TABLE II. DATASETS OF DENSE FEATURES. WE APPLY FIVE-FOLD CROSS VALIDATION ON THE DATASETS MARKED BY *, WHILE USING GIVEN TRAINING/TEST SPLITS ON THE OTHER DATASETS.

| Dataset | #class | #feature | #training sample | #test sample |
|---|---|---|---|---|
| SENSIT-VEHICLE | 3 | 100 | 78,823 | 19,705 |
| SEMEION* | 10 | 256 | 1,275 | 318 |
| ISOLET | 26 | 617 | 6,238 | 1,559 |
| MNIST | 10 | 784 | 60,000 | 10,000 |
| P53* | 2 | 5,408 | 13,274 | 3,318 |

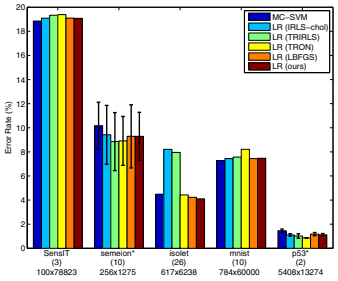
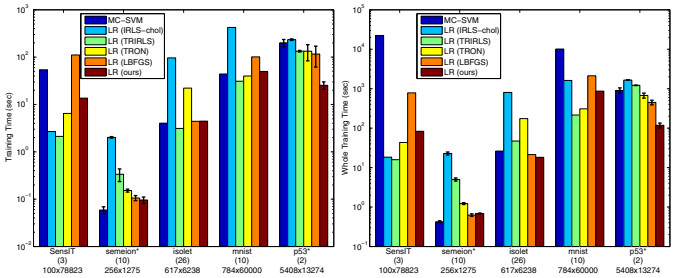



Fig. 1. Error rates on linear classification for dense features. The numbers of classes are indicated in parentheses and the sizes of $\boldsymbol{X}$ (#feature×#sample) are shown in the bottom.




(a) On final learning (b) On whole learning

Fig. 2. Computation times (*log-scale*) on linear classification for dense features. The computation time for learning final classifier is shown in (a), while that for whole training including 3-CV to determine $\lambda$ is in (b).

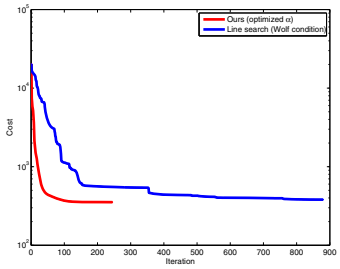



Fig. 3. Comparison to the method using an exhaustive line search. The plot shows the objective cost values through iterations on ISOLET.

## A. Linear classification

As a preliminary experiment to the subsequent kernel-based methods, we applied linear classification methods.

For comparison, we applied multi-class support vector machine (MC-SVM) [7] and for LR, four types of optimization methods other than the proposed method in Section III:

- IRLS with Cholesky decomposition (IRLS-chol) [17]
- IRLS with CG (TRIRLS) [18]
- IRLS with trust region newton method (TRON) [20]
- limited memory BFGS method (LBFGS) [13] and [21].

All of these methods introduce regularization with respect to classifier norm in a similar form to (3), of which the regularization parameter is determined by three-fold cross validation on training samples ($\lambda \in \{1, 10^{-2}, 10^{-4}\}$). We implemented all the methods by using MATLAB with C-mex on Xeon 3GHz (12 threading) PC; we used LIBLINEAR [32] for MC-SVM and TRON, and the code[1] provided by Liu and Nocedal [22] for LBFGS.

We first used the datasets[2] of the dense feature vectors, the details of which are shown in Table II. For evaluation, we used the given training/test splits on some datasets and applied five-fold cross validation on the others. The classification performances (error rates) and the computation times for training the classifier are shown in Fig. 1 and Fig. 2, respectively. The computation times are measured in two ways; Fig. 2(a) shows the computation time only for learning the final classifier and Fig. 2(b) is for 'whole' training process including both the final learning and three-fold cross validations to determine the regularization parameter. The proposed method is favorably compared to the other methods in terms of error rates and computation time; the method of LR with IRLS-chol which is quite close to the ordinary IRLS requires more training time.

We then investigated the effectiveness of the optimized step size $\alpha$ (Section III-B) which is one of our contributions in this paper. Fig. 3 shows how the proposed optimization method works, compared to that using an exhaustive line search. By employing the optimized step size, the objective cost drastically decreases in the fist few steps and reaches convergence in a smaller number of iterations.

In the same experimental protocol, we also applied the methods to datasets which contain sparse feature vectors. The details of the datasets[3] are shown in Table III. Note that the method of LR with IRLS-chol can not deal with such a huge feature vectors since the Hessian matrix is quite large, making it difficult to solve linear equations by Cholesky decomposition in a realistic time. As shown in Fig. 4 and Fig. 5, the computation times of the methods are all comparable (around 10 seconds) with similar classification accuracies.

Though the performances of the proposed method are favorably compared to the others as a whole, they are different from those of IRLS-based methods (TRIRLS and TRON). The reason is as follows. The objective costs of those methods[4] are shown in Table IV. The proposed method produces lower objective costs than those by TRIRLS, and thus we can say that the IRLS-based method does not fully converge to global minimum. Although the objective cost function is convex, there would exist plateau [38] which stop the optimization in the IRLS-based methods before converging to the global minimum. Thus, from the viewpoint of optimization, the proposed method produces favorable results.

[1]The code is available at http://www.ece.northwestern.edu/~nocedal.

[2]SEMEION, ISOLET and P53 are downloded from UCI-repository http://archive.ics.uci.edu/ml/datasets.html, and SENSIT-VEHICLE [33] and MNIST [34] are from http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

[3]REUTERS21578 (UCI KDD Archive) and TDT2 (Nist Topic Detection and Tracking corpus) are downloaded from http://www.zjucadcg.cn/dengcai/Data/TextData.html, and RCV1 [35], SECTOR [36] and NEWS20 [37] are from http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

[4]We do not show the cost of TRON [20] whose formulation is slightly different as described in Section II-A.

TABLE III. DATASETS OF SPARSE FEATURES. WE APPLY FIVE-FOLD CROSS VALIDATION ON THE DATASETS MARKED BY *, WHILE USING GIVEN TRAINING/TEST SPLITS ON THE OTHER DATASETS.

| Dataset | #class | #feature | #training sample | #non zeros | #test sample |
|---|---|---|---|---|---|
| REUTERS21578 | 51 | 18,933 | 5,926 | 283,531 | 2,334 |
| TDT2* | 77 | 36,771 | 8,140 | 1,056,166 | 2,035 |
| RCV1 | 51 | 47,236 | 15,564 | 1,028,284 | 518,571 |
| SECTOR | 105 | 55,197 | 6,412 | 1,045,412 | 3,207 |
| NEWS20 | 20 | 62,060 | 15,935 | 1,272,568 | 3,993 |

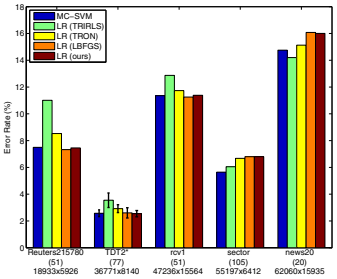
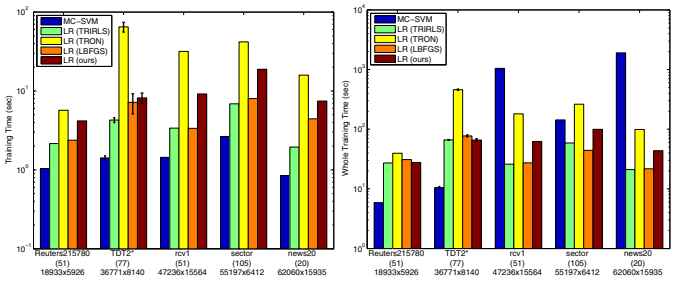



Fig. 4. Error rates on linear classification for sparse features.




(a) On final learning (b) On whole learning

Fig. 5. Computation times on linear classification for sparse features.

TABLE IV. OBJECTIVE COST VALUES OF LR METHODS WITH $\lambda = 10^{-2}$ ON SPARSE DATASETS.

| Dataset | Ours | TRIRLS | LBFGS |
|---|---|---|---|
| REUTERS21578 | **9.98** | 389.26 | 10.32 |
| TDT2 | **12.13** | 387.58 | 13.65 |
| RCV1 | **906.49** | 15687.17 | 969.07 |
| SECTOR | **1102.08** | 29841.19 | 1167.35 |
| NEWS20 | **1949.60** | 7139.07 | 2000.64 |

### B. Kernel-based classification

Next, we conducted the experiments on kernel-based classifications. We applied the proposed kernel logistic regression (KLR) in Section IV and the kernelized methods of the above-mentioned linear classifiers;

- multi-class kernel support vector machine (MC-KSVM) [7]
- KLR using IRLS with CG (TRIRLS) [18]
- KLR using IRLS with trust region newton method (TRON) by [20]
- KLR using limited memory BFGS method (LBFGS) [13], [21].

Note that the KLR methods of TRIRLS, TRON and LBFGS are kernelized in the way described in Section II-D. Table V shows the details of the datasets[5] that we use, and in this experiment, we employed RBF kernel $\mathsf{k}(\boldsymbol{x}, \boldsymbol{\xi}) = \exp(-\frac{\|\boldsymbol{x}-\boldsymbol{\xi}\|^2}{2\sigma^2})$ where $\sigma^2$ is determined as the sample variance. The experimental protocol is the same as in Section VII-A.

As shown in Fig. 6, the classification performances of the proposed method are superior to the other KLR methods and are comparable to MC-KSVM, while the computation times of the proposed method are faster than that of MC-KSVM on most datasets (Fig. 7). As discussed in Section VI, we can employ GPGPU (NVIDIA Tesla C2050) to efficiently compute the matrix multiplications in our method on the datasets except for the huge dataset of SHUTTLE, and the computation time is significantly reduced as shown in Fig. 7.

While the proposed method optimizes the classifier in RKHS, the optimization in the other KLR methods is performed in the subspace spanned by the sample kernel functions (Section IV), possibly causing numerically unfavorable issues such as plateau [38], and the optimizations would terminate before fully converging to the global minimum. The objective costs shown in Table VI illustrates it; the proposed method provides lower costs than those of the other KLR methods. In addition, the obtained classifiers, i.e., coefficients $\boldsymbol{W}$ for samples, are shown in Fig. 8. The proposed method produces near sparse weights compared to those of the other methods and contribute to improve the performance similarly to MC-KSVM, even though any constraints to enhance sparseness are not imposed in the proposed method.

### C. Multiple-kernel learning

Finally, we conducted the experiment on multiple-kernel learning. We applied the proposed multiple-kernel logistic regression (MKLR) described in Section V and simpleMKL [29] for comparison. For simpleMKL, we used the code[6] provided by the author with LIBSVM [41]. The details of the datasets[7] are shown in Table VII; for multi-class classification, in the dataset of PASCAL-VOC2007, we removed the samples to which multiple labels are assigned. In the datasets of PSORT-, NONPLANT and PASCAL-VOC2007, we used the precomputed kernel matrices provided in the authors' web sites. The dataset of PEN-DIGITS contains four types of feature vectors and correspondingly we constructed four types of RBF kernel in the same way as in Section VII-B.

The classification performances are shown in Fig. 9. As a reference, we also show the performances of KLR with the (single) averaged kernel matrix and the (single) best kernel matrix which produces the best performance among the multiple kernel matrices. The MKL methods produce superior performances compared to those of KLR with single kernel, and the proposed method is comparable to simpleMKL. The obtained kernel weights are also shown in Fig. 10. The weights by the proposed method are sparse similarly to those by simpleMKL, due to the formulation based on the combined RKHS $\bar{\mathcal{H}}$ in (14) and its efficient optimization using non-linear CG.

As shown in Fig. 11, the computation time of the proposed method is significantly ($10^2 \sim 10^4$ times) faster than

[5] USPS [39], LETTER (Statlog), PROTEIN [40] and SHUTTLE (Statlog) are downloaded from http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/, and POKER is from UCI repository http://archive.ics.uci.edu/ml/datasets/.

[6] The code is available at http://asi.insa-rouen.fr/enseignants/~arakotom/code/mklindex.html

[7] PASCAL-VOC2007 [42] is downloaded from http://lear.inrialpes.fr/people/guillaumin/data.php, PEN-DIGITS [43] is from http://mkl.ucsd.edu/dataset/pendigits, and PSORT-, NONPLANT [44] are from http://www.fml.tuebingen.mpg.de/raetsch/suppl/protsubloc.

TABLE V. DATASETS FOR KERNEL-BASED CLASSIFICATION.

| Dataset | #class | #feature | #training sample | #test sample |
|---|---|---|---|---|
| USPS | 10 | 256 | 7,291 | 2,007 |
| LETTER | 26 | 16 | 15,000 | 5,000 |
| PROTEIN | 3 | 357 | 17,766 | 6,621 |
| POKER | 10 | 10 | 25,010 | 1,000,000 |
| SHUTTLE | 7 | 9 | 43,500 | 14,500 |

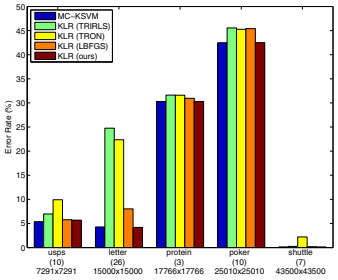
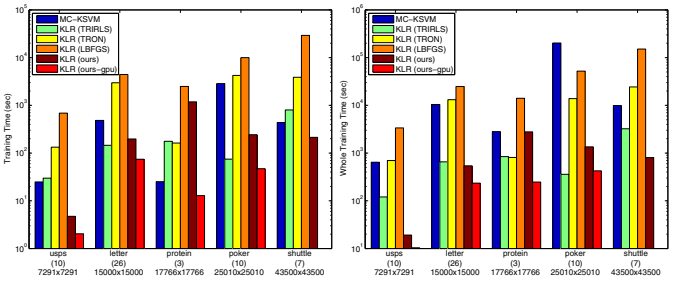



Fig. 6. Error rates on kernel-based classification.

that of simpleMKL. Thus, as is the case with kernel-based classification (Section VII-B), we can say that the proposed method produces comparable performances to simpleMKL with a significantly faster training time.

## VIII. CONCLUDING REMARKS

In this paper, we have proposed an efficient optimization method using non-linear conjugate gradient (CG) descent for logistic regression. The proposed method efficiently minimizes the cost through CG iterations by using the optimized step size without an exhaustive line search. On the basis of the non-linear CG based optimization scheme, a novel optimization method for kernel logistic regression (KLR) is also proposed. Unlike the ordinary KLR methods, the proposed method naturally formulates the classifier as the linear combination of sample kernel functions and directly optimizes the kernel-based classifier in the reproducing kernel Hilbert space, not the linear coefficients for the samples. Thus, the optimization effectively performs while possibly avoiding the numerical issues such as plateau. We have further developed the KLR using single kernel to multiple-kernel LR (MKLR). The proposed MKLR, which is also optimized in the scheme of non-linear CG, produces the kernel-based classifier with optimized weights for multiple kernels. In the experiments on various multi-class classification tasks, the proposed methods produced favorable results in terms of classification performance and computation time, compared to the other methods.




(a) On final learning (b) On whole learning

Fig. 7. Computation times on kernel-based classification.

TABLE VI. OBJECTIVE COST VALUES OF KLR METHODS WITH $\lambda = 10^{-2}$ ON KERNEL DATASETS.

| Dataset | Ours | TRIRLS | LBFGS |
|---|---|---|---|
| USPS | **446.37** | 914.88 | 501.15 |
| LETTER | **4746.13** | 12476.41 | 5789.30 |
| PROTEIN | **5866.16** | 12576.97 | 10650.96 |
| POKER | **22186.19** | 30168.74 | 23345.94 |
| SHUTTLE | **759.99** | 1100.07 | 811.91 |

## REFERENCES

[1] G. Wang, D. Hoiem, and D. Forsyth, "Building text features for object image classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1367–1374.

[2] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert, "An empirical study of context in object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1271–1278.

[3] I. Laptev, M. Marszaek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[4] A. Genkin, D. D. Lewis, and D. Madigan, "Large-scale bayesian logistic regression for text categorization," *Technometrics*, vol. 49, no. 3, pp. 291–304, 2007.

[5] P. J. Bartlett, B. Schölkopf, D. Schuurmans, and A. J. Smola, Eds., *Advances in Large-Margin Classifiers*. MIT Press, 2000.

[6] V. Vapnik, *Statistical Learning Theory*. Wiley, 1998.

[7] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research*, vol. 2, pp. 265–292, 2001.

[8] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999, pp. 185–208.

[9] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 1999.

[10] W. W. Hager and H. Zhang, "A survey of nonlinear conjugate gradient methods," *Pacific Journal of Optimization*, vol. 2, pp. 35–58, 2006.

[11] G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[12] K. Watanabe, T. Kobayashi, and N. Otsu, "Efficient optimization of logistic regression by direct cg method," in *International Conference on Machine Learning and Applications*, 2011.

[13] R. Malouf, "A comparison of algorithms for maximum entropy parameter estimation," in *The Sixth Conference on Natural Language Learning*, 2002, pp. 49–55.

[14] C. Sutton and A. McCallum, *An introduction to conditional random fields for relational learning*, L. Getoor and B. Taskar, Eds. MIT Press, 2006.

[15] T. Minka, "A comparison of numerical optimizers for logistic regression," Carnegie Mellon University, Technical report, 2003.

[16] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.

[17] P. Komarek and A. Moore, "Fast robust logistic regression for large sparse datasets with binary outputs," in *The 9th International Workshop on Artificial Intelligence and Statistics*, 2003, pp. 3–6.

[18] ——, "Making logistic regression a core data mining tool," in *International Conference on Data Mining*, 2005, pp. 685–688.

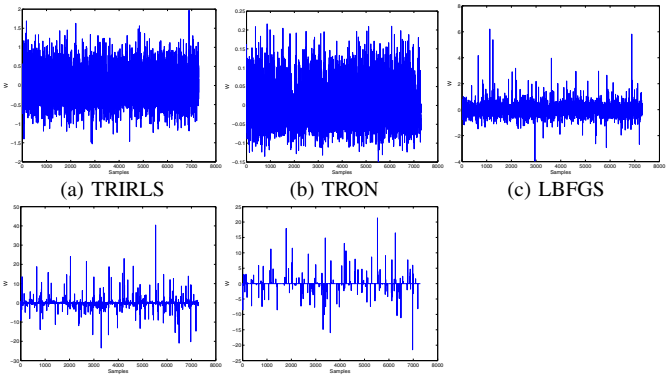
[19] M. R. Hestenes and E. L. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, 1952.

[20] C.-J. Lin, R. Weng, and S. Keerthi, "Trust region newton methods for large-scale logistic regression," in *International Conference on Machine Learning*, 2007, pp. 561–568.

[21] H. Daumé III, "Notes on CG and LM-BFGS optimization of logistic regression," Technical report, 2004. [Online]. Available: http://www.umiacs.umd.edu/~hal/docs/daume04cg-bfgs.pdf

[22] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical Programming*, vol. 45, pp. 503–528, 1989.

[23] S. D. Pietra, V. D. Pietra, and J. Lafferty, "Inducing features of



(a) TRIRLS (b) TRON (c) LBFGS

(d) ours (e) MC-KSVM

Fig. 8. Classifiers (coefficients $\boldsymbol{w}_1$ across samples) of class 1 on USPS.

TABLE VII. DATASETS FOR MULTIPLE-KERNEL LEARNING. WE APPLY FIVE-FOLD CROSS VALIDATION ON THE DATASETS MARKED BY *, WHILE USING GIVEN TRAINING/TEST SPLITS ON THE OTHER DATASETS.

| Dataset | #class | #kernel | #training sample | #test sample |
|---|---|---|---|---|
| PSORT-* | 5 | 69 | 1,155 | 289 |
| NONPLANT* | 3 | 69 | 2,186 | 546 |
| PASCAL-VOC2007 | 20 | 15 | 2,954 | 3,192 |
| PEN-DIGITS | 10 | 4 | 7,494 | 3,498 |

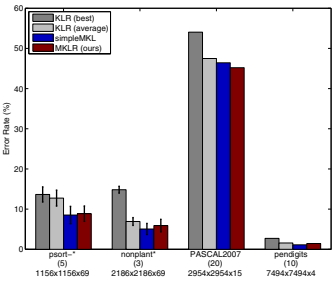
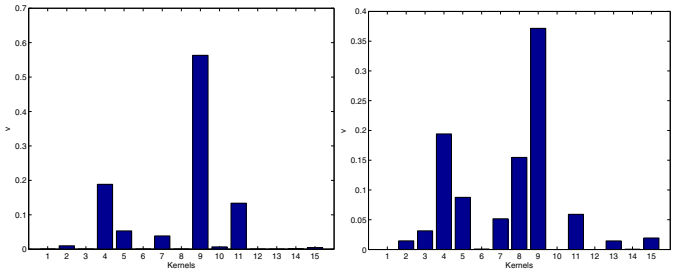
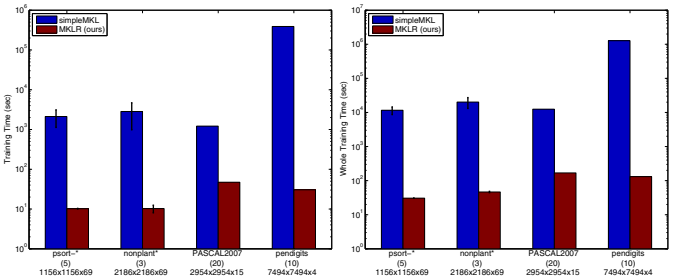



Fig. 9. Error rates and computation times on multiple-kernel learning.




(a) simpleMKL [29] (b) MKLR (ours)

Fig. 10. The obtained kernel weights $\boldsymbol{v}$ on PASCAL-VOC2007.




(a) On final learning (b) On whole learning

Fig. 11. Computation times on multiple-kernel learning.

random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 380–393, 1997.

[24] J. Zhu and T. Hastie, "Kernal logistic regression and the import vector machine," *Journal of Computational and Graphical Statistics*, vol. 14, no. 1, pp. 185–205, 2005.

[25] G. Wahba, C. Gu, Y. Wang, and R. Chappell, "Soft classification, a.k.a. risk estimation, via penalized log likelihood and smoothing spline analysis of variance," in *The Mathematics of Generalization*, D. Wolpert, Ed. Reading, MA, USA: Addison-Wesley, 1995, pp. 329–360.

[26] T. Hastie and R. Tibshirani, *Generalized Additive Models*. Chapman and Hall, 1990.

[27] B. Schölkopf and A. Smola, *Learning with Kernels*. MIT Press, 2001.

[28] Y. Dai and L. Liao, "New conjugacy conditions and related nonlinear conjugate gradient methods," *Applied Mathmatics and Optimization*, vol. 43, pp. 87–101, 2001.

[29] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "Simplemkl," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.

[30] F. Bach, "Consistency of the group lasso and multiple kernel learning," *Journal of Machine Learning Research*, vol. 9, pp. 1179–1225, 2008.

[31] J. Bonnans, J. Gilbert, C. Lemaréchal, and C. Sagastizábal, *Numerical Optimization: Theoritical and Practical Aspects*. Springer, 2006.

[32] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, "Liblinear: A library for large linear classificatio," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008, Software available at http://www.csie.ntu.edu.tw/~cjlin/liblinear.

[33] M. Duarte and Y. H. Hu, "Vehicle classification in distributed sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 826–838, 2004.

[34] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[35] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "Rcv1: A new benchmark collection for text categorization research," *Journal of Machine Learning Research*, vol. 5, pp. 361–397, 2004.

[36] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *AAAI'98 Workshop on Learning for Text categorization*, 1998.

[37] K. Lang, "Newsweeder: Learning to filter netnews," in *International Conference on Machine Learning*, 1995, pp. 331–339.

[38] K. Fukumizu and S. Amari, "Local minima and plateaus in hierarchical structures of multilayer perceptrons," *Neural Networks*, vol. 13, no. 3, pp. 317–327, 2000.

[39] J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550–554, 1994.

[40] J.-Y. Wang, "Application of support vector machines in bioinformatics," Master's thesis, Department of Computer Science and Information Engineering, National Taiwan University, 2002.

[41] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[42] M. Guillaumin, J. Verbeek, and C. Schmid, "Multimodal semi-supervised learning for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 902–909.

[43] F. Alimoglu and E. Alpaydin, "Combining multiple representations and classifiers for pen-based handwritten digit recognition," in *International Conference on Document Analysis and Recognition*, 1997, pp. 637–640.

[44] A. Zien and C. S. Ong, "An automated combination of kernels for predicting protein subcellular localization," in *Proceedings of the 8th Workshop on Algorithms in Bioinformatics*, 2008, pp. 179–186.