

DUOL: A Double Updating Approach for Online Learning

Peilin Zhao¹, Steven C.H. Hoi¹, Rong Jin²

¹Nanyang Technological University, ²Michigan State University

Framework of Online Learning

Initialize prediction function $f_0 = g$
for $t=1,2,\dots,T$ **do**
 Receive instance $x_t \in \mathcal{R}^n$
 Predict a label $\hat{y}_t = \text{sign}(f_{t-1}(x_t)) \in \{-1,+1\}$
 Receive the true label $y_t \in \{-1,+1\}$
 If $\hat{y}_t \neq y_t$, then algorithm suffer a mistake
if condition C satisfied then
 Update prediction function $f_{t-1} \rightarrow f_t$
end if
end for

g can be any reasonable function
 If $\text{sign}(f_{t-1}(x_t)) = 0$, we can simply assign $\hat{y}_t = +1$
 C can be any reasonable condition, for example, $\hat{y}_t \neq y_t$

Kernel function $\kappa(x, y): \mathcal{R}^n \times \mathcal{R}^n \rightarrow \mathcal{R}$ measures the similarity between the two instances x and y .

Kernel Based Perceptron

initialize $f_0 = 0$, **weight** $\alpha = 1$
for $t = 1, 2, \dots, T$ **do**
 Receive new instance
 Predict $\hat{y}_t = \text{sign}(f_{t-1}(x_t))$
 Receive label
if $\hat{y}_t \neq y_t$, **then**
 $f_t(x) = f_{t-1}(x) + \alpha y_t \kappa(x, x)$
end if
end for

Limitation of Perceptron: the weights α assigned to the misclassified examples (or support vectors) remain unchanged during the entire learning process.

Similar with Perceptron, most online learning algorithms keep the weights of the existing support vectors unchanged.

Although some algorithms are capable of dynamically adjusting the weights, they are designed not to improve the classification accuracy.

ALGORITHM: Perceptron

Motivation

Proposition 1. Let (x_s, y_s) be an example misclassified by the current classifier $f(x) = \sum_{s=1}^t \alpha_s \kappa(x, x_s)$ with $\alpha_s \geq 0, s=1, \dots, t$, i.e., $y_s f(x_s) < 0$. Then let the updated classifier be $f'(x) = \beta_s \kappa(x, x_s) + f(x)$ with $\beta_s > 0$. There exists at least one support vector $x_{i_1}, 1 \leq i_1 \leq n$ such that $y_{i_1} f'(x_{i_1}) > y_{i_1} f(x_{i_1})$.

So, we propose to update the weight for one existing support vector, and refer it as **auxiliary example**, which in particular satisfy:

- $y_{i_1} f(x_{i_1}) \leq 0$, support vector is misclassified by the current classifier;
- $\kappa(x_s, x_{i_1}) y_s y_{i_1} \leq -\rho$, where $\rho \geq 0$ is a predefined threshold, i.e., support vector (x_s, y_s) "conflicts" with the new misclassified example (x_{i_1}, y_{i_1}) ;

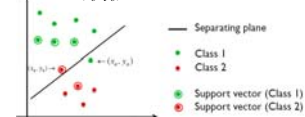


Figure 1. Illustration of an auxiliary example.

Related Work

- Perceptron:**
 - [1] The perceptron: A probabilistic model for information storage and organization in the brain, Rosenblatt, F., Psychological Review, 65, 386–407, 1958
 - [2] Large margin classification using the perceptron algorithm. Freund, Y., & Schapire, R. E., Mach. Learn., 37, 277–296, 1999.
- Dual Ascent Framework:**
 - [3] Online learning meets optimization in the dual, Shalev-Shwartz, S., & Singer, Y. COLT'06.
 - [4] A primal-dual perspective of online learning algorithms, Shalev-Shwartz, S., & Singer, Y. Machine Learning'07
- Large Margin Classifiers:**
 - [5] The relaxed online maximum margin algorithm. Li, Y., & Long, P. M. NIPS'99.
 - [6] A new approximate maximal margin classification algorithm. Gentile, C. JMLR'01.
 - [7] Online passive-aggressive algorithms, Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. JMLR'06.

DUOL

Primal problem: $\min_{\gamma} \frac{1}{2} \|\gamma\|_{H_c}^2 + C \sum_{t=1}^T I(y_t f(x_t))$
Dual problem: $D_t(\Delta, \gamma_1, \dots, \gamma_T) = \sum_{t=1}^T \gamma_t - \sum_{t=1}^T \gamma_t f_t(x_t) + \frac{1}{2} \|\gamma\|_{H_c}^2$, where $f_t(x) = \sum_{s=1}^t \gamma_s \kappa(x, x_s)$ and $\gamma_t \in [0, 1]$.
Dual ascent: $\Delta_t = D_t - D_{t-1}$.

If $\Delta_t \geq \Delta \forall t$, then $M \leq \frac{1}{\Delta} \left(\min_{\Delta} \frac{1}{2} \|\gamma\|_{H_c}^2 + C \sum_{t=1}^T I(y_t f(x_t)) \right)$, where M denotes the number of mistakes (According to paper [3] and [4]).

When adjust the weights for (x_s, y_s) and auxiliary example (x_{i_1}, y_{i_1}) , the dual ascent is $\Delta_t(\Delta, \gamma_s, \gamma_{i_1}) = D_t(\gamma_s, \dots, \gamma_s + \Delta \gamma_s, \dots, \gamma_{i_1} - \Delta, \gamma_{i_1}, \dots, \gamma_{i_1}) - D_{t-1}(\gamma_s, \dots, \gamma_s, \dots, \gamma_{i_1}, \dots, \gamma_{i_1})$

Theorem 1. Assume $C \geq \hat{\gamma}_s + 1/(1-\rho)$ for the selected auxiliary example (x_s, y_s) . Then setting the two weights as $\gamma_s = \min(C, 1/(1-\rho))$ and $\hat{\gamma}_s = \min(C, \hat{\gamma}_s + 1/(1-\rho))$ will guarantee $\Delta_t \geq \frac{\rho}{1-\rho}$.

Mistake Bound

Theorem 3. Let us denote by $(x_1, y_1), \dots, (x_T, y_T)$ a sequence of instance-label pairs where $x_t \in \mathcal{R}^n, y_t \in \{-1,+1\}$ and $\kappa(x_t, x_t) \leq 1, \forall t$. Assume C is sufficiently large. For any function $f \in H_c$, the number of prediction mistakes M made by DUOL on this sequence of examples is bounded by:

$$2 \left(\min_{f \in H_c} \frac{1}{2} \|f\|_{H_c}^2 + C \sum_{t=1}^T I(y_t f(x_t)) \right) - \frac{1+\rho}{1-\rho} M_d(\rho)$$

where $M_d(\rho)$ is the number of mistakes when there is an auxiliary example, which depends on the threshold and the dataset.

Experimental Results

Experimental Testbed and Setup

Datasets: "german", "splice", "spambase", "MITFace", "a7a", and "w7a".
 Download from:
<http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>
<http://www.ics.uci.edu/~mllearn/MLRepository.html>
<http://cbcl.mit.edu/software-datasets>

Algorithms: We compared DUOL with Perceptron, ROMMA and its aggressive version "agg-ROMMA", $ALMA_p(\alpha)$, PA-1 and PA-1I.

We set $\rho = 2$ and $\alpha = 0.9$ for $ALMA_p(\alpha)$.

All the algorithms employ RBF kernel $\kappa(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$. The kernel width σ is set to be 8 and parameter C is set to be 5 for all the datasets. The threshold ρ for DUOL is set to be 0.2.

Table 1: Evaluation on german (n=1000, d=24)					Table 2: Evaluation on splice (n=1000, d=6)				
Algorithm	Mistake (%)	Support Vectors (%)	Time (s)	Time (ms)	Algorithm	Mistake (%)	Support Vectors (%)	Time (s)	Time (ms)
Perceptron	18.25 ± 0.150	151.00 ± 11.80	0.016	1.54	Perceptron	27.120 ± 0.075	251.20 ± 9.74	0.016	1.54
ROMMA	18.108 ± 1.189	151.00 ± 11.80	0.154	15.4	ROMMA	25.960 ± 0.814	255.60 ± 8.14	0.085	8.5
agg-ROMMA	18.390 ± 1.287	164.20 ± 12.31	1.068	106.8	agg-ROMMA	22.900 ± 0.700	462.90 ± 7.43	0.800	80.0
ALMA _{0.9} (0.9)	34.025 ± 0.910	402.00 ± 7.31	0.228	22.8	ALMA _{0.9} (0.9)	28.840 ± 0.908	314.80 ± 9.41	0.075	7.5
PA-I	18.670 ± 0.274	280.00 ± 6.74	0.029	2.9	PA-I	23.815 ± 1.042	465.60 ± 5.60	0.028	2.8
PA-II	18.175 ± 1.229	273.00 ± 10.02	0.029	2.9	PA-II	23.515 ± 1.045	469.00 ± 7.45	0.028	2.8
Online-SVM	28.860 ± 0.651	646.10 ± 3.00	16.087	1608.7	Online-SVM	17.455 ± 0.518	614.00 ± 2.92	12.243	1224.3
DUOL	20.990 ± 0.033	622.30 ± 12.87	0.089	8.9	DUOL	20.960 ± 0.566	573.85 ± 8.93	0.076	7.6

Table 3: Evaluation on spambase (n=4601, d=57)					Table 4: Evaluation on mushrooms (n=8124, d=112)				
Algorithm	Mistake (%)	Support Vectors (%)	Time (s)	Time (ms)	Algorithm	Mistake (%)	Support Vectors (%)	Time (s)	Time (ms)
Perceptron	21.987 ± 0.184	1102.00 ± 17.17	0.208	20.8	Perceptron	2.683 ± 0.274	109.20 ± 22.58	0.132	13.2
ROMMA	21.955 ± 0.510	1102.00 ± 17.14	10.129	1012.9	ROMMA	2.420 ± 0.101	197.35 ± 8.24	0.208	20.8
agg-ROMMA	21.242 ± 0.174	2380.00 ± 21.32	950.028	95002.8	agg-ROMMA	1.588 ± 0.098	332.80 ± 39.99	17.441	1744.1
ALMA _{0.9} (0.9)	21.579 ± 0.441	1530.15 ± 15.65	25.284	2528.4	ALMA _{0.9} (0.9)	2.538 ± 0.297	564.80 ± 30.02	0.405	40.5
PA-I	22.112 ± 0.174	2380.00 ± 21.32	0.090	9.0	PA-I	1.681 ± 0.089	322.55 ± 22.80	0.442	44.2
PA-II	21.907 ± 0.140	3029.10 ± 24.87	0.505	50.5	PA-II	1.657 ± 0.088	328.20 ± 22.85	0.473	47.3
Online-SVM	11.118 ± 0.221	2286.00 ± 20.70	252.045	25204.5	Online-SVM	0.480 ± 0.050	405.20 ± 6.75	1384.665	138466.5
DUOL	19.436 ± 0.421	2526.30 ± 20.57	0.085	8.5	DUOL	1.170 ± 0.077	409.00 ± 11.40	0.366	36.6

Table 5: Evaluation on a7a (n=16100, d=123)					Table 6: Results on w7a (n=24292, d=300)				
Algorithm	Mistake (%)	Support Vectors (%)	Time (s)	Time (ms)	Algorithm	Mistake (%)	Support Vectors (%)	Time (s)	Time (ms)
Perceptron	22.020 ± 0.202	545.50 ± 31.40	2.043	204.3	Perceptron	4.037 ± 0.055	961.40 ± 23.97	1.333	133.3
ROMMA	21.297 ± 0.272	3428.45 ± 43.77	306.793	30679.3	ROMMA	4.158 ± 0.087	1028.70 ± 31.92	13.060	1306.0
agg-ROMMA	20.845 ± 0.234	4341.00 ± 100.30	661.652	66165.2	agg-ROMMA	3.580 ± 0.061	2317.30 ± 38.02	137.975	13797.5
ALMA _{0.9} (0.9)	20.990 ± 0.214	3571.05 ± 40.38	338.609	33860.9	ALMA _{0.9} (0.9)	3.518 ± 0.071	1011.00 ± 15.53	13.245	1324.5
PA-I	21.826 ± 0.294	4700.70 ± 47.80	4.296	429.6	PA-I	3.701 ± 0.081	2830.60 ± 45.87	3.732	373.2
PA-II	21.478 ± 0.287	3968.40 ± 51.16	4.536	453.6	PA-II	3.571 ± 0.083	3391.80 ± 51.94	4.719	471.9
DUOL	19.390 ± 0.227	3090.85 ± 38.91	10.121	1012.1	DUOL	2.771 ± 0.041	1008.80 ± 23.78	2.877	287.7

Observations from experimental results:

- DUOL achieves significantly smaller mistake rates than the other single-updating algorithms in all cases.
- DUOL results in sparser classifiers than the three aggressive online learning algorithms, and denser classifiers than the three non-aggressive algorithms.
- DUOL is overall as efficient as the other state-of-the-art online learning algorithms.

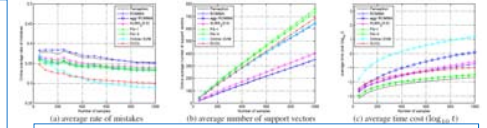


Figure 2: Evaluation on the german dataset. The data size is 1000 and the dimensionality is 24.

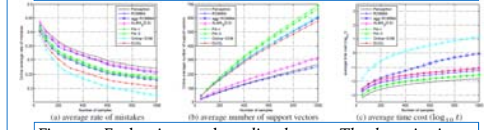


Figure 3: Evaluation on the splice dataset. The data size is 1000 and the dimensionality is 60.

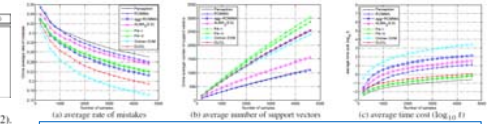


Figure 4: Evaluation on the spambase dataset. The data size is 4601 and the dimensionality is 57.

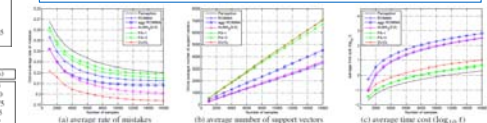


Figure 5: Evaluation on the a7a dataset. The data size is 16100 and the dimensionality is 123.

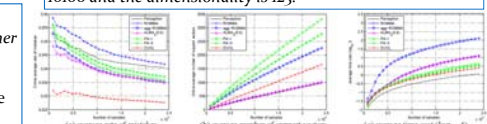


Figure 6: Evaluation on the w7a dataset. The data size is 24292 and the dimensionality is 300.

Conclusions

- A novel approach to online learning, which not only updates the weight of the newly added support vector, but also adjusts the weight of one existing support vector that seriously conflicts with the new support vector;
- Compared with a number of competing algorithms, the mistake bound can be significantly reduced by the proposed DUOL;
- Future work: DUOL for multi-class online learning and budget online learning.