Saliency Detection Based Hashing

Zhengwei Shen, Supervisor, USTB, Qian Chen, Senior , USTB
Zhe Chen, Qing Sun, Wenwen Bao, Yongqi Yuan, Yijun Mo, Member, USTB

Abstract—Hashing has recently been widely used in data dimension reduction. However, learning compact codes with good performance is still a challenge. For our real-world data, we can find some low-dimensional features from their high-dimensional structures to take the place of their high-dimensional features to save the data storage space. Most existing hashing methods try to map data points into some low-dimensional certain projected dimensions with a random projection matrix. We recently get an idea from the saliency detection. For a big dataset, maybe we can find some common ground between some of the data. And then we divide the dataset into some parts in which data have the same common ground. And then we can decompose the big dataset into two parts, one low-rank part and one sparse part. And use the information to generate the hashing codes.(Further improvement is needed.)

Index Terms—Hashing, saliency detection, low-rank matrix, sparse matrix.

I. Introduction

WHat we want to do is to find out a model to construct the relationship between Hash and saliency map. In addition, we hope to have a fast and accurate algorithm to solve this problem. Regarding the construction of the hash model, it can be divided into two mainstream models. One is the local sensitive hash (LSH)(we call it one-step hash) method formed by the random projection matrix, and the other is the spectral hash (SH)(we call it two-step hash) method formed by the Laplacian matrix in graph theory. Given instances data $X = \{x_1, ..., x_n\} \in R^{p\times n}$, LSH model utilize the random projection matrix W to re-arrange the feature of data, then utilize the binary thresholding to get the hash codes. LSH model can be written as follows

$$\min_{B,W} ||B - sgn(WX^T)||_2^2 \tag{1}$$

In general, there is a constraint on W that W is a row of orthogonal matrix, i.e., $W^TW = I$. In actual processing, W is generally initialized as a random matrix generated by Gaussian distribution.

On the contrary, SH model utilize the feature representation model in Machine Learning. SH model can be written as follows:

$$\min_{B} trB^TLB$$
$$s.t. \quad B^T1 = 0, B^TB = I, B \in \{-1, 1\}^{p\times n}$$

where L is the Laplacian matrix of a graph composed of instance data. There, $B^T1 = 0$ is called Balanced Constraint and $B^TB = I$ is called Uncorrelated Constraint. $B \in \{-1, 1\}^{p\times n}$ ensure the discreteness and validity of Hash codes.

In fact, LSH and SH have a certain connection, they actually use the same thinking method with different tools. LSH hopes to combine the features of the instances out of order through an random arrangement, and then generate Hash codes. In statistics, the data generated after a certain random combination of different data is the same should be a small probability event. When this event occurs, it means that the assumption is not true, the original data is the same. I think SH achieves the purpose of this random arrangement through two processes, which is the processing from data to graph and from graph to Laplace matrix.

As for the nature of Hash that both hopes to have, they are actually the same. The first is about the binary constraint, one is to directly *sgn* the results of the arrangement, and the other is to pass a discrete constraint. Regarding the item of irrelevance, one is the uncorrelated in the process of changing by constraint, i.e., $W^TW = I$, and the other is the uncorrelated of the result of changing by constraint. As for the balance constraint, it is actually contained in the model in the LSH. The model generated by random projection, imposing a balance constraint will destroy the randomness of the projection.

Regarding saliency detection, in fact, superpixels that are difficult to be well interpreted by other locations in the image are detected as saliency content by a certain model. Given a image X, it can be divide into n superpixel by simple linear iterative clustering (SLIC), then utilize the feature exaction method get the image feature matrix $F = \{f_1, ..., f_n\} \in R^{l\times n}$. The simplest model that uses interpretation is the model of rPCA for significance detection, which can be written as follow:

$$\min_{L,S} ||L||_* + ||S||_1$$
$$s.t. \quad F = L + S$$

where L is a low-rank matrix which can be seen as the background (non-saliency) and S is a sparse matrix which is the foreground (saliency). Regarding the problem of saliency detection, it is to detect a minimum error model under the mutual representation of super-pixels in the optimization problem.

First of all, there is a certain similarity between Hash and saliency detection, both of which are about the mutual representation between instances. In Hash, if an image can be well represented by several other images, it means that there is a certain degree of similarity between these images. The difficulty of linking Hash and saliency detection lies in how to deal with problems in two dimensions. Hash is an established relationship between images and images,

TABLE I

symbols	meanings
$X = [x_1, x_2, \cdots, x_n] \in \mathbf{R}^{p\times n}$	Feature matrix of data
$L = [l_1, l_2, \cdots, l_n] \in \mathbf{R}^{p\times n}$	Low rank component of X
$S = [s_1, s_2, \cdots, s_n] \in \mathbf{R}^{p\times n}$	Sparse component of X
$Xt = [x_{t1}, x_{t2}, \cdots, x_{tm}] \in \mathbf{R}^{p\times m}$	Test data matrix
$B = [b_1, b_2, \cdots, b_n] \in \mathbf{R}^{p\times n}$	Hash codes of X
$Bt = [b_{t1}, b_{t2}, \cdots, b_{tm}] \in \mathbf{R}^{p\times m}$	Hash codes of Xt
$Y = [y_1, y_2, \cdots, y_n] \in \mathbf{R}^{p\times n}$	Label information of X
$Yt = [y_{t1}, y_{t2}, \cdots, y_{tm}] \in \mathbf{R}^{p\times m}$	Label information of Xt

whose dimensions are equivalent to a ”three-dimensional” problem. Saliency detection is a kind of connection established within an image, and its temperature is equivalent to a ”two-dimensional” problem. How to transform ”two-dimensional” to ”three-dimensional” problem, or ”three-dimensional” to ”two-dimensional” problem is the difficulty of establishing this model.

II. Related Works

A. About Our Model

Suppose that we have n samples $X = \{x_i\}_{i=1}^{n} \in \mathbf{R}^{p\times n}$. And at same time we have m test samples $Xt = \{X_{ti}\}_{i=1}^{n} \in \mathbf{R}^{p\times m}$. We aim to learn a set of binary codes $B = \{b_i\}_{i=1}^{n} \in \{-1,1\}^{l\times n}$ to well preserve their spatial structure, where the i^{th} column b_i is the l-bits codes for x_i. To take advantage of the label information, we're going to introduce the ground truth label matrix $Y = \{y_i\}_{i=1}^{n} \in \mathbf{R}^{c\times n}$. We also introduce the test data label matrix $Yt = \{y_{ti}\}_{i=1}^{m} \in \mathbf{R}^{c\times m}$. And at the same time we want to try a hash model with the idea we got from saliency detection. So the model can first written as:

$$\min_{W,B,L,S} \quad \|Y - W^TB\|_F^2 + \lambda_1\|B^TB - S^TS\|_F^2 + \lambda_2\|L\|_* + \lambda_3\|S\|_1 \tag{1}$$

$$s.t. \quad L + S = X, \;\; B \in \{-1,1\}^{l\times n}, \;\; B1_n = 0_l, \;\; BB^T = nI_l$$

The W is of a size $l\times c$, and W^T is of a size $c\times l$. Here we also have $y_i = [w_1^Tb_i, \cdots, w_C^Tb_i]^T$. w_k is the classification vector for class $k(k = 1, \cdots, c)$ and $y_i \in \mathbf{R}^{c\times 1}$ is the label vector, of which the maximum item indicates the assigned class of x_i. F is the feature matrix and we want to decompose it as two parts with one low-rank part L and the other one sparse part S. The constraints here are balance constraint and decorrelation constraint.

For easy reference, TABLE I is a list of key symbols used in this paper: A capital notation will be used for a matrix.

III. Introduce Auxiliary Variables

Here I want to rewrite the model as:

$$\min_{W,B,Z,L,S} \quad \|Y - W^TB\|_F^2 + \lambda_1\|B^TZ - S^TS\|_F^2 + \alpha\|B - Z\|_F^2 + \lambda_2\|L\|_* + \lambda_3\|S\|_1 \tag{2}$$

$$s.t. \begin{cases} B \in \{-1,+1\}^{l\times n}, \\ Z \in R^{l\times n}, Z1_n = 0_l, ZZ^T = nI_l \\ X = L + S \end{cases}$$

With the constraints, the model can be rewritten as:

$$\min_{W,B,Z,L,S} \begin{cases} \|Y - W^TB\|_F^2 + \lambda_1\|B^TZ - S^TS\|_F^2 \\ + \alpha\|B - Z\|_F^2 + \lambda_2\|L\|_* + \lambda_3\|S\|_1 \\ + < Y_1, X - L - S > + \dfrac{\mu}{2}\|X - L - S\|_F^2 \end{cases}$$

A. Updating W with others fixed:

With other variables fixed, we make only one variable W here. The updating process is:

$$\min_{W} \quad \|Y - W^TB\|_F^2$$
$$\min_{W} \quad \|B^TW - Y^T\|_F^2$$

The solution is

$$W = (BB^T)^{-1}BY^T$$

B. Updating B with others fixed:

The updating process is:

$$\min_{B} \quad \|Y - W^TB\|_F^2 + \lambda_1\|B^TZ - S^TS\| + \alpha\|B - Z\|_F^2$$
$$s.t. \quad B \in \{-1,+1\}^{l\times n}$$

This is equivalent to the optimization problem:

$$\max_{B} \quad tr(B^T(WY + \lambda_1ZS^TS + \alpha Z))$$
$$s.t. \quad B \in \{-1,+1\}^{l\times n}$$

It has a closed form solution:

$$B = sgn(WY + \lambda_1ZS^TS + \alpha Z)$$

C. Updating Z with others fixed:

The updating process is:

$$\min_{Z} \quad \lambda_1\|B^TZ - S^TS\|_F^2 + \alpha\|B - Z\|_F^2$$
$$s.t. \quad Z \in \mathbf{R}^{l\times n}, Z1_n = 0_l, ZZ^T = nI_l$$

It can be further reduced to:

$$\max_{Z} \quad tr(Z^T(\lambda_1BS^TS + \alpha B))$$
$$s.t. \quad Z \in \mathbf{R}^{l\times n}, Z1_n = 0_l, ZZ^T = nI_l$$

Here we introduce two variables E and J, let $E = \lambda_1BS^TS + \alpha B$ and $J = I_N - \frac{1}{N}1_n1_n^T$:

$$JE^T = U\Sigma V^T = \Sigma_{k=1}^{K'}\sigma_ku_kv_k^T$$
$$U = [u_1, u_2, \cdots, u_{K'}] \quad and \quad V = [v_1, v_2, \cdots, v_{K'}]$$

Note that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{K'} > 0$. Then, by employing a Gram-Schmidt process one can easily construct matrices

$$\bar{U} \in \mathbf{R}^{n\times(L-K')} \quad and \quad \bar{V} \in \mathbf{R}^{L\times(L-K')}$$

$$s.t. \quad \begin{cases} \bar{U}^T\bar{U} = I_{L-K'}, \; [U \; 1_n]^T\bar{U} = 0_{(K'+1)\times(L-K')} \\ \bar{V}^T\bar{V} = I_{L-K'}, \; V^T\bar{V} = 0_{K'\times(L-K')} \end{cases}$$

Here

$$EJE^T = [V\ \bar{V}] \begin{bmatrix} \Sigma^2 & 0 \\ 0 & 0 \end{bmatrix} [V\ \bar{V}]^T$$

so we get the V, $\bar{V}$ and Σ, and then immediately leads to $U = JE^T V \Sigma^{-1}$. The matrix $\bar{U}$ is set to a random matrix followed by Gram-Schmidt process. It can be seen that Z is uniquely optimal when L=K’, which means JE^T is full column rank. (Actually I guess JE^T is full column rank in high probability.) So we have a closed form for Z:

$$Z = \sqrt{N}[V\ \bar{V}][U\ \bar{U}]^T$$

D. Updating L with others fixed:

The process is:

$$\begin{aligned} \min_L \quad & \lambda_2 \|L\|_* + < Y_1, X - L - S > \\ & + \frac{\mu}{2}\|X - L - S\|_F^2 \end{aligned} \tag{3}$$

$$\min_L \quad \frac{\lambda_2}{\mu}\|L\|_* + \frac{1}{2}\|L - (X - S + \frac{Y_1}{\mu})\|_F^2 \tag{4}$$

The solution is:

$$L = UT_{\frac{\lambda_2}{\mu}}(S')V^T$$

where $US'V^T = SVD(X - S + \frac{Y_1}{\mu})$.

E. Updating S with others fixed:

The process is:

$$\begin{aligned} \min_S \quad & \lambda_1 \|B^T Z - S^T S\|_F^2 + \lambda_3 \|S\|_1 \\ & + < Y_1, X - L - S > + \frac{\mu}{2}\|X - L - S\|_F^2 \end{aligned} \tag{5}$$

$$\begin{aligned} \min_S \quad & \lambda_1 \|B^T Z - S^T S\|_F^2 + \lambda_3 \|S\|_1 \\ & + \frac{\mu}{2}\|S - (X - L + \frac{Y_1}{\mu})\|_F^2 \end{aligned} \tag{6}$$

So the solution is:

$$S = T_{\frac{\lambda_3}{\mu}}(F - L + M + 2\frac{Y_1}{\mu})$$

IV. Experiments

```
clear; close
dataset=’cifar_10_gist.mat’;
load([’testbed\’,dataset]);
traindata = double(traindata);
testdata = double(testdata);


traingnd = traingnd - 1;
testgnd = testgnd - 1;


X_all = traindata’;
X_selected=X_all(:,15001:30000);
Y_all = double(traingnd)’;
Y_selected=Y_all(:,15001:30000);
Xt_all=testdata’;
Yt_all=double(testgnd)’;

Xt_selected=Xt_all(:,1:500);
Yt_selected=Yt_all(:,1:500);


[W, B_model, L, S , Y, Y1, a1] = RobustPCA(X_selected, Y_selected

[B_normal, W1]=leastsquare(X_selected);


A=(X_selected*X_selected’)^-1*X_selected*B_model’;

B_model=(B_model+1)/2;
Bt_model=(sign(A’*Xt_selected)+1)/2;

Bc_model=compactbit(B_model’)’;
Btc_model=compactbit(Bt_model’)’;
Dh1=hammingDist(Bc_model’,Btc_model’);
[~,HammingRank1]=sort((Dh1’),1);
ap1=cal_map(Y_selected’,Yt_selected’, Dh1);


A=(X_selected*X_selected’)^-1*X_selected*B_normal’;

B_normal=(B_normal+1)/2;
Bt_normal=(sign(A’*Xt_selected)+1)/2;
Bc_normal=compactbit(B_normal’)’;
Btc_normal=compactbit(Bt_normal’)’;
Dh2=hammingDist(Bc_normal’,Btc_normal’) ;
[~,HammingRank2]=sort((Dh2’),1);
ap2=cal_map(Y_selected’,Yt_selected’, Dh2);
```

Appendix A
RPCA Problem

For a RPCA problem, given a data matrix $X \in \mathbf{R}^{p \times n}$, and transform it to a related feature matrix $F \in \mathbf{R}^{D \times n}$ find L and S that solve the problem:

$$\begin{aligned} \min_{L,S} \quad & rank(L) + \|S\|_0 \\ s.t. \quad & F = L + S \end{aligned}$$

Reformulate it as follows:

$$\begin{aligned} \min_{L,S} \quad & \|L\|_* + \|S\|_1 \\ s.t. \quad & F = L + S \end{aligned}$$

For an $m \times n$ matrix M with SVD $US'V^T$:

$$UT_\epsilon(S')V^T = \arg\min_X \quad \epsilon\|X\|_* + \frac{1}{2}\|X - M\|_F^2$$

$$T_\epsilon(M) = \arg\min_X \quad \epsilon\|X\|_1 + \frac{1}{2}\|X - M\|_F^2$$

where the $T_\epsilon(M)$ is the soft threshold operator:

$$T_\epsilon(M) = \begin{cases} M-\epsilon, & M > \epsilon \\ M+\epsilon, & M < -\epsilon \\ \quad 0 \quad, & otherwise \end{cases}$$

and S' is from $SVD(M) = US'V^T$. Consider the problem:

$$\min_X \quad f(X)$$
$$s.t. \quad c_j(X) = 0, j = 1, \cdots, m$$

By using the Augmented Lagrange Multipliers Method:
1.Initialize Λ, $\mu > 0$, $\rho \geq 0$
(Repeat until convergence:)
2.Compute $X = \arg\min_X \quad L(X)$ where

$$L(X) = f(X) + < \Lambda, C(X) > + \frac{\mu}{2}\|C(X)\|_F^2$$

3.Update $\Lambda = \Lambda + \mu C(X)$
4.Update $\mu = \rho\mu$

Appendix B
ALM-RPCA

With ALM, RPCA problem is reformulated as:

$$\min_{L,S} \quad \|L\|_* + \lambda\|S\|_1 + < Y, F-L-S > + \frac{\mu}{2}\|F-L-S\|_F^2$$

By using the Alternating Direction Method(ADM), we have:

A. Updating S with L fixed:

$$\begin{aligned} \min_S \quad & \lambda\|S\|_1 + < Y, F-L-S > \\ & + \frac{\mu}{2}\|F-L-S\|_F^2 \\ \min_S \quad & \frac{\lambda}{\mu}\|S\|_1 + tr(\frac{Y^T}{\mu}(F-L-S)) \\ & + \frac{1}{2}\|F-L-S\|_F^2 + \frac{1}{2}\|\frac{Y}{\mu}\|_F^2 \\ \min_S \quad & \frac{\lambda}{\mu}\|S\|_1 + \frac{1}{2}\|S-(F-L+\frac{Y}{\mu})\|_F^2 \end{aligned} \tag{7}$$

So here we have the solution:

$$S = T_{\frac{\lambda}{\mu}(F-L+\frac{Y}{\mu})} \tag{8}$$

B. Updating L with S fixed:

$$\begin{aligned} \min_L \quad & \|L\|_* + < Y, F-L-S > \\ & + \frac{\mu}{2}\|F-L-S\|_F^2 \\ \min_L \quad & \frac{1}{\mu}\|L\|_* + tr(\frac{Y^T}{\mu}(F-L-S)) \\ & + \frac{1}{2}\|F-L-S\|_F^2 + \frac{1}{2}\|\frac{Y}{\mu}\|_F^2 \\ \min_L \quad & \frac{1}{\mu}\|L\|_* + \frac{1}{2}\|L-(F-S+\frac{Y}{\mu})\|_F^2 \end{aligned} \tag{9}$$

So here we have the solution:

$$L = UT_{\frac{1}{\mu}}(S')V^T \tag{10}$$

where S' is from $SVD(F-S+\frac{Y}{\mu}) = US'V^T$. And every time we update:

$$Y = Y + \mu(F-L-S) \tag{11}$$

$$\mu = \rho\mu \tag{12}$$

until convergence.Typical initialization:

1.$Y = \frac{sgn(F)}{max(\|F\|_2, \frac{\|F\|_\infty}{\lambda})}$, $\|F\|_2$ is spectral norm, largest singular value of elements of F, and $\|F\|_\infty$ is the largest absolute value of elements of F.
2.$\mu = 1.25\|F\|_2$.
3.$\rho = 1.5$.
4.$\lambda = 1/\sqrt{max(D,n)}$ for $D \times n$ matrix F.