Appendix of "Linear and Kernel Classification: When to Use Which?"

Hsin-Yuan Huang* Chih-Jen Lin†

In the following content, we assume that the loss function $\xi(\boldsymbol{w}; \boldsymbol{x}, y)$ can be represented as a function of $\boldsymbol{w}^T\boldsymbol{x}$ when y is fixed. That is,

$$\xi(\boldsymbol{w}; \boldsymbol{x}, y) = \bar{\xi}(\boldsymbol{w}^T\boldsymbol{x}; y).$$

Note that the three loss functions in Section 2 satisfy the above property.

I Parameter r Is not Needed for Degree-2 Expansions

From the definition of ϕ, we have

$$\phi_{\gamma,r}(\boldsymbol{x}) = \gamma\phi_{1,\frac{r}{\gamma}}(\boldsymbol{x}).$$

By Theorem 4.1, the optimal solution for (C, γ, r) and $(\gamma^2 C, 1, r/\gamma)$ are $\boldsymbol{w}/\gamma$ and $\boldsymbol{w}$, respectively. Therefore the two decision functions are the same:

$$(\frac{\boldsymbol{w}}{\gamma})^T\phi_{\gamma,r}(\boldsymbol{x}) = \boldsymbol{w}^T\phi_{1,\frac{r}{\gamma}}(\boldsymbol{x}).$$

Thus, if C is chosen properly, γ is not needed.

II Proof of Theorem 3.1

We prove the result by contradiction. Assume there exists $\boldsymbol{w}'$ such that

$$\sum_{i=1}^{l} \xi(\boldsymbol{w}'; \bar{\boldsymbol{x}}_i, y_i) < \sum_{i=1}^{l} \xi(D^{-1}\boldsymbol{w}^*; \bar{\boldsymbol{x}}_i, y_i).$$

Then from (3.8) we have

$$\sum_{i=1}^{l} \bar{\xi}((D\boldsymbol{w}')^T\boldsymbol{x}_i; y_i) < \sum_{i=1}^{l} \bar{\xi}((DD^{-1}\boldsymbol{w}^*)^T\boldsymbol{x}_i; y_i).$$

Thus,

$$\sum_{i=1}^{l} \xi(D\boldsymbol{w}'; \boldsymbol{x}_i, y_i) < \sum_{i=1}^{l} \xi(\boldsymbol{w}^*; \boldsymbol{x}_i, y_i),$$

which contradicts the assumption that $\boldsymbol{w}^*$ is an optimal solution for (3.9). Thus $D^{-1}\boldsymbol{w}^*$ is an optimal solution for (3.10).

*Department of Computer Science, National Taiwan University. momohuang@gmail.com

†Department of Computer Science, National Taiwan University. cjlin@csie.ntu.edu.tw

III Proof of Theorem 4.1

It can be clearly seen that the following two optimization problems are equivalent

$$\min_{\boldsymbol{w}} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{l} \xi(\boldsymbol{w}; y_i, \boldsymbol{x}_i) \tag{III.1}$$

and

$$\min_{\boldsymbol{w}} \frac{1}{2}(\frac{\boldsymbol{w}}{\Delta})^T(\frac{\boldsymbol{w}}{\Delta}) + \frac{C}{\Delta^2}\sum_{i=1}^{l} \xi(\frac{\boldsymbol{w}}{\Delta}; y_i, \Delta\boldsymbol{x}_i). \tag{III.2}$$

Problem (III.2) can be written as

$$\min_{\bar{\boldsymbol{w}}} \frac{1}{2}\bar{\boldsymbol{w}}^T\bar{\boldsymbol{w}} + \frac{C}{\Delta^2}\sum_{i=1}^{l} \xi(\bar{\boldsymbol{w}}; y_i, \Delta\boldsymbol{x}_i), \tag{III.3}$$

which trains $\Delta\boldsymbol{x}_i, \forall i$ under the regularization parameter C/Δ^2. Therefore, if $\boldsymbol{w}$ is optimal for (III.1), then $\boldsymbol{w}/\Delta$ is optimal for (III.3).

IV Details of Experimental Settings

All experiments run on a 4GHz Intel Core i7 (I7-4790K) with 16G RAM. We transform some problems from multi-class to binary. For news20, we consider classes $2, \ldots, 7$ and $12, \ldots, 15$ as positive, while others as negative. For the digit-recognition problem mnistOvE, we consider odd numbers as positive and even numbers as negative. For poker, class 0 forms the positive class, and the rest forms the negative class.

For the reference linear classifier and all linear classifiers built within our kernel-check methods, we consider the primal-based Newton method for L2-loss SVM in LIBLINEAR. For the reference Gaussian kernel model, we use LIBSVM, which implements L1-loss SVM. We do not consider L1-loss SVM in LIBLINEAR because the automatic parameter-selection procedure is available only for logistic and L2 hinge losses. We take this chance to see if our kernel-check methods are sensitive to the choices of loss functions. All default settings of LIBSVM and LIBLINEAR are used except that the stopping tolerance of LIBLINEAR is reduced to 10^{-4}.

We conduct feature-wise scaling such that each instance $\boldsymbol{x}_i$ becomes $D\boldsymbol{x}_i$, where D is a diagonal matrix

with

$$D_{jj} = \frac{1}{\max_t |(\boldsymbol{x}_t)_j|}, j = 1, \ldots, n.$$

The resulting feature values are in $[-1, 1]$. Although the ranges of features may be slightly different, this scaling ensures that the sparsity of the data is kept.

V Experiments on Data Scaling

We compare the performance (CV accuracy) of original, feature-wisely scaled and instance-wisely scaled data under different C in Figure (I). In addition to data sets considered in the paper, we include more sets from LIBSVM data sets without modifications.[1] For a few problems, curves with/without scaling are the same. Problem a9a has 0/1 values, so the set remains the same after feature-wise scaling. For gisette, the original data is already feature-wisely scaled. For rcv1, real-sim and webspam, the original data is already instance-wisely scaled. We consider the primal-based Newton method in LIBLINEAR for L2-loss SVM. The stopping tolerance is set to be 10^{-4}, except 10^{-6} for breast cancer and 10^{-2} for yahoo-japan.

From Figure (I), we make the following observations.

1. The best CV accuracy is about the same for all three settings, except that instance-wise normalization gives significantly lower CV accuracy for australian.

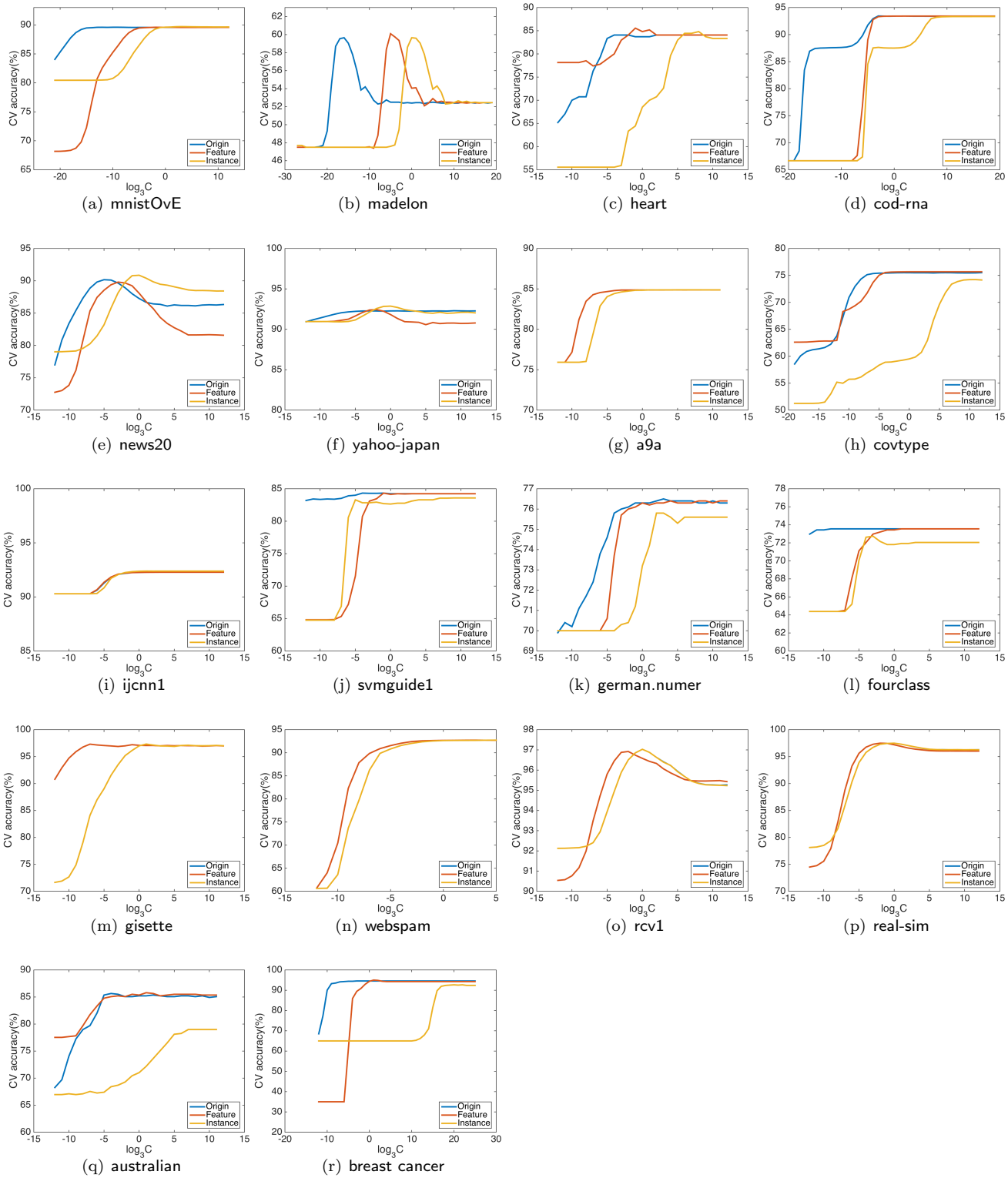
2. The curve for instance-wise scaling is all to the right of the original data, and for many data sets the shape of the curve is also very similar.

The above observations are consistent with our analysis in Section 4.

VI MultiLinear SVM using Different Settings

Experiments of MultiLinear SVM using different clustering algorithms and numbers of clusters are given at Figures (II) and (III) for all 15 data sets.

[1] One exception is yahoo-japan, which is not publicly available.



(a) mnistOvE (b) madelon (c) heart (d) cod-rna

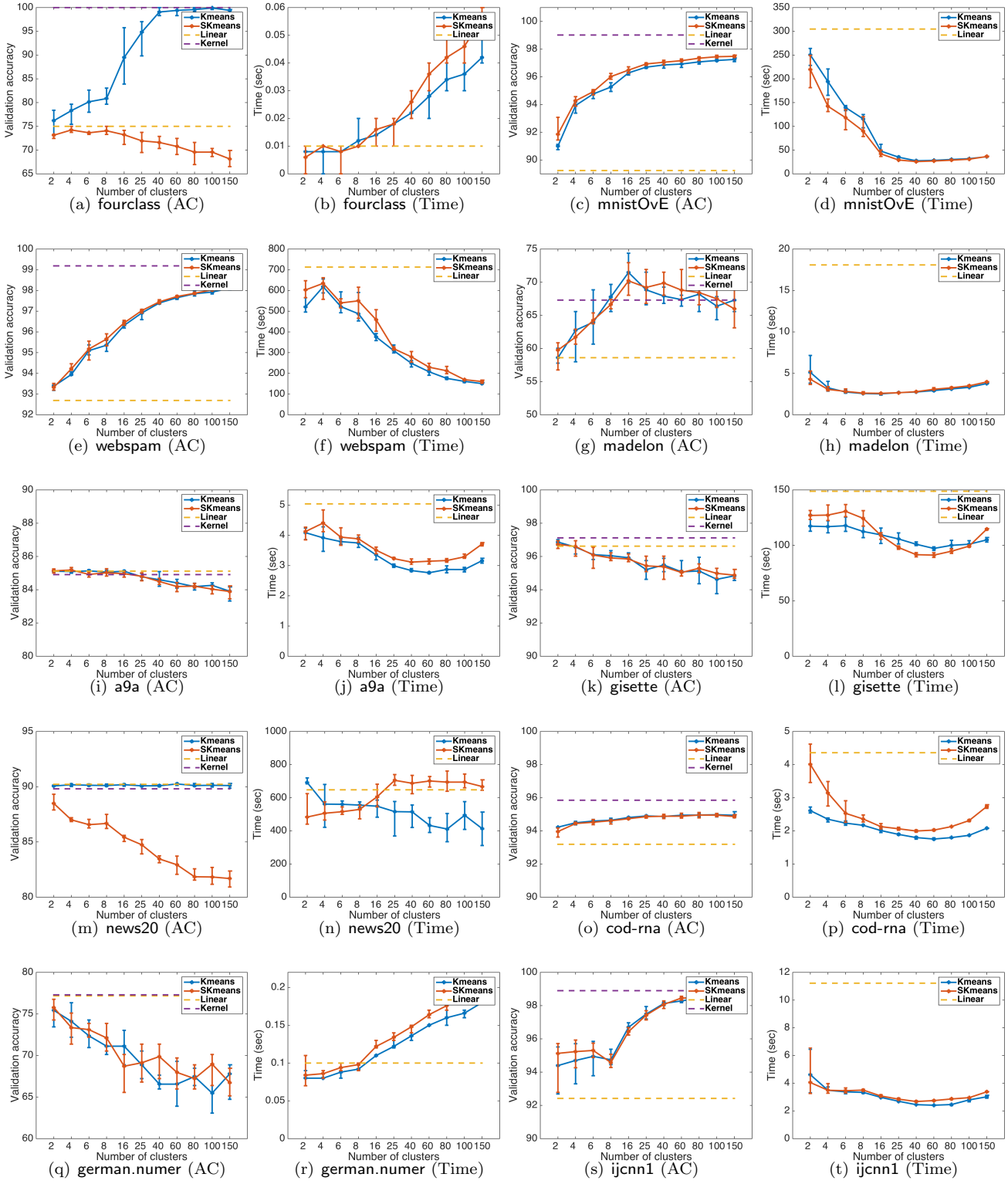
(e) news20 (f) yahoo-japan (g) a9a (h) covtype

(i) ijcnn1 (j) svmguide1 (k) german.numer (l) fourclass

(m) gisette (n) webspam (o) rcv1 (p) real-sim

(q) australian (r) breast cancer

Figure (I): Performance of linear classifiers using different scaling methods.



(a) fourclass (AC) (b) fourclass (Time) (c) mnistOvE (AC) (d) mnistOvE (Time)

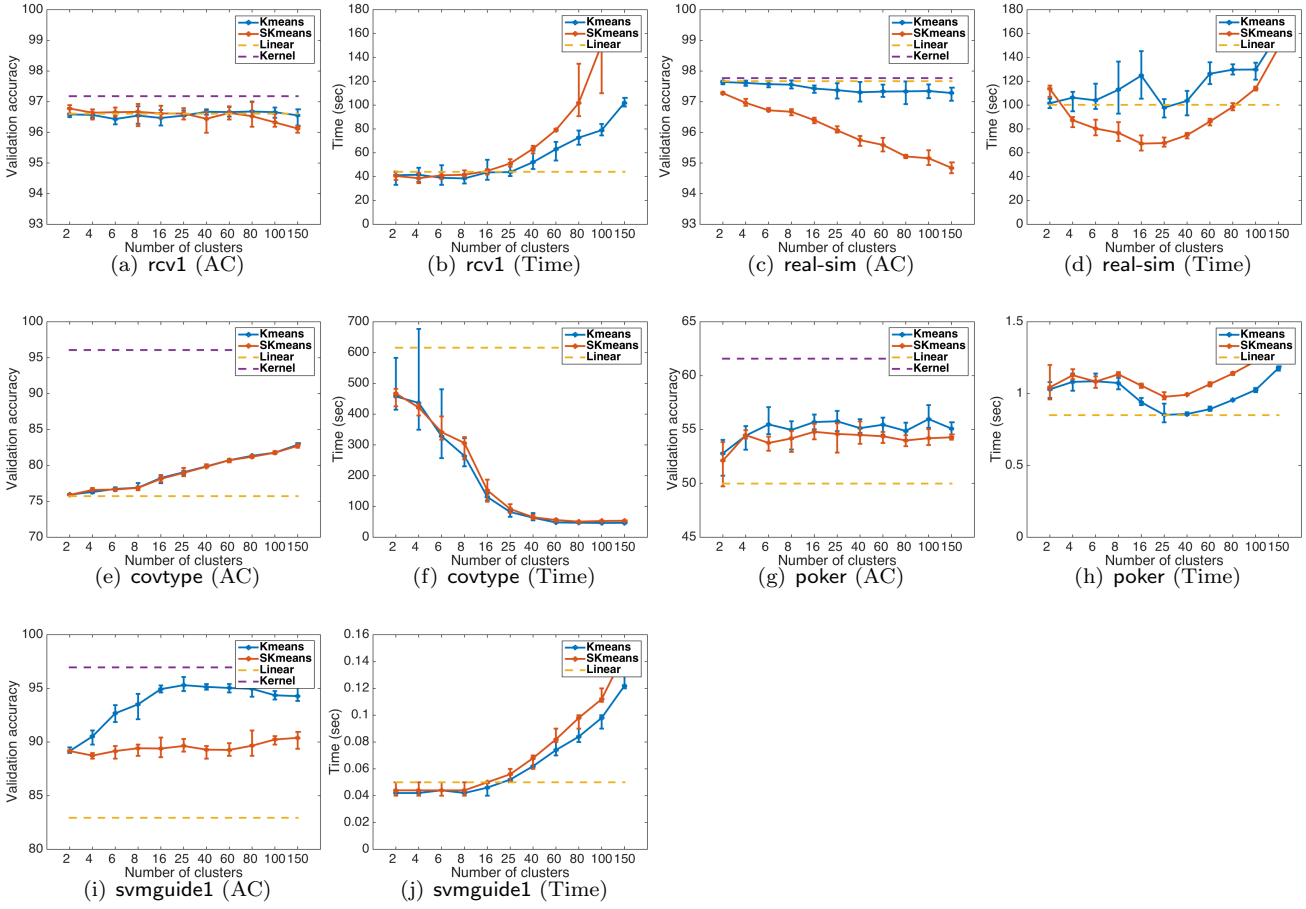
(e) webspam (AC) (f) webspam (Time) (g) madelon (AC) (h) madelon (Time)

(i) a9a (AC) (j) a9a (Time) (k) gisette (AC) (l) gisette (Time)

(m) news20 (AC) (n) news20 (Time) (o) cod-rna (AC) (p) cod-rna (Time)

(q) german.numer (AC) (r) german.numer (Time) (s) ijcnn1 (AC) (t) ijcnn1 (Time)

Figure (II): Performance of MultiLinear SVM under different settings



(a) rcv1 (AC) (b) rcv1 (Time) (c) real-sim (AC) (d) real-sim (Time)

(e) covtype (AC) (f) covtype (Time) (g) poker (AC) (h) poker (Time)

(i) svmguide1 (AC) (j) svmguide1 (Time)

Figure (III): Performance of MultiLinear SVM under different settings (Continued)