Orthogonality-based label correction in multi-class classification

H. Xue and S. Chen

Orthogonality-based label coding is an often-used technique in multi-class classification. Through coding the labels into some multi-dimensional orthogonal codewords, many binary classifiers can be naturally extended to multi-class cases. For an unseen sample, the classifiers firstly estimate its codeword and then compute the corresponding distances from the labels. Finally, the nearest one is assigned as its class label. However, these classifiers actually hardly guarantee that the estimated codewords still maintain the inter-orthogonality with the other classes, which more likely causes the codewords in different classes overlapping each other to some extent and thus affects the classification performance. Proposed is a novel label correction strategy which aims to keep as much as possible orthogonality between the estimated sample codewords and the other classes' labels in order to preserve further as much as possible the inter-orthogonality of the codewords. The strategy is combined with two state-of-the-art classifiers: regularised least square classifier and the least square support vector machine. Experiments on UCI datasets demonstrate the effectiveness of the method.

Introduction: Multi-class classification is widespread in real applications, such as face recognition, text mining and medical analysis [1]. Compared to binary classification, multi-class classification is more delicate, since many existing successful classifiers are basically designed to focus on binary rather than multi-class issues [2]. Up to now, many strategies have been developed to solve this problem, which can fall into three basic categories [2]. The first category is label coding which can extend some binary classifiers to multi-class scenarios directly. By transforming the labels into some codewords, classification is changed into computing the nearest distance between the estimated sample codewords and the labels. The second category is decomposing the multi-class problem into several binary classification tasks that can be efficiently solved using binary classifiers, e.g. the support vector machine [2]. The corresponding strategies involve one-versus-all, all-versus-all and error-correcting output coding. The third category relies on arranging the classes in a hierarchy tree and utilising a number of binary classifiers at the nodes of the tree till a leaf node is reached [2].

Orthogonality-based label coding is the most often-used coding format in the first category, where a typical paradigm is one-of-*c* coding. Through transforming the labels into some orthogonal multi-dimensional codewords, the coding attempts to maximise the diversity of the classes. Many binary classifiers can directly use these codewords instead of the one-dimensional binary-class labels to solve multi-class classification, and aim to make the estimated sample codewords close to the corresponding label codewords, such as decision trees [3], neural networks [4], the regularised least square classifier (RLSC) [4] and the least square support vector machine (LSSVM) [5]. However, these classifiers basically overlook an important problem – the estimated sample codewords actually hardly keep the inter-orthogonality as the label codewords, which more likely leads to the codewords of different classes overlapping each other to some extent and further increase the incorrect classification. In this Letter, we present a simple but general strategy to correct these codewords. By constraining the inner products between the codewords and the other classes' labels as an extra regulariser embedded into the original optimisation objectives of these classifiers, the strategy can make the codewords as orthogonal to the other classes' labels as possible, and further as close to the corresponding class's label as possible. As a result, the strategy can naturally preserve the inter-orthogonality among the codewords of different classes as much as possible. To validate the effectiveness of our proposed strategy, we further apply it in RLSC and LSSVM. The experiments on several UCI datasets demonstrate that the new methods have much better classification performance than the original RLSC and LSSVM.

Orthogonality-based label correction in RLSC (OLC-RLSC): Assume a multi-class dataset$\{(\boldsymbol{x}_i, z_i)\}_{i=1}^{N}$, where $\boldsymbol{x}_i \in \boldsymbol{R}^Mandz_i \in \{1, \cdots, C\}$ is the corresponding class label to $\boldsymbol{x}_i$. Following the orthogonality-based label coding method, we can code z_i into a new C-dimensional vector $\boldsymbol{y}_i$. Consequently, we obtain the label codeword matrix:

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}^{(1)^T} \\ \vdots \\ \boldsymbol{y}^{(C)^T} \end{bmatrix} \in \boldsymbol{R}^{C \times N} \tag{1}$$

where $\boldsymbol{y}^{(i)^T} \boldsymbol{y}^{(j)} = 0,\ i \neq j,\ i, j = 1, \cdots, C$.

A simplest thought of preserving the inter-orthogonality is directly minimising the inner products of the sample codewords in different classes. However, this may lead to complex optimisation owing to its non-dual form and destroy the solution framework of RLSC. Consequently, here we adopt an alternative strategy. We penalise the inner products of the codewords with the other classes' labels as a regulariser, which fully utilises the inter-orthogonality of the labels. We term the regulariser as an orthogonality-based label correction regulariser and directly embed it into RLSC [4] to improve OLC-RLSC. The corresponding objective function of OLC-RLSC can be described as follows:

$$\min_{\boldsymbol{W}} \frac{1}{2} |\boldsymbol{W}^T \boldsymbol{X} - \boldsymbol{Y}|^2 + \frac{\lambda_1}{2} |\boldsymbol{W}|_F^2 + \frac{\lambda_2}{2} \sum_{c=1}^{C} \sum_{c' \neq c} [\boldsymbol{w}_c^T \boldsymbol{X} \cdot \boldsymbol{y}^{(c')}]^2 \tag{2}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N] \in \boldsymbol{R}^{M \times N}$. $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_C] \in \boldsymbol{R}^{M \times C}$ is the wanted discriminant vector matrix. λ_1 and λ_2 are two regularisation parameters. Obviously, when $\lambda_2 = 0$, OLC-RLSC will degenerate to the original RLSC. When $\lambda_2 \neq 0$, by minimising the regulariser, OLC-RLSC can make the estimated sample codeword $\boldsymbol{w}_c^T \boldsymbol{X}$ in the cth class orthogonal to the different classes' labels $\boldsymbol{y}^{(c')}$. As a result, $\boldsymbol{w}_c^T \boldsymbol{X}$ will more approximate to the cth class label $\boldsymbol{y}^{(c)}$ and thus naturally inherits the inter-orthogonality as the labels.

Similar to RLSC, we can arrive at a convex differentiable function of $\boldsymbol{w}_c$:

$$\begin{aligned} \boldsymbol{w}_c^* = \arg\min \frac{1}{2} \sum_{c=1}^{C} \boldsymbol{w}_c^T \boldsymbol{X} \boldsymbol{X}^T \boldsymbol{w}_c - \sum_{c=1}^{C} \boldsymbol{w}_c^T \boldsymbol{X} \boldsymbol{y}^{(c)} + \frac{1}{2} tr(\boldsymbol{Y}^T \boldsymbol{Y}) + \frac{\lambda_1}{2} \sum_{c=1}^{C} \boldsymbol{w}_c^T \boldsymbol{w}_c \\ + \frac{\lambda_2}{2} \sum_{c=1}^{C} \boldsymbol{w}_c^T \sum_{c' \neq c} (\boldsymbol{X} \boldsymbol{y}^{(c')} \boldsymbol{y}^{(c')^T} \boldsymbol{X}^T) \boldsymbol{w}_c \end{aligned} \tag{3}$$

The derivative of the function vanishes at the minimiser:

$$\boldsymbol{X}\boldsymbol{X}^T \boldsymbol{w}_c - \boldsymbol{X}\boldsymbol{y}^{(c)} + \lambda_1 \boldsymbol{w}_c + \lambda_2 \sum_{c' \neq c} (\boldsymbol{X} \boldsymbol{y}^{(c')} \boldsymbol{y}^{(c')^T} \boldsymbol{X}^T) \boldsymbol{w}_c = \boldsymbol{0} \tag{4}$$

which leads to the following solution:

$$\boldsymbol{w}_c^* = \left[\boldsymbol{X}\boldsymbol{X}^T + \lambda_1 \boldsymbol{I} + \lambda_2 \sum_{c' \neq c} (\boldsymbol{X} \boldsymbol{y}^{(c')} \boldsymbol{y}^{(c')^T} \boldsymbol{X}^T) \right]^{-1} \boldsymbol{X} \boldsymbol{y}^{(c)} \tag{5}$$

Orthogonality-based label correction in LSSVM (OLC-LSSVM): LSSVM is another state-of-the-art binary classifier which can be directly generalised to solve multi-class classification analytically. We further integrate the orthogonality-based label correction strategy with LSSVM [6] and extend to OLC-LSSVM by solving the following problem:

$$\begin{aligned} & \min_{\boldsymbol{w}_c} \frac{1}{2} \sum_{c=1}^{C} \boldsymbol{w}_c^T \boldsymbol{w}_c + \frac{\lambda_1}{2} \sum_{k=1}^{N} \sum_{c=1}^{C} e_{k,c}^2 + \frac{\lambda_2}{2} \sum_{c=1}^{C} \sum_{c' \neq c} [\boldsymbol{w}_c^T \boldsymbol{X} \cdot \boldsymbol{y}^{(c')}]^2 \\ & s.t. \begin{cases} \boldsymbol{y}_k^{(1)} (\boldsymbol{w}_1^T \boldsymbol{x}_k + b_1) = 1 - e_{k,1}, & k = 1, \cdots, N \\ \vdots & \vdots \\ \boldsymbol{y}_k^{(C)} (\boldsymbol{w}_C^T \boldsymbol{x}_k + b_C) = 1 - e_{k,C}, & k = 1, \cdots, N \end{cases} \end{aligned} \tag{6}$$

Introducing the Lagrangian with $\alpha_{k,i}$ as Lagrange multipliers:

$$\begin{aligned} L = \frac{1}{2} \sum_{c=1}^{C} \boldsymbol{w}_c^T \boldsymbol{w}_c + \frac{\lambda_1}{2} \sum_{k=1}^{N} \sum_{c=1}^{C} e_{k,c}^2 + \frac{\lambda_2}{2} \sum_{c=1}^{C} \boldsymbol{w}_c^T \sum_{c' \neq c} (\boldsymbol{X} \boldsymbol{y}^{(c')} \boldsymbol{y}^{(c')^T} \boldsymbol{X}^T) \boldsymbol{w}_c \\ - \sum_{k=1}^{N} \sum_{c=1}^{C} \alpha_{k,c} [\boldsymbol{y}_k^{(c)} (\boldsymbol{w}_c^T \boldsymbol{x}_k + b_c) - 1 + e_{k,c}] \end{aligned} \tag{7}$$

The conditions for optimality w.r.t. $\boldsymbol{w}_c, b_c, e_{k,c}, \alpha_{k,c}$ for the training respectively become:

$$
\begin{aligned}
&\partial L/\partial \boldsymbol{w}_c = 0 \rightarrow \boldsymbol{w}_c = \sum_{k=1}^{N} \alpha_{k,c}\boldsymbol{y}_k^{(c)}\left[\boldsymbol{I} + \lambda_2 \sum_{c' \neq c} (\boldsymbol{X}\boldsymbol{y}^{(c')}\boldsymbol{y}^{(c')^T}\boldsymbol{X}^T)\right]^{-1} \boldsymbol{x}_k \\
&\partial L/\partial b_c = 0 \rightarrow \sum_{k=1}^{N} \alpha_{k,c}\boldsymbol{y}_k^{(c)} = 0 \\
&\partial L/\partial e_{k,c} = 0 \rightarrow \alpha_{k,c} = \lambda_1 e_{k,c} \\
&\partial L/\partial \alpha_{k,c} = 0 \rightarrow \boldsymbol{y}_k^{(c)}(\boldsymbol{w}_c^T\boldsymbol{x}_k + b_c) = 1 - e_{k,c}
\end{aligned}
\tag{8}
$$

where $k = 1, \cdots, N$ and $c = 1, \cdots, C$. Elimination of $\boldsymbol{w}_c$ and $e_{k,c}$ gives the linear system:

$$
\begin{bmatrix} \boldsymbol{0} & \boldsymbol{Y}_C^T \\ \boldsymbol{Y}_C & \boldsymbol{\Omega}_C \end{bmatrix} \begin{bmatrix} \boldsymbol{b}_C \\ \boldsymbol{\alpha}_C \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \vec{\boldsymbol{1}} \end{bmatrix}
\tag{9}
$$

where

$$
\boldsymbol{Y}_C = blockdiag\left\{\begin{bmatrix} \boldsymbol{y}_1^{(1)} \\ \vdots \\ \boldsymbol{y}_N^{(1)} \end{bmatrix}, \cdots, \begin{bmatrix} \boldsymbol{y}_1^{(C)} \\ \vdots \\ \boldsymbol{y}_N^{(C)} \end{bmatrix}\right\}
$$

$$
\boldsymbol{\Omega}_C = blockdiag\{\boldsymbol{\Omega}^{(1)}, \cdots, \boldsymbol{\Omega}^{(C)}\}
$$

$$
\boldsymbol{\Omega}_{kl}^{(c)} = \boldsymbol{y}_k^{(c)}\boldsymbol{y}_l^{(c)}\boldsymbol{x}_k^T\left[\boldsymbol{I} + \lambda_2 \sum_{c' \neq c} (\boldsymbol{X}\boldsymbol{y}^{(c')}\boldsymbol{y}^{(c')^T}\boldsymbol{X}^T)\right]^{-1} \boldsymbol{x}_l + \lambda_1^{-1}\boldsymbol{I}
$$

Finally, by solving the linear equations (9), we can get the solution vectors:

$$
\boldsymbol{b}_C^* = [b_1, \cdots, b_C]
$$

$$
\boldsymbol{\alpha}_C^* = [\boldsymbol{\alpha}_{1,1}, \cdots, \boldsymbol{\alpha}_{N,1}; \cdots; \boldsymbol{\alpha}_{1,C}, \cdots, \boldsymbol{\alpha}_{N,C}]
$$

Experiments and analyses: To evaluate the proposed OLC-RLSC and OLC-LSSVM, we perform some experimental comparisons with RLSC and LSSVM on several UCI datasets, including Iris(3), Lenses (3), Tae(3), Balance(3), Ecoli(6) and Yeast(10) where the class number is shown in the brackets. We randomly select half of each class for training and the remaining for testing, and repeat the process ten times. The comparisons are conducted both in the linear and radial basis function (RBF) kernel versions. All the parameters in the algorithms including regularisation and kernel parameters are chosen from $\{2^{-10}, 2^{-9}, \ldots, 2^9, 2^{10}\}$ by cross-validation. Tables 1 and 2 present the average classification accuracies (%) and variances in each algorithm where the best performances are highlighted in bold.

Table 1: Classification results compared between RLSC and OLC-RLSC, LSSVM and OLC-LSSVM on UCI datasets with *linear* kernel

Dataset	RLSC	OLC-RLSC	LSSVM	OLC-LSSVM
Iris	82.93 ± 3.48	**84.13** ± 3.34	91.20 ± 1.30	**93.33** ± 0.55
Lenses	80.00 ± 5.50	**81.54** ± 8.20	77.69 ± 12.4	**85.38** ± 5.90
Tae	46.32 ± 2.10	**48.16** ± 1.10	43.68 ± 2.40	**47.11** ± 2.10
Balance	87.15 ± 0.35	**89.24** ± 0.49	87.83 ± 0.44	**89.50** ± 0.61
Ecoli	82.26 ± 3.92	**84.40** ± 3.45	86.68 ± 1.66	**88.33** ± 2.30
Yeast	55.10 ± 1.49	**56.73** ± 1.00	55.64 ± 0.81	**57.51** ± 1.09
Average	72.29 ± 2.81	**74.03** ± 2.93	73.79 ± 3.17	**76.86** ± 2.09

From the Tables, we can see that OLC-RLSC and OLC-LSSVM outperform RLSC and LSSVM, respectively, in all the datasets. Especially, OLC-RLSC exceeds RLSC by more than 2% on the Balance and Ecoli datasets with the linear kernel, and 3% on the Tae dataset with the RBF kernel. Meanwhile, OLC-LSSVM exceeds LSSVM by more than 7% on the Lenses dataset and 3% on the Tae dataset with the linear kernel, and 4% on the two datasets with the RBF kernel. Furthermore, all the algorithms basically perform better with the RBF kernel than with the linear kernel. However, OLC-LSSVM with the linear kernel has been superior to LSSVM with the RBF kernel beyond 3% on the Lenses dataset, which further validates that the proposed label correction strategy can indeed improve the classifier's performance in complex multi-class classifications.

Table 2: Classification results compared between RLSC and OLC-RLSC, LSSVM and OLC-LSSVM on UCI datasets with *RBF* kernel

Dataset	RLSC	OLC-RLSC	LSSVM	OLC-LSSVM
Iris	98.27 ± 0.81	**98.53** ± 0.97	98.13 ± 0.47	**98.67** ± 0.40
Lenses	82.31 ± 6.60	**83.85** ± 8.50	81.77 ± 8.20	**86.15** ± 6.30
Tae	54.05 ± 5.61	**57.24** ± 2.40	54.47 ± 5.10	**60.26** ± 2.92
Balance	90.60 ± 0.56	**91.95** ± 0.11	90.66 ± 0.12	**92.69** ± 0.10
Ecoli	88.29 ± 1.59	**89.40** ± 1.89	88.64 ± 1.97	**90.06** ± 2.35
Yeast	60.07 ± 1.73	**61.49** ± 1.51	60.32 ± 1.97	**62.18** ± 1.65
Average	78.93 ± 2.82	**80.41** ± 2.56	79.00 ± 2.97	**81.67** ± 2.29

Conclusion: In this Letter, a novel orthogonality-based label correction strategy for multi-class classification is proposed. Through maximising the orthogonality between the estimated sample codewords and the other classes' labels, the strategy aims to maximise the inter-orthogonality among different class sample codewords in order to guarantee the diversity of classes as much as possible. We further integrate the strategy into RLSC and LSSVM as an extra regulariser, and obtain two new multi-class algorithms OLC-RLSC and OLC-LSSVM. Experimental results on several UCI datasets demonstrate the superiority of the two algorithms.

Acknowledgments: This work was partially supported by the NSFC (grants 61170151, 60905002, 61035003) and the Scientific Research Startup Project of New Doctorial Faculties of Southeast University and sponsored by the Quinlan Project. It was also supported by the Key Lab. of Computer Network and Information Integration (Southeast University), Ministry of Education.

© The Institution of Engineering and Technology 2013

20 December 2012
doi: 10.1049/el.2012.4408

H. Xue (*School of Computer Science and Engineering, Southeast University, Nanjing, 210096, People's Republic of China*)

E-mail: hxue@seu.edu.cn

S. Chen (*Department of Computer Science and Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing, 210016, People's Republic of China*)

H. Xue: Also with the State Key Lab. for Novel Software Technology, Nanjing University, People's Republic of China.

References

1 Wang, Y., Chen, S., and Xue, H.: 'Can under-exploited structure of original-classes help ECOC-based multi-class classification?' *Neurocomputing*, 2012, **89**, pp. 158–167

2 Aly, M.: 'Survey on multiclass classification methods, Tech. Rep.', California Institute of Technology, 2005

3 Quinlan, J.R.: 'C4.5: Programs for machine learning' (Morgan Kaufmann, 1993)

4 Haykin, S.: 'Neural networks: A comprehensive foundation.' (Tsinghua University Press, 2001)

5 Suykens, J.A.K., and Vandewalle, J.: 'Least squares support vector machine classifiers'. *Neural Process. Lett.*, 1999, **9**, pp. 293–300

6 Suykens, J.A.K., and Vandewalle, J.: 'Multiclass least squares support vector machines'. Proc. of Int. Joint Conf. on Neural Networks, Washington, OC, USA, July 1999