Optimizing Circulant Support Vector Machines: the Exact Solution

Ramin Raziperchikolaei and **Miguel Á. Carreira-Perpiñán**, UC Merced

1 Abstract

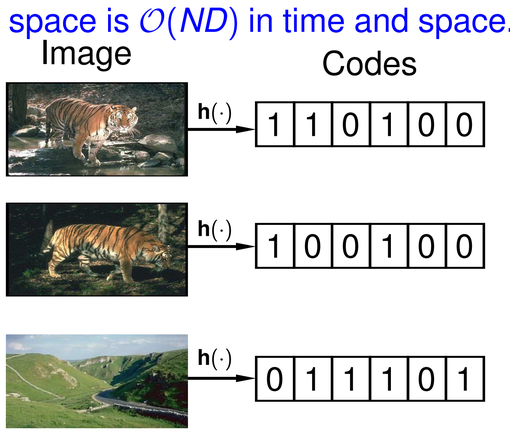
Binary hashing is an established approach for fast, approximate image search. The idea is to learn a hash function that maps a query image to a binary vector so that Hamming distances approximate image similarities. An important subproblem in binary hashing is to solve a set of independent classification problems, usually using support vector machines (SVMs). In this paper, we show that the hash function performs faster if we learn a set of circulant SVMs instead of the independent ones. Unlike the previously proposed algorithm that finds a suboptimal solution of the circulant SVMs, we show that the problem can be solved exactly and efficiently by casting it as a convex maximum margin classification problem on a modified dataset. We confirm experimentally that our approach solves the classification problem and the image search task better than the previous method.

Work supported by NSF award IIS–1423515

2 Binary hash functions for fast image retrieval

In *K* nearest neighbors problem, there are N training points in D-dimensional space (usually $D > 100$) $\mathbf{x}_i \in \mathbb{R}^D$, $i = 1, \dots, N$. The goal is to find the K nearest neighbors of a query $\mathbf{x}_q \in \mathbb{R}^D$. Exact search in the original space is $\mathcal{O}(ND)$ in time and space.

A binary hash function $\mathbf{h}$ takes as input a high-dimensional vector $\mathbf{x} \in \mathbb{R}^D$ and maps it to an L-bit vector $\mathbf{z} = \mathbf{h}(\mathbf{x}) \in \{0,1\}^L$ or $\mathbf{z} = \mathbf{h}(\mathbf{x}) \in \{-1,1\}^L$.
The main goal is preserving the neighborhood, i.e., assign (dis)similar codes to (dis)similar patterns.



In the binary space, time complexity is computed based on two operations:

- Time needed to generate the binary code for the query: mapping a D-dimensional vector to an L-bit vector takes $\mathcal{O}(LD)$.
- $\mathcal{O}(1)$ to search for similar codes using inverted index.

3 Learning the binary hash function

Most hashing papers try to minimize an affinity-based objective, which directly tries to preserve the original similarities in the binary space. With the Laplacian loss, the objective function has the following form:

$$\min_{\mathbf{h}} E(\mathbf{h}) = \sum_{i,j=1}^{N} w_{ij} \left\| \mathbf{h}(\mathbf{x}_i) - \mathbf{h}(\mathbf{x}_j) \right\|^2 \quad \text{s.t.} \quad \sum_{i=1}^{N} \mathbf{h}_l(\mathbf{x}_i) = 0, \quad \mathbf{h}(\mathbf{X})\mathbf{h}(\mathbf{X})^T = N\mathbf{I}$$

where $\mathbf{x}_i \in \mathbb{R}^D$ is the ith input data, $\mathbf{h}$ is the parameters of the hash function, y_{nm} is the ground-truth value of the pair of points $\mathbf{x}_n$ and $\mathbf{x}_m$ ($y_{nm} = 1$ for similar pairs and -1 for dissimilar pairs). In this paper, we consider the linear hash function $\mathbf{h}(\mathbf{x}_i) = \operatorname{sgn}(\mathbf{W}\mathbf{x}_i) \in \{-1,+1\}^L$ that maps each image into an L-bit binary code.

Optimizing $E(\mathbf{h})$ is difficult because $\mathbf{h}$ is discrete.

One popular way to solve the problem is to introduce the coordinates $\mathbf{Z} \in \{-1,1\}^{L \times N}$ (one L bit code per point), define the objective over $\mathbf{Z}$, and learn the codes and the hash function in separate steps:

- Over $\mathbf{Z}$: Learn the codes by alternating optimization over each bit.
- Over $\mathbf{h}$: Learn a binary classifier for each bit independently.

Learning the hash function $\mathbf{h}$ corresponds to solving L binary classification problems independently. We fit classifier l to the data $(\mathbf{X}, \mathbf{Z}_{:,i})$ for $l = 1, \dots, L$ ($\mathbf{X}_{D \times N} = (\mathbf{x}_1, \dots, \mathbf{x}_N) =$ images, $\mathbf{Z}_{:,i} = (\mathbf{z}_{:1}, \dots, \mathbf{z}_{:N}) \in \{-1,1\}^N =$ binary codes).

4 Hashing with a circulant weight matrix

A D-dim. vector $\mathbf{w} = (w_0, \dots, w_{D-1})$ is the basis for the $D \times D$ circulant matrix $\mathbf{W}$:

$$\mathbf{W} = \operatorname{circ}(\mathbf{w}) \equiv \begin{bmatrix} w_0 & w_{D-1} & \cdots & w_2 & w_1 \\ w_1 & w_0 & w_{D-1} & \cdots & w_2 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ w_{D-1} & w_{D-2} & \cdots & w_1 & w_0 \end{bmatrix}$$

For $L < D$ bits, we only need the first L rows of $\operatorname{circ}(\mathbf{w})$: $\operatorname{circ}(\mathbf{w})_L$.

For a query $\mathbf{x}_i$, generating the binary code $\operatorname{sgn}(\mathbf{W}\mathbf{x}_i)$ involves a matrix-vector multiplication. This can be computed faster when $\mathbf{W}$ is circulant:

	Space Complexity	Time Complexity
Linear function	$\mathcal{O}(LD)$	$\mathcal{O}(LD)$
Circulant function	$\mathcal{O}(D)$	$\min(\mathcal{O}(LD), \mathcal{O}(D \log D))$

The reason is that the Discrete Fourier Transform $\mathcal{F}(\cdot)$ can be computed in $\mathcal{O}(D \log D)$. The binary code is generated using DFT: $\mathbf{h}(\mathbf{x}) = \operatorname{sgn}(\mathbf{W}\mathbf{x}) = \operatorname{sgn}\left(\mathcal{F}^{-1}(\mathcal{F}(\mathbf{x}) \circ \mathcal{F}(\mathbf{w}))\right)$.

5 Learning circulant support vector machines

Consider the dataset $\mathbf{X} \in \mathbb{R}^{D \times N}$ and the labels $\mathbf{Z} \in \{-1,1\}^{L \times N}$.
We want to learn the circulant matrix $\mathbf{W} = \operatorname{circ}(\mathbf{w})_L \in \mathbb{R}^{L \times D}$ and the bias $\mathbf{b} \in \mathbb{R}^L$ that minimize the binary classification error.

Previous work: A previous work, Circulant binary embedding (Yu et al. 2014), learns the circulant matrix $\mathbf{W} \in \mathbb{R}^{L \times D}$ to solve the classification problem as follows:

1. They pad the label matrix $\mathbf{Z}$ with $D - L$ zero rows to make it $D \times N$.
2. They solve the classification problem in the frequency domain.
3. They pick the first L rows of the resulting $\mathbf{W}$.

The padding step makes this algorithm incorrect, except for $L = D$. For $L < D$, the resulting $\operatorname{circ}(\mathbf{w})_L$ is not the optimal solution. As we make the L smaller, the error becomes larger..

Circulant Support Vector Machines:
We consider the maximum margin formulation of the support vector machines (SVMs) and we propose a correct way to learn the optimal circulant matrix.
Consider $\mathbf{w}_l^T$ as the lth row of the matrix $\mathbf{W}$. The lth classification problem has the following form:

$$\min_{\mathbf{w}_l \in \mathbb{R}^D, b_l \in \mathbb{R}} \frac{1}{2} \|\mathbf{w}_l\|^2 + C \sum_{n=1}^{N} \xi_{ln} \quad \text{s.t.} \quad z_{ln}(\mathbf{w}_l{}^T \mathbf{x}_n + b_l) \geq 1 - \xi_{ln},\ \xi_{ln} \geq 0,\ n = 1, \dots, N$$

The L problems are coupled because of $\mathbf{W} = \operatorname{circ}(\mathbf{w})_L$.

We can write row l of $\mathbf{W}$ as $\mathbf{w}_l^T = \mathbf{w}^T \mathbf{P}_l$, where $\mathbf{P}_l \in \mathbb{R}^{D \times D}$ is a permutation matrix. The SVM formulation of the lth classification problem becomes:

$$\min_{\mathbf{w} \in \mathbb{R}^D, b_l \in \mathbb{R}} \frac{1}{2} \|\mathbf{w}^T \mathbf{P}_l\|^2 + C \sum_{n=1}^{N} \xi_{ln} \quad \text{s.t.} \quad z_{ln}(\mathbf{w}^T \mathbf{P}_l \mathbf{x}_n + b_l) \geq 1 - \xi_{ln},\ \xi_{ln} \geq 0,\ n = 1, \dots, N.$$

Since $\mathbf{P}_l^T \mathbf{P}_l = \mathbf{I}$, $\|\mathbf{w}^T \mathbf{P}_l\|^2 = \|\mathbf{w}\|^2$, all the L classification problems have the same margin term.
Let us define $\mathbf{t}_{ln} = \mathbf{P}_l \mathbf{x}_n \in \mathbb{R}^D$ and rewrite the objective function:

$$\min_{\mathbf{w} \in \mathbb{R}^D, b_l \in \mathbb{R}} \|\mathbf{w}\|^2 + \frac{2C}{L} \sum_{l=1}^{L} \sum_{n=1}^{N} \xi_{ln} \quad \text{s.t.} \quad \begin{cases} z_{ln}([\mathbf{w}; \mathbf{b}]^T [\mathbf{t}_{ln}; \mathbf{e}_l]) \geq 1 - \xi_{ln},\ \xi_{ln} \geq 0 \\ n = 1, \dots, N,\ l = 1, \cdots, L. \end{cases}$$

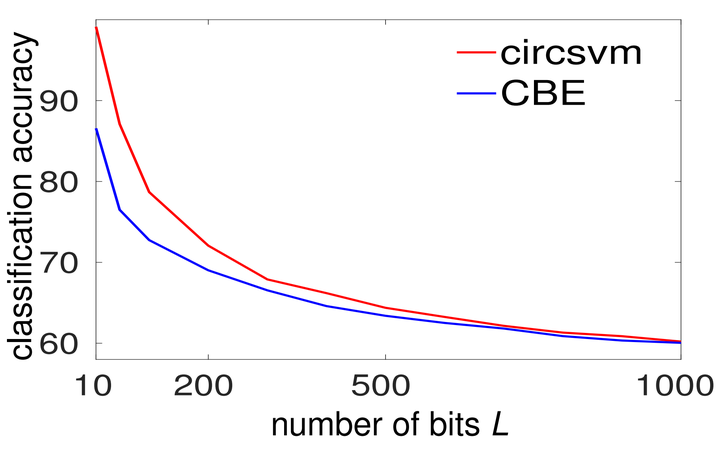
where $\mathbf{e}_l \in \mathbb{R}^L$ has 1 in the lth element and zeros everywhere else.
This is an SVM problem, with NL inputs $\mathbf{y}_{ln} = [\mathbf{t}_{ln}; \mathbf{e}_l]$ and labels z_{ln}.

Advantages of the circulant support vector machines:

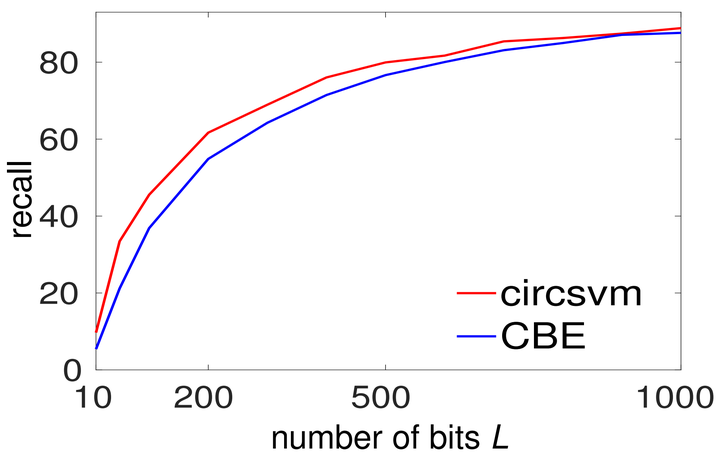
- It always returns the optimal solution, even for the case of $L < D$.
- It is a convex quadratic program with a unique solution.
- There are libraries available that solve SVM problems for a large number of points in a few seconds.
- Our circulant SVM performs better than CBE in retrieval results.

6 Experiments

CIFAR-10 dataset contains 60 000 32×32 color images in 10 classes. We randomly select 58 000/2 000 as the training/test set. Each image is represented by a $D = 4\,096$-dim. VGG feature vector: the output of the last layer of the VGG network. We use the L-bit codes generated by a hashing method (ITQ) as the labels.



We report the average accuracy of the L classification problems. For smaller number of bits, CBE finds a suboptimal solution. Our proposed method (Circulant SVM) always finds the optimal solution and gives a better classification accuracy.



We use the hash functions of the previous experiments in the hashing setting. We report recall for different number of bits. circsvm outperforms CBE. The improvement is more clear for smaller number of bits where CBE is unable to find the optimal solution.