Supplementary Material: Supervised Incremental Hashing

Bahadir Ozdemir[1]
ozdemir@cs.umd.edu
Mahyar Najibi[1]
najibi@cs.umd.edu
Larry S. Davis[2]
lsd@umiacs.umd.edu

[1] Department of Computer Science
University of Maryland
College Park, MD 20742

[2] Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742

1 Incremental Hashing

1.1 Training SVMs

The SVM problem can be described as an unconstrained regularized risk minimization problem as follows:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) := \frac{1}{2}\|\mathbf{w}\|^2 + C\,R(\mathbf{w}), \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^d$ denotes the weight vector to be learned, $\frac{1}{2}\|\mathbf{w}\|^2$ is a quadratic regularization term, $C > 0$ is a fixed regularization constant and $R : \mathbb{R}^d \to \mathbb{R}$ is a non-negative convex risk function *e.g.* hinge loss on training data, $R(\mathbf{w}) = \sum_{i=1}^{n} \max(0, 1 - y_i\mathbf{w}^\top\mathbf{x}_i)$.

The cutting plane method approximates the convex function $F(\mathbf{w})$ by a piecewise linear function $F_t(\mathbf{w})$ with t cutting planes. Let $\mathbf{w}_t$ and $\mathbf{w}_t^b$ be the *current solution* and the *best-so-far solution* of the OCAS method at iteration t, respectively. The stopping condition of the OCAS method is defined for a given tolerance parameter ε as $F(\mathbf{w}_t^b) - F_t(\mathbf{w}_t) \le \varepsilon$. Given in [1], assume that $\|\partial R(\mathbf{w})\| \le G$ for all $\mathbf{w} \in \mathbb{R}^d$, and $F(\mathbf{w}_t^b) - F_t(\mathbf{w}_t) = \varepsilon_t > 0$, then

$$\varepsilon_t - \varepsilon_{t+1} \ge \frac{\varepsilon_t}{2} \min\left(1, \frac{\varepsilon_t}{4C^2G^2}\right). \tag{2}$$

From (2), for any $\varepsilon_t > 4C^2G^2$, we say that $\varepsilon_{t+1} \le \frac{\varepsilon_t}{2}$; starting from a weight vector $\mathbf{w}_0$, we can reach at a level precision better than $4C^2G^2$ after at most $\log_2 \frac{F(\mathbf{w}_0)}{4C^2G^2}$ iterations [1]. Subsequently, we can find the remaining number of iterations by solving the following differential equation:

$$\varepsilon_{t+1} + \frac{\varepsilon_t^2}{8C^2G^2} - \varepsilon_t = 0. \tag{3}$$

Franc and Sonnenburg provide a solution to this equation in [1]. We need $\frac{8C^2G^2}{\varepsilon} - 2$ more iterations until convergence. The OCAS method is initialized its computation from the zero vector. Therefore, the total number of iterations of the OCAS method to solve an SVM problem is at most

$$\log_2 \frac{F(\mathbf{0})}{4C^2G^2} + \frac{8C^2G^2}{\varepsilon} - 2. \tag{4}$$

In Supervised Incremental Hashing, we use SVMs with hinge losses. Thus, $F(\mathbf{0})$ is equal to nC for a dataset with n images. The initial weight vector $\mathbf{w}_0$ has no effect on the number of iterations following a precision level better than $4C^2G^2$. However, considering an initial vector $\mathbf{w}_0$ in our incremental setting such that $F(\mathbf{w}_0) < nC$ *i.e.* having a better solution $\mathbf{w}_0$ than the zero vector, we can reduce the number of iterations until achieving such a precision level by at most

$$\log_2 \frac{F(\mathbf{0})}{4C^2G^2} - \log_2 \frac{F(\mathbf{w}_0)}{4C^2G^2} = \log_2 \frac{nC}{F(\mathbf{w}_0)}. \tag{5}$$

This concludes the claim in the main paper. A simulation of our incremental SVM approach is presented on a sample dataset in Figure 1.

2 Experiments

Precision-recall curves of all methods on all datasets are displayed in Figure 2 for 32-bit length codes.

2.1 Retrieval Performance Analysis for Dynamic Datasets

The mAP scores and training time are shown in Figures 3 and 4 for the three types of modifications as discussed in Section 2.2 of the paper on the MNIST and NUS-WIDE datasets, respectively. Note that the training time reported for the passive strategy shows the initial computation only.

References

[1] V. Franc and S. Sonnenburg. Optimized cutting plane algorithm for large-scale risk minimization. *JMLR*, 10:2157–2192, December 2009.

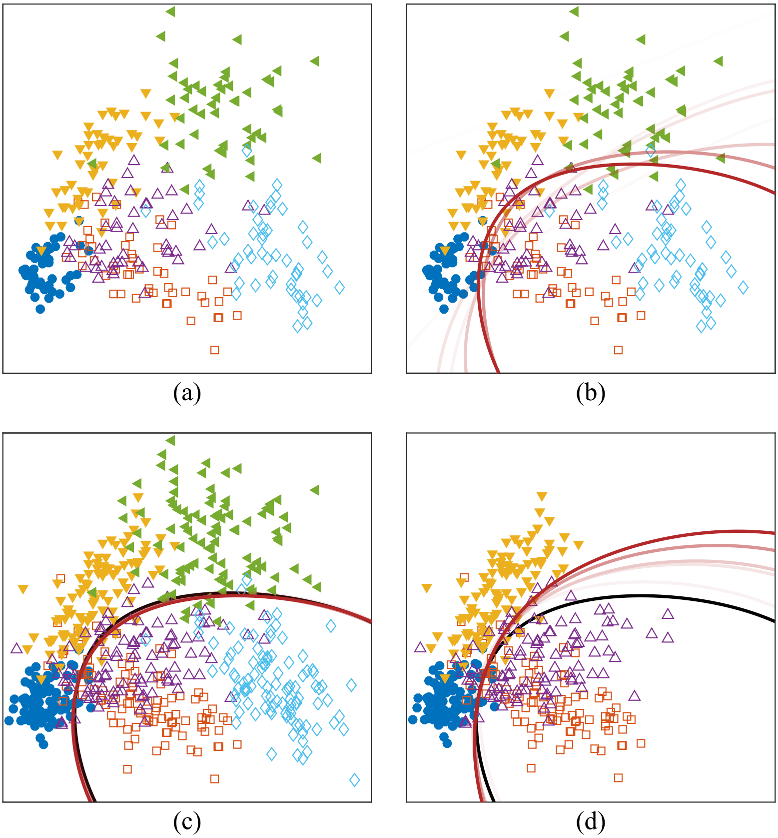

Figure 1: (a) A sample dataset of 300 data points from 6 classes represented by colors and shapes. Assignments of $+1$ and -1 are indicated by filled and empty shapes, respectively. (b) Each red line represents a hyperplane corresponding to the weight vector at each step of the OCAS method *i.e.* $(\mathbf{w}_t^b)^\top \mathbf{x} = 0$. Increasing transparency of the lines indicates earlier iterations of the execution *i.e.* smaller t. (c) The number of data points is increased to 600 by adding new points from the same distributions in (b). Red lines represent the hyperplanes when the OCAS method is initialized with the solution at (b) shown by a black line. (d) Two classes are deleted from (c). Red lines represent the hyperplanes when the OCAS method is initialized with the solution at (c) shown by a black line.

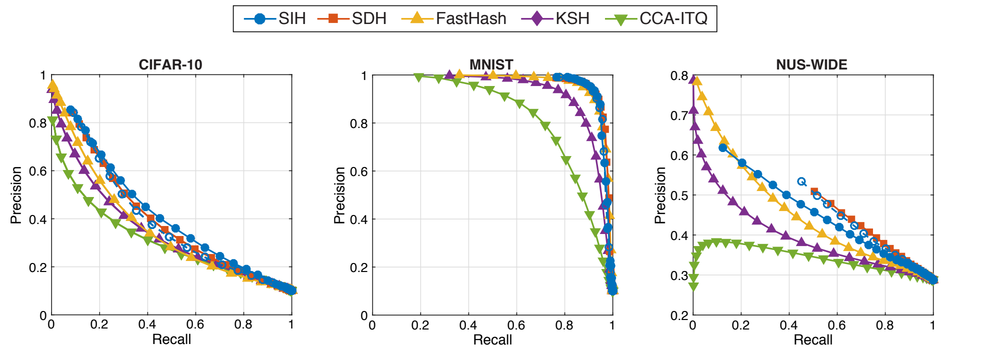

Figure 2: Our method (SIH) is compared with the state-of-the-art methods on, from left to right, the CIFAR-10, MNIST and NUS-WIDE datasets by precision-recall curves for 32-bit length hash codes. Dashed line represents SIH without imbalance penalty.

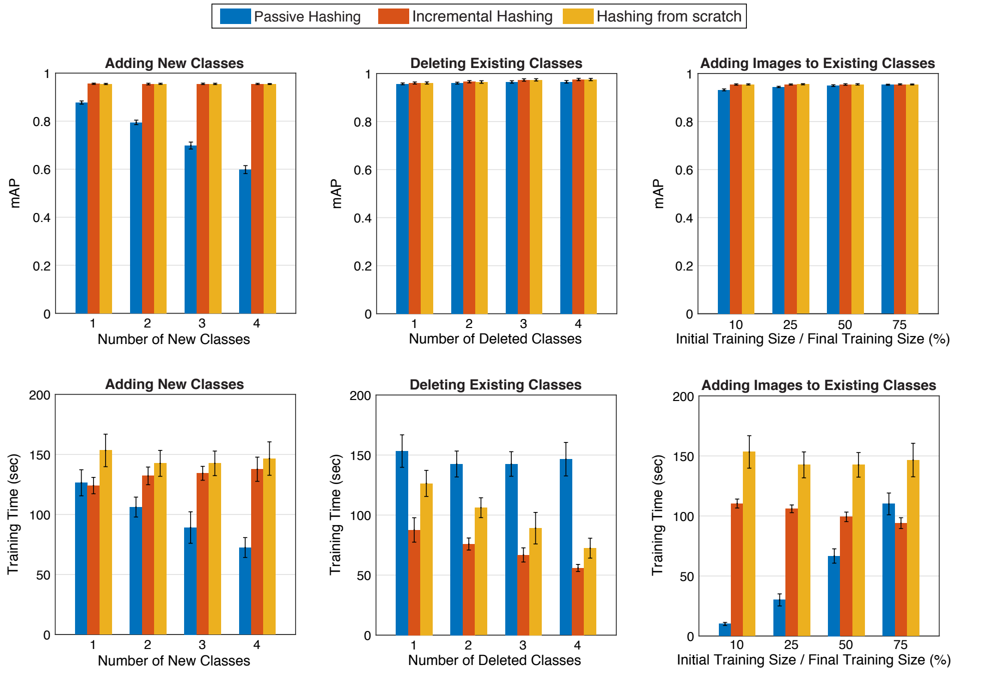

Figure 3: Incremental Hashing is compared with from-scratch and passive hashing for different types of modifications, from left to right, adding new classes, deleting existing classes and adding new images to existing classes on MNIST in terms of mean average precision (mAP) in the first row and in terms of training time in the second row at 32-bits.

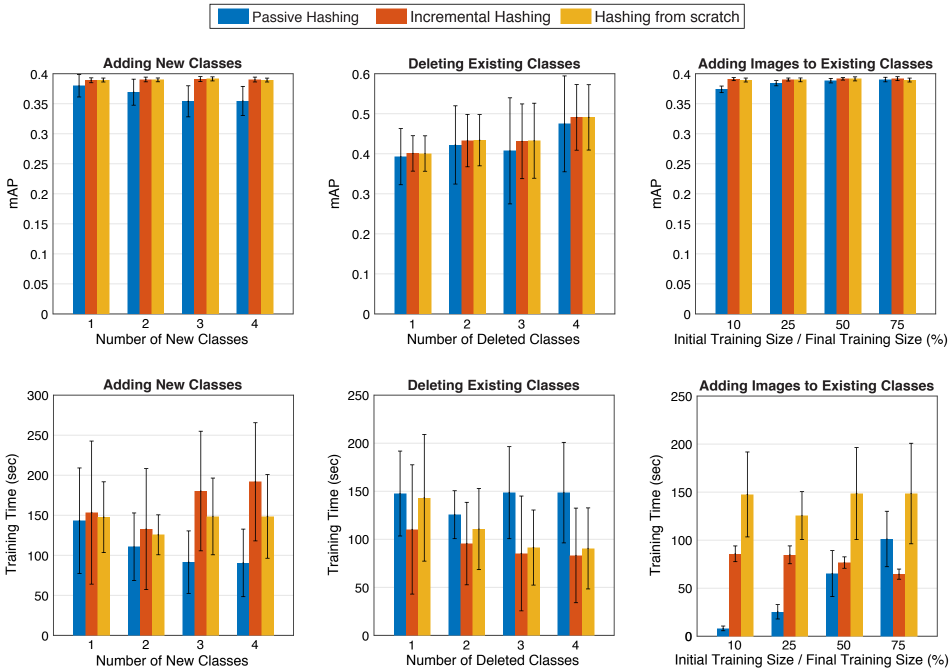

Figure 4: Incremental Hashing is compared with from-scratch and passive hashing for different types of modifications, from left to right, adding new classes, deleting existing classes and adding new images to existing classes on NUS-WIDE in terms of mean average precision (mAP) in the first row and in terms of training time in the second row at 32-bits.