1 Style 1.

Input: The set of positive samples for current batch, P_n; The set of unlabelled samples for current batch, U_n; Ensemble of classifiers on former batches, E_{n-1};
Output: Ensemble of classifiers on the current batch, E_n;
1: Extracting the set of reliable negative and/or positive samples T_n from U_n with help of P_n;
2: Training ensemble of classifiers E on $T_n \cup P_n$, with help of data in former batches;
3: $E_n = E_{n-1} cup E$;
4: Classifying samples in $U_n - T_n$ by E_n;
5: Deleting some weak classifiers in E_n so as to keep the capacity of E_n;
6: **return** E_n;

2 Style 2.

Algorithm 1 An example for format For & While Loop in Algorithm

1: **for** each $i \in [1, 9]$ **do**
2: initialize a tree T_i with only a leaf (the root);
3: $T = T \cup T_i$;
4: **end for**
5: **for all** c such that $c \in RecentMBatch(E_{n-1})$ **do**
6: $T = T \cup PosSample(c)$;
7: **end for**;
8: **for** $i = 1$; $i < n$; $i++$ **do**
9: // Your source here;
10: **end for**
11: **for** $i = 1$ to n **do**
12: // Your source here;
13: **end for**
14: // Reusing recent base classifiers.
15: **while** $(|E_n| \leq L_1) and (D \neq \phi)$ **do**
16: Selecting the most recent classifier c_i from D;
17: $D = D - c_i$;
18: $E_n = E_n + c_i$;
19: **end while**

3 Style 3.

Algorithm 2 Conjugate Gradient Algorithm with Dynamic Step-Size Control

Input: $f(x)$: objective funtion; x_0: initial solution; s: step size;
Output: optimal x^*

1: initial $g_0 = 0$ and $d_0 = 0$;
2: **repeat**
3: compute gradient directions $g_k = \bigtriangledown f(x_k)$;
4: compute Polak-Ribiere parameter $\beta_k = \frac{g_k^T(g_k - g_{k-1})}{\|g_{k-1}\|^2}$;
5: compute the conjugate directions $d_k = -g_k + \beta_k d_{k-1}$;
6: compute the step size $\alpha_k = s/ \parallel d_k \parallel_2$;
7: **until** $(f(x_k) > f(x_{k-1}))$

Algorithm 3 Test

Step 1.

Step 2.