. Comparison of the correctness rate of LSVM and LASVM with KNN, SVM, OSVM, MEGM methods on UCIML databases, balance (3(a)), glass (3(b)), hayesroth (3(c)), sonar (3(d)), hepatitis (3(e)), monks3 (3(f)), segment (3(g)) and tictactoe (3(h)). LSVM and LASVM results with the best neighborhood size from figure 4 are reported for comparison.

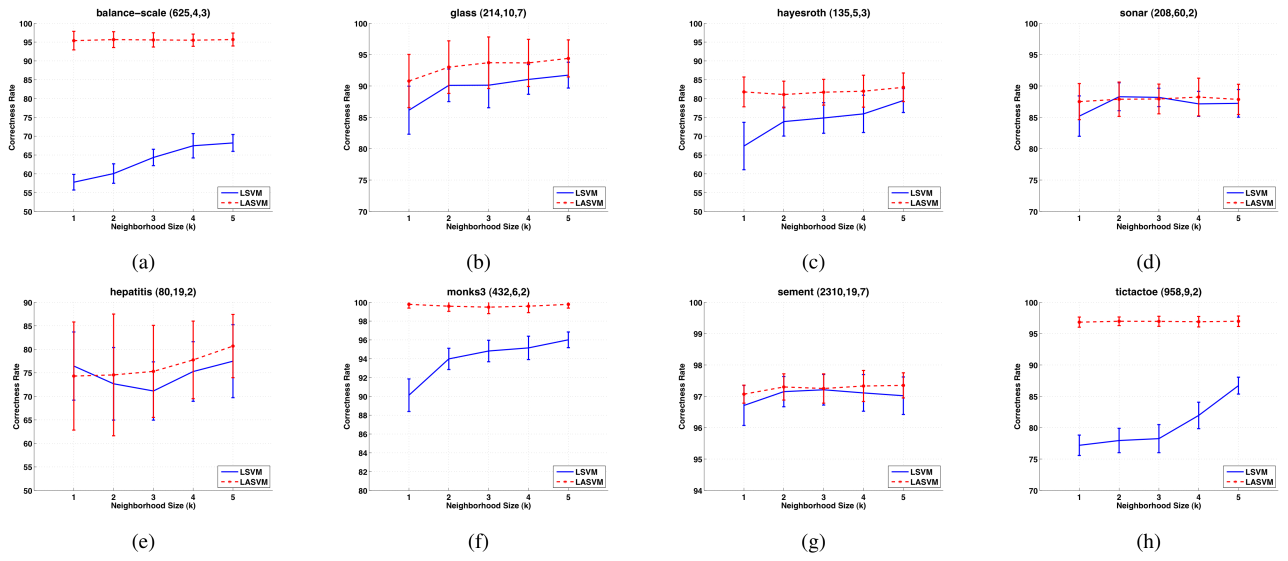

Figure 4. Comparison of the correctness rate of LSVM and LASVM by varying the neighborhood size k on UCIML databases, balance (4(a)), glass (4(b)), hayesroth (4(c)), sonar (4(d)), hepatitis (4(e)), monks3 (4(f)), segment (4(g)) and tictactoe (4(h)).

[10] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, “Local features and kernels for classification of texture and object categories: A comprehensive study,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (Workshop paper)*, 2006.

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics, 2001.

[12] H. Cheng, P. Tan, and R. Jin, “Localized support vector machine and its efficient algorithm,” in *Proceedings of the SIAM International Conference on Data Mining*, 2007.

[13] N. Zaidi and D. M. Squire, “A gradient-based metric learning algorithm for k-nn classifiers,” in *Proceedings of the Australasian Joint Conference on Artificial Intelligence*, 2010.

[14] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, “Neighborhood component analysis,” in *Proceedings of Neural Inforamtion and Processing Systems*, 2005.

[15] K. Weinberger, J. Blitzer, and L. Saul, “Distance metric learning for large margin nearest neighbor classification,” in *Proceedings of Neural Inforamtion and Processing Systems*, 2006.