Supplemental Material of "Active Learning by Querying Informative and Representative Examples"

Sheng-Jun Huang[1] **Rong Jin**[2] **Zhi-Hua Zhou**[1]
[1]National Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210093, China
[2]Department of Computer Science and Engineering,
Michigan State University, East Lansing, MI 48824
{huangsj, zhouzh}@lamda.nju.edu.cn rongjin@cse.msu.edu

The Connection between Eq. 2 and Eq. 3

In this section, we try to establish the connection between Eq. 2 and Eq. 3, i.e., the connection between

$$s^* = \arg\min_{n_l < s \leq n} |f^*(\mathbf{x}_s)|$$

and

$$s^* = \arg\min_{n_1 < s \leq n} \mathcal{L}(\mathcal{D}_l, \mathbf{x}_s),$$

where

$$\mathcal{L}(\mathcal{D}_l, \mathbf{x}_s) = \max_{y_s=\pm 1} \min_{f \in \mathcal{H}} \frac{\lambda}{2}|f|^2_{\mathcal{H}} + \sum_{i=1}^{n_l} \ell(y_i, f(\mathbf{x}_i)) + \ell(y_s, f(\mathbf{x}_s)).$$

Proof. Denote by $\mathcal{J}(f)$ the object function, i.e.,

$$\mathcal{J}(f) = \frac{\lambda}{2}|f|^2_{\mathcal{H}} + \sum_{i=1}^{n_l} \ell(y_i, f(\mathbf{x}_i)),$$

We have

$$\begin{aligned}
s^* &= \arg\min_{n_l < s \leq n} |f^*(\mathbf{x}_s)| \\
&= \arg\min_{n_l < s \leq n} \min_{f \in \mathcal{H}; f: \mathcal{J}(f) \leq \mathcal{J}(f^*)} |f(x_s)| \\
&= \arg\min_{n_l < s \leq n} \min_{f \in \mathcal{H};} |f(x_s)| + C\mathcal{J}(f) \\
&= \arg\min_{n_l < s \leq n} \max_{y_s=\pm 1} \min_{f \in \mathcal{H}} \ell(y_s, f(\mathbf{x}_s)) + C\mathcal{J}(f) \\
&= \arg\min_{n_l < s \leq n} \max_{y_s=\pm 1} \min_{f \in \mathcal{H}} C(\frac{\lambda}{2}|f|^2_{\mathcal{H}} + \sum_{i=1}^{n_l} \ell(y_i, f(\mathbf{x}_i))) + \ell(y_s, f(\mathbf{x}_s))
\end{aligned}$$

Let $C = 1$, we have

$$s^* = \arg\min_{n_1 < s \leq n} \mathcal{L}(\mathcal{D}_l, \mathbf{x}_s)$$

□

Proof of Theorem 1

Theorem 1. *Let*

$$L_{a,a}^{-1} = \begin{pmatrix} L_{s,s} & L_{s,u} \\ L_{u,s} & L_{u,u} \end{pmatrix}^{-1} = \begin{pmatrix} a & -\mathbf{b}^\top \\ -\mathbf{b} & D \end{pmatrix}.$$

We have

$$L_{u,u}^{-1} = D - \frac{1}{a}\mathbf{b}\mathbf{b}^\top.$$

Proof. Using the matrix inversion lemma, we have

$$L_{a,a}^{-1} = \left(\begin{array}{cc} L_{s,s} & L_{s,u} \\ L_{u,s} & L_{u,u} \end{array} \right)^{-1} = \left(\begin{array}{cc} a & -\mathbf{b}^T \\ -\mathbf{b} & D \end{array} \right) = \left(\begin{array}{cc} C_1^{-1} & -\frac{1}{L_{s,s}} L_{u,s}^T C_2^{-1} \\ -\frac{1}{L_{s,s}} C_2^{-1} L_{u,s} & C_2^{-1} \end{array} \right)$$

where $$C_1 = L_{s,s} - L_{u,s}^T {L_{u,u}}^{-1} L_{u,s}, \quad C_2 = L_{u,u} - \frac{1}{L_{s,s}} L_{u,s} L_{u,s}^T.$$

With the equation above, we can express a, $\mathbf{b}$ and D in terms of L as follows:

$$\begin{aligned}
\frac{1}{a} &= C_1 = L_{s,s} - L_{u,s}^T {L_{u,u}}^{-1} L_{u,s} \\
D &= C_2^{-1} = \left(L_{u,u} - \frac{1}{L_{s,s}} L_{u,s} L_{u,s}^T \right)^{-1} \\
&= {L_{u,u}}^{-1} + {L_{u,u}}^{-1} L_{u,s} \left(L_{s,s} - L_{u,s}^T {L_{u,u}}^{-1} L_{u,s} \right)^{-1} L_{u,s}^T {L_{u,u}}^{-1} \\
&= {L_{u,u}}^{-1} + a {L_{u,u}}^{-1} L_{u,s} L_{u,s}^T {L_{u,u}}^{-1} \\
\mathbf{b} &= \frac{1}{L_{s,s}} C_2^{-1} L_{u,s} = \frac{1 + a L_{u,s}^T {L_{u,u}}^{-1} L_{u,s}}{L_{s,s}} {L_{u,u}}^{-1} L_{u,s} = a {L_{u,u}}^{-1} L_{u,s}
\end{aligned}$$

We complete the proof by combining the above relationships. □

Data set information

Table 1: Data set information. *Size*: the number of instances. *Feature*: the number of features.

Data	*Size*	*Feature*	*Data*	*Size*	*Feature*	*Data*	*Size*	*Feature*
austra	690	14	*titato*	958	9	*letterEvsF*	1543	16
digit1	1500	241	*vehicle*	435	18	*letterIvsJ*	1502	16
g241n	1500	241	*wdbc*	569	30	*letterMvsN*	1575	16
isolet	600	617	*letterDvsP*	1608	16	*letterUvsV*	1577	16

Wilcoxon signed ranks test result

Table 2 summarizes the win/tie/loss counts of QUIRE versus the other methods based on Wilcoxon signed ranks test at 95% significance level. We observe that the results are almost as same as that based on paired t-tests.

Computational cost

All the experiments are performed with MATLAB 7.6 on a 3.00GHZ Intel(R) Core(TM)2 DUO PC running Windows 7 with 4GB main memory, average CPU time of each round for all the six approaches on average of each data set is reported in Table 3.

Table 2: Win/tie/loss counts of QUIRE versus the other methods with varied numbers of queries based on Wilcoxon signed ranks test at 95% significance level.

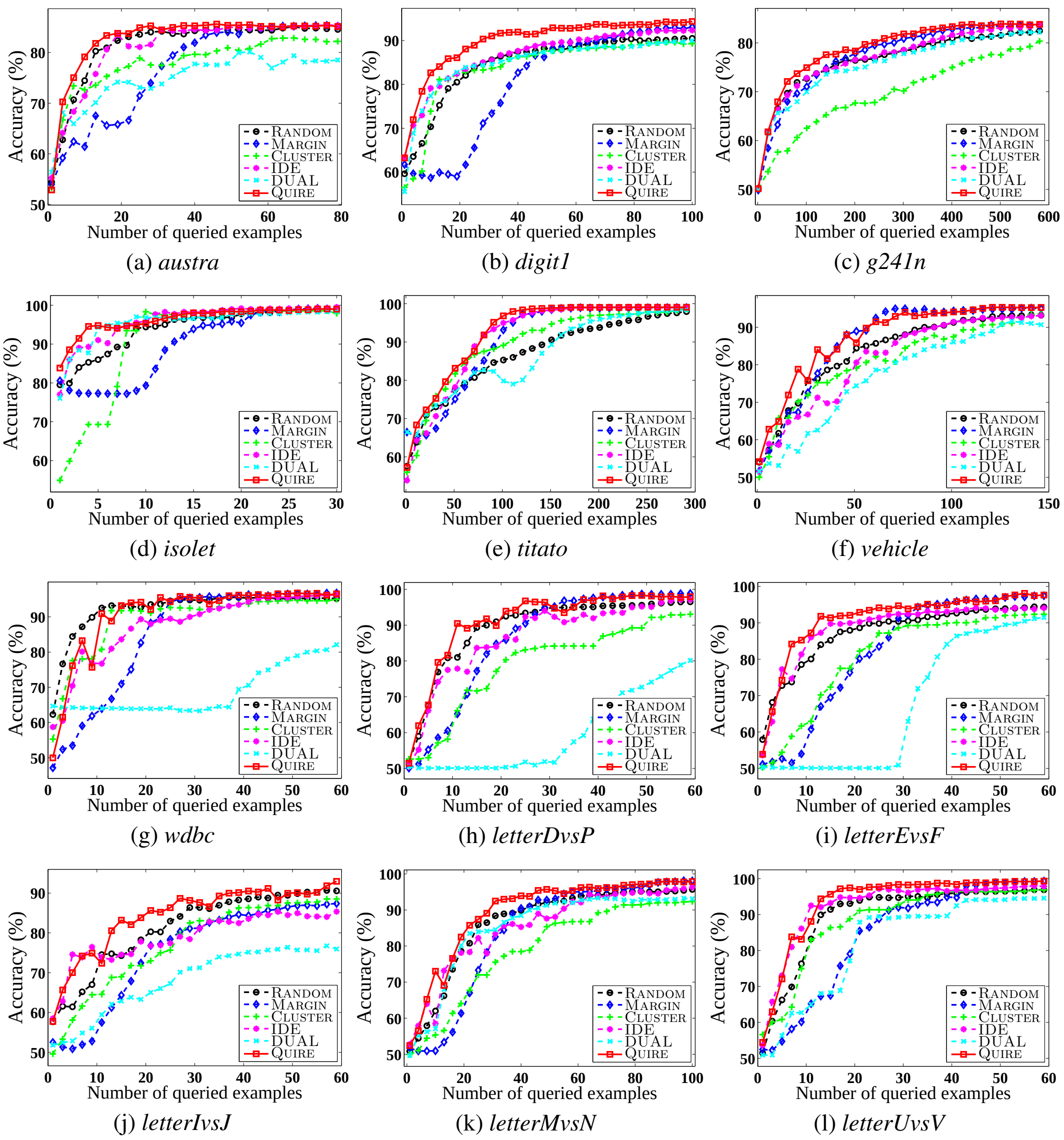
Algorithms	Number of queries (percentage of the unlabeled data)							In All
	5%	10%	20%	30%	40%	50%	80%	
RANDOM	4/8/0	7/5/0	9/3/0	8/3/1	10/2/0	9/3/0	7/5/0	54/29/1
MARGIN	8/4/0	4/7/1	3/7/2	2/8/2	0/10/2	0/11/1	0/12/0	17/59/8
CLUSTER	6/6/0	8/4/0	8/4/0	11/1/0	8/4/0	6/6/0	2/10/0	49/35/0
IDE	5/7/0	7/4/1	6/5/1	7/5/0	8/3/1	8/4/0	3/9/0	44/37/3
DUAL	7/5/0	10/2/0	11/1/0	11/1/0	11/1/0	11/1/0	9/3/0	70/14/0
In All	30/30/0	36/22/2	37/20/3	39/18/3	37/20/3	34/25/1	21/39/0	234/174/12

Table 3: Average CPU time of each round for compared methods

Data	Algorithms					
	RANDOM	MARGIN	CLUSTER	IDE	DUAL	QUIRE
austra	0.0001	0.0173	0.0072	0.0265	2.0109	.1880
digit1	0.0002	0.2018	0.0109	0.0435	9.3486	3.3787
g241n	0.0002	0.3955	0.0198	0.0725	6.6166	3.3816
isolet	0.0001	0.0686	0.0059	0.0284	7.9308	0.1445
titato	0.0001	0.0310	0.0085	0.0335	1.8330	0.8326
vehicle	0.0001	0.0057	0.0048	0.0176	0.1845	0.0535
wdbc	0.0001	0.0070	0.0053	0.0224	0.5171	0.1313
letterDvsP	0.0002	0.0311	0.0131	0.0405	5.1526	3.7448
letterEvsF	0.0002	0.0331	0.0120	0.0395	1.1038	4.2273
letterIvsJ	0.0002	0.0470	0.0135	0.0424	1.6074	3.6689
letterMvsN	0.0002	0.0417	0.0121	0.0442	4.5766	3.5365
letterUvsV	0.0002	0.0275	0.0118	0.0415	4.7951	4.6030
Average	0.0002	0.0756	0.0104	0.0377	3.8064	2.3242

Experiment of Active Learning with A Few Initially Labeled Examples

In this experiment, we consider two settings: in the first setting, only one positive example and one negative example are available at the beginning of active learning; in the second setting, we increase the number of initially labeled examples to ten, with five positive examples and five negative examples. Figures 1 and 2 show the classification accuracy of the proposed algorithm and the baseline methods for these two settings, respectively. First, we observe that for most of the cases, the proposed algorithm still outperforms the baseline methods, even though the advantage of the proposed algorithm starts to diminish as more and more initially labeled examples are available. Second, we observe that the DUAL approach, which performs poorly when no initially labeled examples are available, is able to improve its performance significantly for some data sets.



(a) *austra* (b) *digit1* (c) *g241n*

(d) *isolet* (e) *titato* (f) *vehicle*

(g) *wdbc* (h) *letterDvsP* (i) *letterEvsF*

(j) *letterIvsJ* (k) *letterMvsN* (l) *letterUvsV*

Figure 1: Comparison on classification accuracy with 2 initially labeled examples

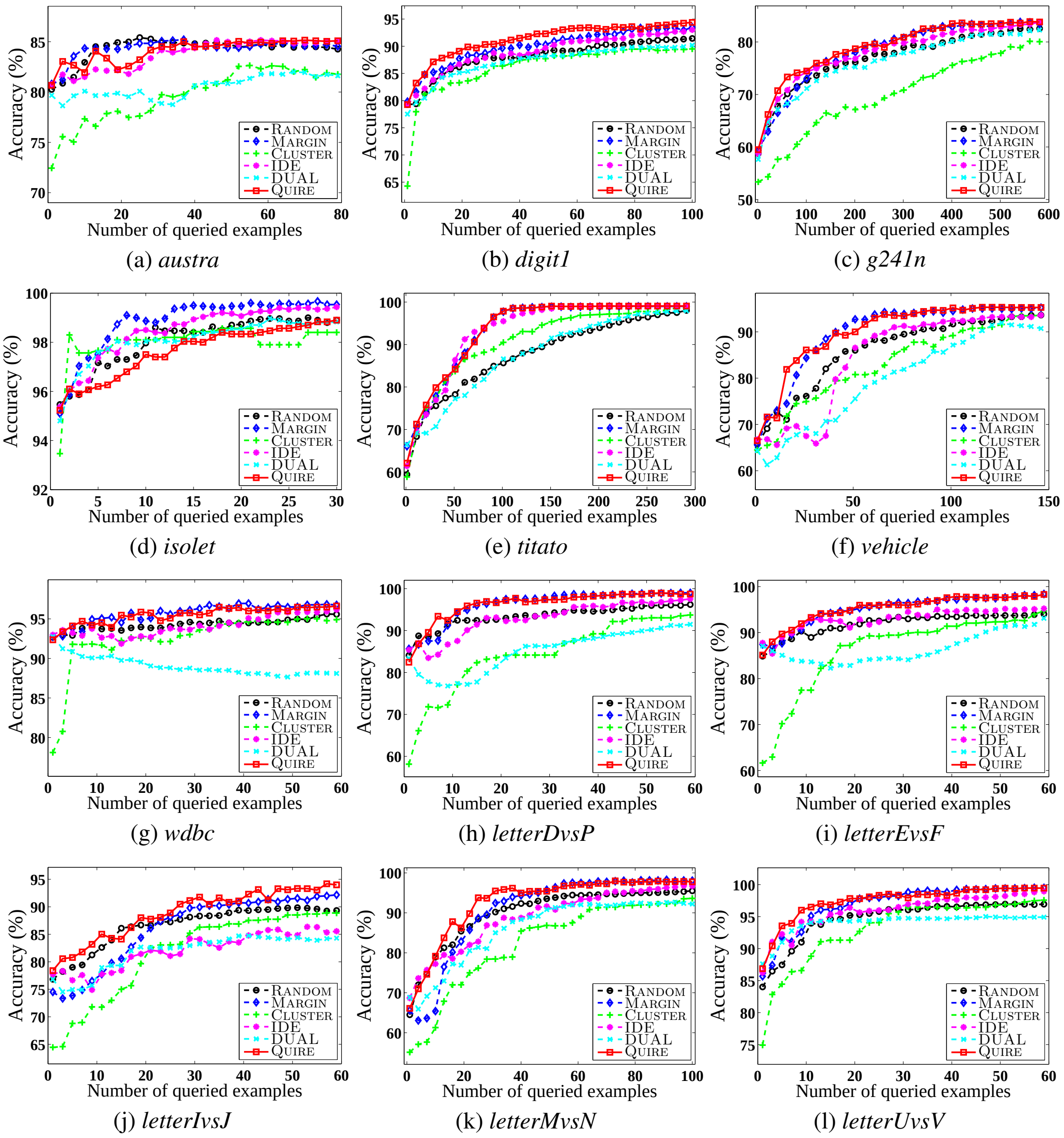


Figure 2: Comparison on classification accuracy with 10 initially labeled examples