Co-Regularized Hashing for Multimodal Data

Yi Zhen,Dit-Yan Yeung-NIPS2012

Yunfei Wang

Department of Computer Science & Technology
Huazhong University of Science & Technology

April 1, 2013

Table of contents

Objective Function I

Notations

Two data sets from two modalities:$\{x_i \in \mathcal{X}\}_{i=1}^{I}$,$\{y_j \in \mathcal{Y}_{j=1}^{J}\}$.
N inter-modality pairs: $\Theta = \{(x_{a1}, y_{b1}), \cdots, (x_{aN}, y_{bN})\}$.
Pair label $s_n = 1$(x_{an} and y_{bn} are similar);$s_n = 0$(otherwise).

Two linear hash functions for each bit of hash codes:

$$f(x) = sgn(w_x^T x)$$

$$g(y) = sgn(w_y^T y)$$

where $sgn(\cdot)$ denotes the sign function,w_x and w_y are projection vectors mapping similar points to the same hash bins and dissimilar points to different bins.

Objective Function II

Intra-modality loss terms:

$$\ell_i^x = [1 - f(x_i)(w_x^T x_i)]_+ = [1 - |w_x^T x_i|]_+$$

$$\ell_j^y = [1 - g(y_j)(w_y^T y_j)]_+ = [1 - |w_y^T y_j|]_+$$

where $[a]_+$ equals a if $a \geq 0$ and 0 otherwise.

Intra-modality loss terms:

$$\ell_n^* = s_n d_n^2 + (1 - s_n)\tau(d_n)$$

where $d_n = w_x^T x_{an} - w_y^T y_{bn}$,$\tau(d)$ requires the similar inter-modality points have small distance after projection and the dissimilar ones have large distance.

Objective Function III

The SCISD(smoothly clipped inverted squared deviation) function:

$$\tau(d) = \begin{cases} -\frac{1}{2}d^2 + \frac{a\lambda^2}{2}, & |d| \le \lambda \\ \frac{d^2 - 2a\lambda|d| + a^2\lambda^2}{2(a-1)}, & \lambda < |d| \le a\lambda \\ 0, & a\lambda < |d| \end{cases}$$

where a and λ are user-specified parameters.SCISD function penalizes projection vectors resulting in small distance between dissimilar points after projection.

Express SCISD function as a difference of two convex functions:$\tau(d) = \tau_1(d) - \tau_2(d)$ where

$$\tau_1(d) = \begin{cases} 0, & |d| \le \lambda \\ \frac{ad^2 - 2a\lambda|d| + a\lambda^2}{2(a-1)}, & \lambda < |d| \le a\lambda \\ \frac{1}{2}d^2 - \frac{a\lambda^2}{2}, & a\lambda < |d| \end{cases} \qquad \tau_2(d) = \frac{1}{2}d^2 - \frac{a\lambda^2}{2}$$

Objective Function IV

The final objective function:

$$\mathcal{O} = \frac{1}{I}\sum_{i=1}^{I} \ell_i^x + \frac{1}{J}\sum_{j=1}^{J} \ell_j^y + \gamma \sum_{n=1}^{N} \omega_n \ell_n^* + \frac{\lambda_x}{2}\|w_x\|^2 + \frac{\lambda_y}{2}\|w_y\|^2$$

$$\{w_x^*, w_y^*\} = \min_{w_x, w_y} \mathcal{O}$$

where $\sum_{n=1}^{N} \omega_n = 1$.

Optimization I

Concave convex procedure(CCCP)

Given an objective function $p(x) - q(x)$ where $p(x)$ and $q(x)$ are convex,CCCP works iteratively:

- Initialize x to x^0 randomly
- Minimize the following convex upper bound of $p(x) - q(x)$ at location x^t:

$$p(x) - (q(x^t) + \partial_x q(x^t)(x - x^t))$$

- Obtain x^{t+1} using convex optimization solver.

The solution sequence $\{x^t\}$ found by CCCP is guaranteed to reach a local minimum.

Optimization II

Objective function is nonconvex with respect to w_x and w_y,but we optimize it with respect to w_x and w_y alternatively.
Taking w_x for example,we remove the irrelevant terms and get the following objective:

$$F(w_x) = \frac{1}{I}\sum_{i=1}^{I} \ell_i^x + \gamma \sum_{n=1}^{N} \omega_n \ell_n^* + \frac{\lambda_x}{2}\|w_x\|^2$$

where

$$\ell_i^x = \begin{cases} 0, & |w_x^T x_i| \geq 1 \\ 1 - w_x^T x_i, & 0 \leq w_x^T x_i < 1 \\ 1 + w_x^T x_i, & -1 < w_x^T x_i < 0 \end{cases}$$

$$\ell_n^* = s_n d_n^2 + (1 - s_n)\tau_1(d_n) - (1 - s_n)\tau_2(d_n)$$

Optimization III

Setting

$$p(w_x) = \frac{1}{I}\sum_{i=1}^{I}\ell_i^x + \gamma\sum_{n=1}^{N}\omega_n(s_n d_n^2 + (1-s_n)\tau_1(d_n)) + \frac{\lambda_x}{2}\|w_x\|^2$$

$$q(w_x) = \gamma\sum_{n=1}^{N}\omega_n(1-s_n)\tau_2(d_n)$$

where $d_n = w_x^T x_{an} - w_y^T y_{bn}$. Then $F(w_x) = p(w_x) - q(w_x)$.
First derivative of $q(w_x)$ at w_x^t:

$$\frac{\partial q(w_x)}{\partial w_x^t} = \gamma\sum_{n=1}^{N}\omega_n(1-s_n)d_n^t\frac{\partial d_n}{\partial w_x^t} = \gamma\sum_{n=1}^{N}\omega_n(1-s_n)d_n^t x_{an}^T$$

Optimization IV

Upper bound of $F(w_x)$ with respect to w_x based on CCCP:

$$\mathcal{O}_x = \frac{\lambda_x \|w_x\|^2}{2} + \gamma \sum_{n=1}^{N} \omega_n (s_n d_n^2 + (1 - s_n)\zeta_n^x) + \frac{1}{I} \sum_{i=1}^{I} \ell_i^x$$

where

$\zeta_n^x = \tau_1(d_n) - \tau_2(d_n^t) - d_n^t x_{an}^T (w_x - w_x^t)$, $d_n^t = (w_x^t)^T x_{an} - w_y^T y_{bn}$, w_x^t is the value of w_x at tth iteration.

Solve the optimization problem using gradient-based method:

$$\frac{\partial \mathcal{O}_x}{\partial w_x} = \lambda_x w_x + 2\gamma \sum_{n=1}^{N} \omega_n s_n d_n x_{an} + \gamma \sum_{n=1}^{N} \omega_n \mu_n^x - \frac{1}{I} \sum_{i=1}^{I} \pi_i^x$$

where $\mu_n^x = (1 - s_n)(\frac{\partial \tau_1}{\partial d_n} - d_n^t) x_{an}$.

Optimization V

$$\frac{\partial \tau_1}{\partial d_n} = \begin{cases} 0, & |d_n| \leq \lambda \\ \frac{a d_n - 2a\lambda sgn(d_n)}{a-1}, & \lambda < |d_n| \leq a\lambda \\ d_n, & a\lambda < |d_n| \end{cases} \quad \pi_i^x \begin{cases} 0, & |w_x^T x_i| \geq 1 \\ sgn(w_x^T x_i) x_i, & |w_x^T x_i| < 1 \end{cases}$$

Similarly,the objective function for the optimization problem with respect to w_y at the tth iteration of CCCP is:

$$\mathcal{O}_y = \frac{\lambda_y \|w_y\|^2}{2} + \gamma \sum_{n=1}^{N} \omega_n (s_n d_n^2 + (1 - s_n)\zeta_n^y) + \frac{1}{J} \sum_{j=1}^{J} \ell_j^y$$

where
$\zeta_n^y = \tau_1(d_n) - \tau_2(d_n^t) + d_n^t y_{bn}^T (w_y - w_y^t)$,$d_n^t = w_x^T x_{an} - (w_y^t)^T y_{bn}$,$w_y^t$
is the value of w_y at tth iteration

Optimization VI

The corresponding gradient is given by:

$$\frac{\partial \mathcal{O}_y}{\partial w_y} = \lambda_y w_y - 2\gamma \sum_{n=1}^{N} \omega_n s_n d_n y_{bn} - \gamma \sum_{n=1}^{N} \omega_n \mu_n^y - \frac{1}{J} \sum_{j=1}^{J} \pi_i^y$$

where $\mu_n^y = (1 - s_n)(\frac{\partial \tau_1}{\partial d_n} - d_n^t) y_{bn}$.

$$\frac{\partial \tau_1}{\partial d_n} = \begin{cases} 0, & |d_n| \leq \lambda \\ \frac{a d_n - 2a\lambda sgn(d_n)}{a-1}, & \lambda < |d_n| \leq a\lambda \\ d_n, & a\lambda < |d_n| \end{cases} \quad \pi_j^y \begin{cases} 0, & |w_y^T y_j| \geq 1 \\ sgn(w_y^T y_j) y_j, & |w_y^T y_j| < 1 \end{cases}$$

Optimization VII

Relationships between different bits is important.
Using standard AdaBoost to learn multiple bits sequentially.

Algorithm 1 Co-Regularized Hashing

Input:
$\mathcal{X}, \mathcal{Y}$ – multimodal data
Θ – inter-modality point pairs
K – code length
$\lambda_x, \lambda_y, \gamma$ – regularization parameters
a, λ – parameters for SCISD function
Output:
$\mathbf{w}_x^{(k)}, k = 1, \ldots, K$ – projection vectors for $\mathcal{X}$
$\mathbf{w}_y^{(k)}, k = 1, \ldots, K$ – projection vectors for $\mathcal{Y}$

Procedure:
Initialize $\omega_n^{(1)} = 1/N, \forall n \in \{1, 2, \ldots, N\}$.
for $k = 1$ **to** K **do**
  repeat
    Optimize Equation (3) to get $\mathbf{w}_x^{(k)}$;
    Optimize Equation (5) to get $\mathbf{w}_y^{(k)}$;
  until convergence.
  Compute error of current hash functions

$$\epsilon_k = \sum_{n=1}^{N} \omega_n^{(k)} \mathbf{I}_{[s_n \neq h_n]},$$

  where $\mathbf{I}_{[a]} = 1$ if a is true and $\mathbf{I}_{[a]} = 0$ otherwise, and

$$h_n = \begin{cases} 1 & \text{if } f(\mathbf{x}_{a_n}) = g(\mathbf{y}_{b_n}) \\ 0 & \text{if } f(\mathbf{x}_{a_n}) \neq g(\mathbf{y}_{b_n}). \end{cases}$$

  Set $\beta_k = \epsilon_k/(1 - \epsilon_k)$.
  Update the weight for each point pair:

$$\omega_n^{(k+1)} = \omega_n^{(k)} \beta_k^{1 - \mathbf{I}_{[s_n \neq h_n]}}.$$

end for

Extentions

- Learning nonlinear hash functions via kernel trick: $w_x = \sum_{i=1}^{I} \alpha_i \phi_x(x_i)$ and $w_y = \sum_{j=1}^{J} \beta_j \phi_y(y_j)$
- Supporting more modalities: Incorporate loss and regularization term for new modalities and all pairwise loss terms.