

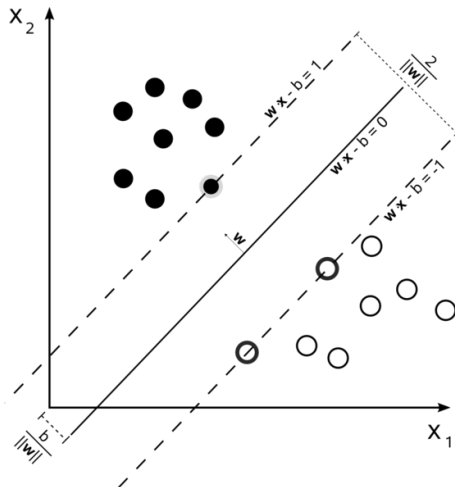
Choosing the kernel parameters for SVMs by the
inter-cluster distance in the feature space
Authors: Kuo-Ping Wu, Sheng-De Wang
Published 2008

Presented by: Nandini Deka
UH Mathematics Spring 2014 Workshop "Kernel Based Automatic
Learning"

April 29, 2014

Introduction

SVMs are used as binary classifiers which constructs a separating hyperplane, for classification, regression etc. The hyperplane is represented by a small number of data points, called support vectors (SVs).



Kernel types:

- Linear : $K(x, x') = \langle x, x' \rangle$
- Polynomial : $K(x, x') = (\gamma \langle x, x' \rangle + r)^d$, $\gamma > 0$
- Gaussian : $K(x, x') = \exp(-\gamma \|x - x'\|^2)$, $\gamma > 0$

The values $\{d, \gamma\}$ and γ in the polynomial and gaussian kernels are the kernel parameters which need to be chosen optimally for the SVM.

SVM formulation

Training data set (x_i, y_i) , $i = 1, 2, \dots, l$, $x_i \in R^n$, $y_i \in \{+1, -1\}$, where R^n is the input space, x_i the sample vector and y_i the class label of x_i .

The separating hyperplane (w, b) is a linear discriminating function, that solves the optimization problem:

$$\min_{w, b, \xi} \langle w, w \rangle + C \sum_{i=1}^l \xi_i$$

$$\text{subject to } y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i, i = 1, 2, \dots, l, \xi_i \geq 0$$

Here C is the penalty parameter of error, about which there is no a priori knowledge.

Model Selection

Aim: to use different SVM training methods to choose the optimal combination of parameters $\langle C, \text{kernel parameters} \rangle$ for the classifier with the maximum testing accuracy.

Methods already used:

- Grid search algorithm for all values of $\langle C, \text{kernel parameters} \rangle$
- Use the receiver operating characteristic (ROC) curve , as done by Ancona et al. (2002)
- Genetic Algorithms (GAs) (2002 - 2005)

Training time complexity (for a sample size l) : $O(l) - O(l^2)$.

In the worst case it has been found to be : $O(l^4)$ (Hush and Scovel, 2003)

Separation index

The separation index represents the degree at which the data are separated in the self reproducing Hilbert space, H , defined by the parameters and the corresponding kernel functions.

Cluster validation index

The clustering problem is equivalent to partitioning the unlabeled data into subsets, with each subset being a cluster. The clusters should be as far away as possible, in addition to not overlapping with each other. To judge the degree of the separation of clusters several inter-cluster distance functions are used to validate whether the data are well-clustered by a specific partition.

Proposed method

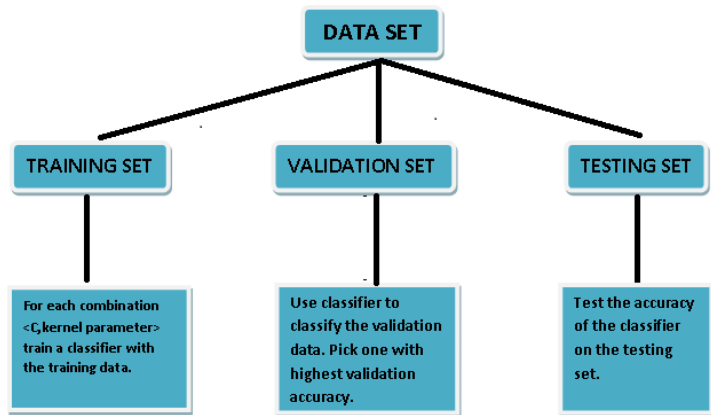
Use one of the Cluster Validation measures as the data separation index to predict possible good SVM parameter combination of :

- the kernel parameters
- the penalty parameter of error, C , is the weight of the slack variables or the error terms.

Experimental Results:

- The parameter combinations with higher values of the cluster validation index, have high testing accuracy in most cases.
- The computation time is much shorter than training many SVMs to select one parameter combination.

Grid Search Method:



- When the validation data are not available, k-fold cross validation can be used to acquire the validation accuracy.

Other indices used to estimate accuracy

- Takahashi used the ratio of the number of Support Vectors to the size of the training set as an index. (2003)
- Phetkaew proposed using the SVM margin to identify a classifier that causes wrong classifications. (2002)
- Bezdek and Pal mentioned several inter-cluster distance measures (δ_i) as an index. (1998)
- Wu and Wang proposed an index derived from distance between the mean of the two classes (clusters). Bezdek used this index for unsupervised data clustering. (2006)

Inter-Cluster distances

In the sample space the inter-cluster distances can be :

- shortest dist: $\delta_1(X, Y) = \min_{x \in X, y \in Y} d(x, y)$
- longest dist: $\delta_2(X, Y) = \max_{x \in X, y \in Y} d(x, y)$
- average dist: $\delta_3(X, Y) = \frac{1}{l_+ + l_-} \sum_{x \in X, y \in Y} d(x, y)$
- distance between two class means:
$$\delta_4(X, Y) = d(m_+, m_-) = d\left(\frac{\sum_{x \in X} x}{l_+}, \frac{\sum_{y \in Y} y}{l_-}\right)$$
- $$\delta_5(X, Y) = \frac{1}{l_+ + l_-} \left(\sum_{x \in X} d(x, m_-) + \sum_{y \in Y} d(y, m_+) \right)$$

where X and Y are positive and negative classes, l_+ and l_- are the sample sizes resp., and m_+ and m_- are the class means of X and Y .

Distances in the Hilbert space:

The distances can be calculated in the Hilbert space associated to the kernel, with the Hilbert space norm and the kernel function incorporated. The distance between two samples in the Hilbert space can be evaluated as:

$$d(\phi(x), \phi(y)) = \sqrt{|\phi(x) - \phi(y)|_2^2} = \sqrt{K(x, x) + K(y, y) - 2K(x, y)}$$

This can be used to calculate the inter cluster distance measures δ_{4f} and δ_{5f} in the Hilbert space.

Choosing $\langle C, \text{kernel parameters} \rangle$

- The indices quantify the degree of separation of the classes and thus can estimate the classifier generalization ability. Thus the validation accuracy can be substituted by using the separation indexes to choose the kernel parameters.
- Since calculating the index values involve the kernel functions, the kernel parameters can be chosen to maximize separation index.
- Choose only the penalty parameter C by the validation process.
- The training time is composed of the calculation time of all separation index values and the SVM model training time for one kernel parameter combination and all C values.

Using distance index to choose a desired classifier

for each choice of kernel parameter combination

→ calculate the distance index of the training data

→ pick the parameter combination which leads to best separation index

→ for each C train a classifier with the training data

→ pick the classifier with highest validation accuracy

The C which results in this classifier and the kernel parameter selected by the separation index value form the parameter combination for the classification problem.

This classifier is used to find the testing accuracy of the model.

binary-class classification problems

- Random subsets of Adult database: to predict whether income exceeds \$50k per yr based on census data. The data set has 14 features, among which 6 are continuous and 8 are categorical. In this data set, continuous features are discretized into quantiles, and each quantile is represented by a binary feature. Also, a categorical feature with m categories is converted to m binary features.
- Random subsets of Web database: to classify whether a web page belongs to a category or not. Each input was 300 sparse binary keyword attributes extracted from each web page.

multi-class classification problems

- Modified DNA data set for recognizing splice junctions (points on a DNA sequence at which 'superfluous' DNA is removed during the process of protein creation in higher organisms). The problem posed in this dataset is to recognize, given a sequence of DNA, the boundaries between exons (the parts of the DNA sequence retained after splicing) and introns (the parts of the DNA sequence that are spliced out)
- Satimage database : consists of the multi-spectral values of pixels in 3x3 neighbourhoods in a satellite image, and the classification associated with the central pixel in each neighbourhood. The aim is to predict this classification, given the multi-spectral values. In the sample database, the class of a pixel is coded as a number.
- Letter database: to identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet.

Specifications of the data sets

| Data Set | # of classes | # of features | # training data |
|----------|--------------|---------------|-----------------|
| a1a | 2 | 123 | 1605 |
| a2a | 2 | 123 | 2265 |
| a3a | 2 | 123 | 3185 |
| a4a | 2 | 123 | 4781 |
| a5a | 2 | 123 | 6414 |
| w1a | 2 | 300 | 2477 |
| w2a | 2 | 300 | 3407 |
| w3a | 2 | 300 | 4912 |
| w4a | 2 | 300 | 7366 |
| w5a | 2 | 300 | 9888 |
| DNA | 3 | 180 | 1400 |
| letter | 26 | 16 | 10500 |
| satimage | 6 | 36 | 3104 |

Results: Gaussian kernel for binary-class a1a, subset of Adult database ($\gamma \in \{2^{-20}, \dots, 2^{20}\}$)

Using fivefold cross validation: a speed up of 40.8 is observed

| Method Used | Chosen (C, γ) | accuracy(%) | Training time (s) |
|-----------------|------------------------|-------------|-------------------|
| Grid Search | $(2^2, 2^{-5})$ | 84.40 | 3315.3 |
| Proposed method | $(2^2, 2^{-4})$ | 83.89 | 81.3 |

Using onefold cross validation: a speed up of 19 is observed

| Method Used | Chosen (C, γ) | accuracy(%) | Training time (s) |
|-----------------|------------------------|-------------|-------------------|
| Grid Search | $(2^7, 2^{-11})$ | 84.33 | 660.6 |
| Proposed method | $(2^1, 2^{-4})$ | 83.73 | 34.8 |

Results: Gaussian kernel for binary-class w1a, subset of Web database ($\gamma \in \{2^{-20}, \dots, 2^{20}\}$)

Using fivefold cross validation: a speed up of 41.8 is observed

| Method Used | Chosen (C, γ) | accuracy(%) | Training time (s) |
|-----------------|------------------------|-------------|-------------------|
| Grid Search | $(2^4, 2^{-4})$ | 97.83 | 6458.9 |
| Proposed method | $(2^4, 2^{-4})$ | 97.83 | 154.6 |

Using onefold cross validation: a speed up of 16.3 is observed

| Method Used | Chosen (C, γ) | accuracy(%) | Training time (s) |
|-----------------|------------------------|-------------|-------------------|
| Grid Search | $(2^4, 2^{-4})$ | 97.83 | 1274.2 |
| Proposed method | $(2^4, 2^{-4})$ | 97.83 | 78.0 |

Results: Gaussian kernel for multi-class database

$$(\gamma \in \{2^{-20}, \dots, 2^{20}\})$$

Using fivefold cross validation for DNA database: a speed up of 14.2 is observed

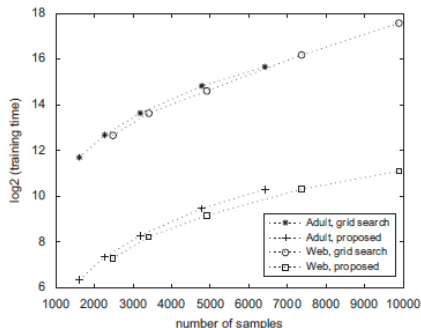
| Method Used | Chosen (C, γ) | accuracy(%) | Training time (s) |
|-----------------|----------------------|-------------|-------------------|
| Grid Search | $(2^3, 2^{-6})$ | 94.69 | 4096.3 |
| Proposed method | $(2^3, 2^{-6})$ | 94.69 | 288.0 |

Using fivefold cross validation for letter database: a speed up of 51.1 is observed

| Method Used | Chosen (C, γ) | accuracy(%) | Training time (s) |
|-----------------|----------------------|-------------|-------------------|
| Grid Search | $(2^3, 2^3)$ | 97.00 | 327668.6 |
| Proposed method | $(2^6, 2^{-1})$ | 96.19 | 6406.5 |

Training time vs training sample size

Training times of the grid search method and the proposed method grow in a similar way, while the grid search method costs more time.



- Both methods can decrease training time with a sub-sampling strategy.

Using Polynomial kernel with one-against-one strategy for multi-class databases

DNA database with a speed-up of 5.7

| Method Used | Chosen (C, γ, d) | accuracy(%) | Training time (s) |
|-----------------|-------------------------|-------------|-------------------|
| Grid Search | $(2^{-7}, 2^2, 2^1)$ | 94.35 | 728.4 |
| Proposed method | $(2^{-7}, 2^3, 2^2)$ | 92.41 | 128.6 |

letter database with a speed-up of 10.7

| Method Used | Chosen (C, γ, d) | accuracy(%) | Training time (s) |
|-----------------|-------------------------|-------------|-------------------|
| Grid Search | $(2^{-2}, 2^0, 2^1)$ | 95.32 | 10070.3 |
| Proposed method | $(2^{-5}, 2^3, 2^2)$ | 94.82 | 939.7 |

satimage database with a speed-up of 4

| Method Used | Chosen (C, γ, d) | accuracy(%) | Training time (s) |
|--------------------|---|--------------------|--------------------------|
| Grid Search | $(2^{-6}, 2^2, 2^2)$ | 88.9 | 452.4 |
| Proposed method | $(2^{-5}, 2^3, 2^2)$ | 88.65 | 114.0 |

- for kernel parameters: $r \in \{0, 2^0, \dots, 2^3\}$, $d \in \{2^0, 2^1, 2^2\}$, i.e. using 15 parameter combinations, a 15 time speed-up is expected.

Reduced parameter search space

If the parameter search space is reduced, the speed up of the proposed method decreases.

Using the gaussian kernel with $\gamma \in \{2^{-5}, 2^{-4}, \dots, 2^5\}$, i.e. 10 available parameter choices, the expected time speed-up is about 10 times, as compared to the 40 times for the wider parameter search space before.

Using both LIBSVM and another SVM training tool SVMtorch, the speed up times are :

| Database | LIBSVM speed up | SVMtorch speed up |
|----------------------------|-----------------|-------------------|
| a1a | 15.4 | 13.7 |
| a2a | 15.0 | 13.5 |
| a3a | 15.6 | 13.7 |
| a4a | 15.2 | 15.2 |
| a5a | 15.5 | 21.9 |
| w1a | 23.5 | 23.1 |
| w2a | 24.8 | 24.6 |
| w3a | 26.8 | 26.2 |
| w4a | 35.7 | 46.7 |
| w5a | 52.4 | 55.0 |
| DNA (one-against-one) | 9.2 | 6.2 |
| DNA (one-against-rest) | 8.5 | 5.5 |
| letter(one-against-one) | 11.4 | 10.3 |
| letter(one-against-rest) | 4.7 | 2.9 |
| setimage(one-against-one) | 6.3 | 6.3 |
| setimage(one-against-rest) | 3.8 | 3.1 |

CONCLUSIONS

It is proposed to calculate the inter-cluster distance in the associated hilbert space to help determine the kernel parameters for training the SVM models

- Results in SVM models that perform as good as the models chosen by the grid search method in testing accuracy.
- The proposed index can be calculated much faster than training SVMs with non-deterministic iterations.
- Time complexity of calculating indexes can be further reduced by the sub-sampling strategy.
- Method is suitable for a larger kernel search space, such as a finer grid for exhaustive searching for better kernel parameters

References

References

- [1] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.
- [2] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, *IEEE Trans. Neural Networks* 13 (2) (2002) 415–425.
- [3] N. Ancona, G. Cicirelli, E. Stella, A. Distanti, Object detection in images: runtime complexity and parameter selection of support vector machines, 16th International Conference on Pattern Recognition, vol. 2, 11–15 August, 2002, pp. 426–429.
- [4] S.A. Rojas, D. Fernandez-Reyes, Adapting multiple kernel parameters for support vector machines using genetic algorithms, The 2005 IEEE Congress on Evolutionary Computation, vol. 1, 02–05 September, 2005, pp. 626–631.
- [5] X. Liang, F. Liu, Choosing multiple parameters for SVM based on genetic algorithm, 6th International Conference on Signal Processing, vol. 1, 26–30 August, 2002, pp. 117–119.
- [6] H.-J. Liu, Y.-N. Wang, X.-F. Lu, A method to choose kernel function and its parameters for support vector machines, in: Proceedings of 2005 International Conference on Machine Learning and Cybernetics, vol. 7, 18–21 August, 2005, pp. 4777–4780.
- [7] R. Debnath, H. Takahashi, An efficient method for tuning kernel parameter of the support vector machine, *IEEE International Symposium on Communications and Information Technology*, vol. 2, 26–29 October, 2004, pp. 1023–1028.
- [8] L.-P. Bi, H. Huang, Z.-Y. Zheng, H.-T. Song, New heuristic for determination Gaussian kernel's parameter, in: Proceedings of 2005 International Conference on Machine Learning and Cybernetics, vol. 7, 18–21 August, 2005, pp. 4299–4304.
- [9] S.S. Keerthi, C.-J. Lin, Asymptotic behaviors of support vector machines with Gaussian kernel, *Neural Comput.* 15 (7) (2003) 1667–1689.
- [10] H.-T. Lin, C.-J. Lin, A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods, Technical Report, Department of Computer Science and Information Engineering, National Taiwan University.
- [11] C.-W. Hsu, C.-C. Chang, C.-J. Lin, A practical guide to support vector classification, [Online] Available from World Wide Web: (<http://www.csie.ntu.edu.tw/~cjlin/libsvm>).
- [12] L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, L. Jackel, Y. LeCun, U. Muller, E. Sackinger, P. Simard, V. Vapnik, Comparison of classifier methods: a case study in handwritten digit recognition, in: Proceedings of International Conference on Pattern Recognition, 1994, pp. 77–87.
- [13] U. Kreßel, Pairwise classification and support vector machines, in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), *Advances in Kernel Methods—Support Vector Learning*, MIT Press, Cambridge, MA, 1999.
- [14] J.C. Platt, N. Cristianini, J. Shaw-Taylor, Large margin DAG's for multiclass classification, *Advances in Neural Information Processing Systems*, vol. 12, MIT Press, Cambridge, MA, 1999.
- [15] S. Liu, C.-Y. Jia, H. Ma, A new weighted support vector machine with GA-based parameter selection, in: Proceedings of 2005 International Conference on Machine Learning and Cybernetics, vol. 7, 18–21 August, 2005, pp. 4351–4355.
- [16] A.T. Quang, Q.-L. Zhang, X. Li, Evolving support vector machine parameters, in: Proceedings of 2002 International Conference on Machine Learning and Cybernetics, vol. 4–5 November, 2002, pp. 548–551.
- [17] F. Imbault, K. Lebart, A stochastic optimization approach for parameter tuning of support vector machines, in: Proceedings of 17th International Conference on Pattern Recognition, vol. 4, 23–26 August, 2004, pp. 597–600.
- [18] J.C. Platt, Sequential minimal optimization: a fast algorithm for training support vector machines, Microsoft Research, Technical Report MST-TR-98-14, 1998.
- [19] K.-I. Maruyama, M. Maruyama, H. Miyao, Y. Nakano, A method to make multiple hypotheses with high cumulative recognition rate using SVMs, *Pattern Recognition* 37 (2004) 241–251.
- [20] D. Hush, C. Scovel, Polynomial-time decomposition algorithms for support vector machines, *Mach. Learn.* 51 (2003) 51V71.
- [21] F. Takahashi, S. Abe, Optimizing directed acyclic graph support vector machines, in: Proceedings of the Artificial Neural Networks in Pattern Recognition (ANNPR 2003), September 2003, pp. 166–170.
- [22] T. Phetkaew, B. Kijirikul, W. Rivepiboon, Reordering adaptive directed acyclic graphs: an improved algorithm for multiclass support vector machines, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN 2003), vol. 2, 2003, pp. 1605–1610.
- [23] K.-P. Wu, S.-D. Wang, Choosing the kernel parameters of support vector machines according to the inter-cluster distance, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN 2006), 2006.
- [24] C. Bezdek, N.R. Pal, Some new indexes of cluster validity, *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 28 (3) (1998) 301–315.
- [25] J.C. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), *Advances in Kernel Methods—Support Vector Learning*, MIT Press, Cambridge, MA, 1999.
- [26] D. Michie, D.J. Spiegelhalter, C.C. Taylor, *Machine learning, neural and statistical classification*, 1994. [Online] Available: (<http://ncc.up.pt/pub/statlog/>).
- [27] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, [Online] Available from World Wide Web: (<http://www.csie.ntu.edu.tw/~cjlin/libsvm>).
- [28] J.X. Dong, A. Krzyzak, C.Y. Suen, Fast SVM training algorithm with decomposition on very large datasets, *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (4) (2005) 603–618.
- [29] R. Collobert, S. Bengio, J. Mariethoz, Torch, [Online] Available from World Wide Web: (<http://www.rncb.ch/>).

Appendix

Table: for $\gamma \in \{2^{-20}, 2^{-19}, \dots, 2^{20}\}$ testing accuracy and training times with grid search and proposed indexed method:

Table 2
Testing accuracy (%) and training times (s) with the grid search method and the proposed indexed method

| Data set | Grid search | | | Proposed method | | | |
|----------|------------------------|----------------------|-------------------|------------------------|----------------------|-------------------|------------------|
| | Chosen (C, γ) | Testing accuracy (%) | Training time (s) | Chosen (C, γ) | Testing accuracy (%) | Training time (s) | Speed-up (times) |
| a1a | $(2^2, 2^{-5})$ | 84.40 | 3315.3 | $(2^2, 2^{-4})$ | 83.89 | 81.3 | 40.8 |
| a2a | $(2^5, 2^{-6})$ | 83.83 | 6535.3 | $(2^1, 2^{-4})$ | 84.34 | 161.0 | 40.6 |
| a3a | $(2^5, 2^{-7})$ | 84.51 | 12717.1 | $(2^0, 2^{-4})$ | 84.31 | 308.8 | 41.2 |
| a4a | $(2^5, 2^{-8})$ | 84.62 | 28924.6 | $(2^0, 2^{-4})$ | 84.55 | 704.1 | 41.1 |
| a5a | $(2^5, 2^{-9})$ | 84.45 | 51736.8 | $(2^1, 2^{-4})$ | 84.40 | 1259.0 | 41.1 |
| w1a | $(2^4, 2^{-4})$ | 97.83 | 6458.9 | $(2^4, 2^{-4})$ | 97.83 | 154.6 | 41.8 |
| w2a | $(2^4, 2^{-5})$ | 98.12 | 12627.5 | $(2^4, 2^{-4})$ | 98.05 | 297.3 | 42.5 |
| w3a | $(2^7, 2^{-7})$ | 98.30 | 24874.2 | $(2^3, 2^{-4})$ | 98.32 | 570.0 | 43.6 |
| w4a | $(2^5, 2^{-6})$ | 98.48 | 74027.5 | $(2^4, 2^{-4})$ | 98.38 | 1234.5 | 60.0 |
| w5a | $(2^5, 2^{-6})$ | 98.60 | 194603.0 | $(2^2, 2^{-4})$ | 98.53 | 2179.6 | 89.3 |
| dna | $(2^3, 2^{-6})$ | 94.69 | 4096.3 | $(2^3, 2^{-6})$ | 94.69 | 288.0 | 14.2 |
| letter | $(2^3, 2^2)$ | 97.00 | 327668.6 | $(2^6, 2^{-1})$ | 96.19 | 6406.5 | 51.1 |
| satimage | $(2^4, 2^0)$ | 90.40 | 10039.9 | $(2^7, 2^{-2})$ | 89.10 | 647.2 | 15.5 |

Table: for $\gamma \in \{2^{-20}, 2^{-19}, \dots, 2^{20}\}$ testing accuracy and training times with grid search and proposed indexed method, using one-fold training process:

Table 3
Testing accuracy (%) and training times (s) with the grid search method and the proposed indexed method, using only onefold training process

| Data set | Grid search | | | Proposed method | | | |
|----------|-----------------------|----------------------|-------------------|-----------------------|----------------------|-------------------|------------------|
| | Chosen (C, γ) | Testing accuracy (%) | Training time (s) | Chosen (C, γ) | Testing accuracy (%) | Training time (s) | Speed-up (times) |
| a1a | ($2^7, 2^{-11}$) | 84.33 | 660.6 | ($2^1, 2^{-4}$) | 83.73 | 34.8 | 19.0 |
| a2a | ($2^6, 2^{-3}$) | 84.11 | 1291.3 | ($2^1, 2^{-4}$) | 84.34 | 68.9 | 18.8 |
| a3a | ($2^5, 2^{-7}$) | 84.51 | 2549.4 | ($2^1, 2^{-4}$) | 84.32 | 134.3 | 19.0 |
| a4a | ($2^3, 2^{-9}$) | 84.29 | 5762.2 | ($2^0, 2^{-4}$) | 84.55 | 303.1 | 19.0 |
| a5a | ($2^2, 2^{-7}$) | 84.16 | 10235.9 | ($2^2, 2^{-4}$) | 84.25 | 547.0 | 18.7 |
| w1a | ($2^4, 2^{-4}$) | 97.83 | 1274.2 | ($2^4, 2^{-4}$) | 97.83 | 78.0 | 16.3 |
| w2a | ($2^4, 2^{-4}$) | 98.05 | 2503.5 | ($2^4, 2^{-4}$) | 98.05 | 153.8 | 16.3 |
| w3a | ($2^7, 2^{-7}$) | 98.30 | 4951.1 | ($2^0, 2^{-4}$) | 97.73 | 301.9 | 16.4 |
| w4a | ($2^6, 2^{-7}$) | 98.49 | 14193.2 | ($2^2, 2^{-4}$) | 98.37 | 674.4 | 21.0 |
| w5a | ($2^2, 2^{-4}$) | 98.53 | 40426.3 | ($2^2, 2^{-4}$) | 98.53 | 1192.4 | 33.9 |

Table: testing accuracy and training times with grid search and proposed indexed method, using polynomial kernel, for $r \in \{0, 2^0, \dots, 2^3\}$ and $d \in \{2^0, 2^1, 2^2\}$:

Table 5
Testing accuracy (%) and training time (s) with the polynomial kernel

| Data set | Grid search | | | Proposed method | | | |
|----------|----------------------|----------------------|-------------------|----------------------|----------------------|-------------------|------------------|
| | Chosen (C, r, d) | Testing accuracy (%) | Training time (s) | Chosen (C, r, d) | Testing accuracy (%) | Training time (s) | Speed-up (times) |
| a5a | $(2^{-3}, 2^0, 2^0)$ | 84.48 | 425 756.7 | $(2^{-7}, 2^3, 2^2)$ | 78.82 | 1698.3 | 250.7 |
| w5a | $(2^0, 2^0, 2^0)$ | 98.51 | 4954.3 | $(2^{-7}, 2^3, 2^2)$ | 97.8 | 473.1 | 10.5 |
| dna | $(2^{-7}, 2^2, 2^1)$ | 94.35 | 728.4 | $(2^{-7}, 2^3, 2^2)$ | 92.41 | 128.6 | 5.7 |
| letter | $(2^{-2}, 2^0, 2^2)$ | 95.32 | 10 070.3 | $(2^{-5}, 2^3, 2^2)$ | 94.82 | 939.7 | 10.7 |
| satimage | $(2^{-6}, 2^2, 2^2)$ | 88.9 | 452.4 | $(2^{-5}, 2^3, 2^2)$ | 88.65 | 114.0 | 4.0 |

Table: Comparing the training time of different training tools in a reduced parameter search space, for $\gamma \in \{2^{-5}, 2^{-4}, \dots, 2^5\}$

Table 6
Comparing the training time (s) of different training tools in a reduced parameter search space

| Data set | LIBSVM | | | SVMtorch | | |
|-----------------------------|---------------|-----------------|------------------|---------------|-----------------|------------------|
| | Grid search | Proposed method | | Grid search | Proposed method | |
| | Training time | Training time | Speed-up (times) | Training time | Training time | Speed-up (times) |
| a1a | 750.5 | 48.7 | 15.4 | 924.9 | 67.5 | 13.7 |
| a2a | 1462.5 | 97.4 | 15.0 | 1694.8 | 125.9 | 13.5 |
| a3a | 2841.1 | 182.7 | 15.6 | 3067.5 | 224.2 | 13.7 |
| a4a | 6410.0 | 421.9 | 15.2 | 7241.5 | 476.7 | 15.2 |
| a5a | 11473.6 | 741.4 | 15.5 | 17963.4 | 821.1 | 21.9 |
| w1a | 1688.9 | 71.9 | 23.5 | 2441.7 | 105.7 | 23.1 |
| w2a | 3312.3 | 133.3 | 24.8 | 4477.9 | 182.0 | 24.6 |
| w3a | 6472.3 | 241.7 | 26.8 | 8300.0 | 316.2 | 26.2 |
| w4a | 17704.9 | 496.5 | 35.7 | 28457.6 | 609.4 | 46.7 |
| w5a | 44617.8 | 851.4 | 52.4 | 55850.6 | 1015.6 | 55.0 |
| dna (one-against-one) | 1154.7 | 125.2 | 9.2 | 615.4 | 98.6 | 6.2 |
| dna (one-against-rest) | 2028.9 | 239.4 | 8.5 | 1049.7 | 192.1 | 5.5 |
| letter (one-against-one) | 30446.6 | 2682.7 | 11.3 | 36402.5 | 3536.1 | 10.3 |
| letter (one-against-rest) | 98956.6 | 20925.9 | 4.7 | 61960.8 | 21215.7 | 2.9 |
| satimage (one-against-one) | 1192.4 | 187.9 | 6.3 | 1480.5 | 236.2 | 6.3 |
| satimage (one-against-rest) | 2478.7 | 645.3 | 3.8 | 2113.4 | 672.1 | 3.1 |