Learning Multi-View Neighborhood Preserving Projections

Novi Quadrianto[1] | **Christoph H. Lampert**[2]

1: SML-NICTA & Australian National University | 2: IST Austria (Institute of Science and Technology Austria)

Abstract

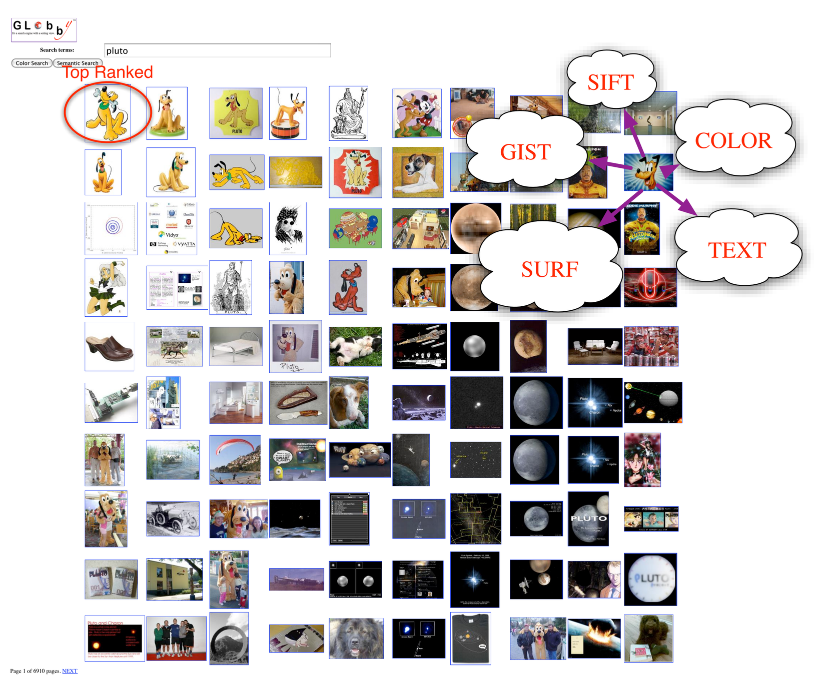
- We address the problem of projecting data in different representations into a shared space, such that the Euclidean distance in this space provides a meaningful within-view as well as between-view similarity;
- We formulate an objective function that expresses the intuitive concept that matching samples are mapped closely together in the output space, whereas non-matching samples are pushed apart;
- We show that the resulting objective function can be efficiently optimized using the convex-concave procedure (CCCP);
- Our proposed approach has a direct application for cross-media and content-based retrieval tasks.

Motivating Example

A Cross-Media and Content-Based Retrieval

Goal:

- Building an object cross-retrieval system that allows query objects and objects in the database to have different representations;
- Building a content-based object retrieval system where several representations can be used to describe a content, such as, for image objects: SIFT, Color, GIST, SURF, HOG, pHOG, Text, …

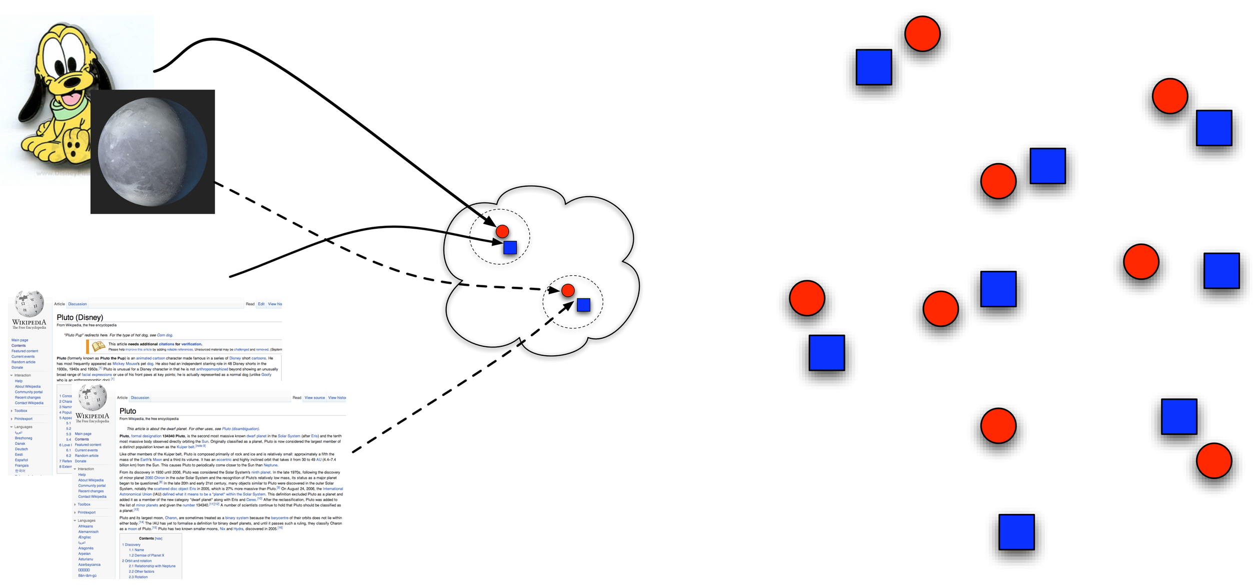


(Potential) Problems:

- During retrieval time, some of the representations might be missing. This renders approaches such as simple feature concatenation and Multiple Kernel Learning (MKL) in-applicable.

Our Solution:

- Learning a multi-view neighborhood preserving projection matrices to a common space.



A Multi-View Neighborhood Preserving Projection

Problem Setting

What we have:

- Two sets of m observed data points, $\{x_1, \ldots, x_m\} \subset \mathcal{X}$ and $\{y_1, \ldots, y_m\} \subset \mathcal{Y}$ describing the same objects;
- A cross-neighborhood set $\mathcal{S}_{x_i}$ for each $x_i \in \mathcal{X}$ that corresponds to a set of data points from $\mathcal{Y}$ that are deemed similar to x_i.

What we want:

- Projection functions, $g_1 : \mathcal{X} \to \mathbb{R}^D$ and $g_2 : \mathcal{Y} \to \mathbb{R}^D$, that respect the neighborhood relationship $\{\mathcal{S}_{x_i}\}_{i=1}^m$.

Assumption:

- A linear parameterization of the functions $g_1^w(x_i) := \langle w_1, \phi(x_i)\rangle$ for H_1 basis functions $\{\phi_h(x_i)\}_{h=1}^{H_1}$ and $w_1 \in \mathbb{R}^{D\times H_1}$ and likewise for g_2 with the weight parameter $w_2 \in \mathbb{R}^{D\times H_2}$.

Regularized Risk Functionals

- Folk Wisdom:
  Keep your friends (read: matching samples) close and your enemies (read: non-matching samples) ~~closer~~ far far away;
- Turning Wisdom into a Regularized Risk Functional:

$$\underbrace{\sum_{i,j=1}^{m} L^{i,j}(w_1, w_2, x_i, y_j, \mathcal{S}_{x_i})}_{\text{The Wisdom Loss}} + \underbrace{\eta\Omega(w_1) + \gamma\Omega(w_2)}_{\text{The Regularizer}}$$

- The wisdom loss function $L^{i,j}(\cdot)$ consists of the friends term $L_1^{i,j}$ and the enemies term $L_2^{i,j}$.

$$L^{i,j}(w_1, w_2, x_i, y_j, \mathcal{S}_{x_i}) = \underbrace{\frac{\mathbf{I}_{[\![y_j \in \mathcal{S}_{x_i}]\!]}}{2} \times L_1^{i,j}}_{\text{The Friends Loss}} + \underbrace{\frac{\left(1 - \mathbf{I}_{[\![y_j \in \mathcal{S}_{x_i}]\!]}\right)}{2} \times L_2^{i,j}}_{\text{The Enemies Loss}}$$

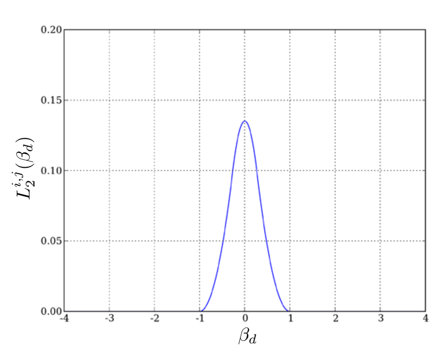
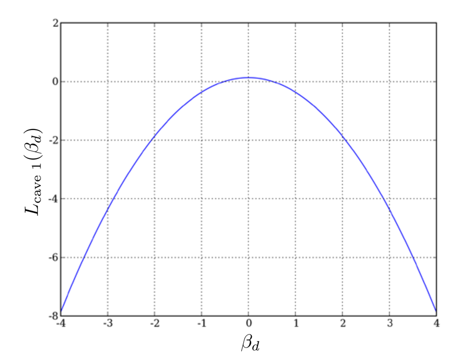
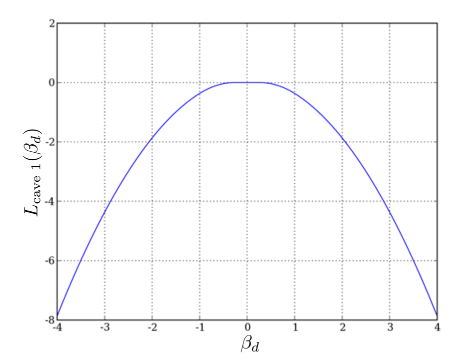
with

$$L_1^{i,j} = \left\|g_1^{w_1}(x_i) - g_2^{w_2}(y_j)\right\|_{\text{Fro}}^2 \qquad L_2^{i,j}\left(\beta_d := \left\|g_1^{w_1}(x_i) - g_2^{w_2}(y_j)\right\|_{\text{Fro}}\right) = \begin{cases} -\frac{1}{2}\beta_d^2 + \frac{a\lambda^2}{2}, & \text{if } 0 \le |\beta_d| < \lambda \\ \frac{|\beta_d|^2 - 2a\lambda|\beta_d| + a^2\lambda^2}{2(a-1)}, & \text{if } \lambda \le |\beta_d| \le a\lambda \\ 0, & \text{if } |\beta_d| \ge a\lambda, \end{cases}$$

Optimization

What so special about the wisdom loss:

- The wisdom loss function is non-convex; the friends term is convex, however the enemies term is non-convex;
- Though the enemies term is non-convex, it has a decomposition form as a difference of two convex functions: $L_2^{i,j}(\beta_d) = L_{\text{cv}}^1(\beta_d) - L_{\text{cv}}^2(\beta_d)$.

CCCP Procedure

The concave convex procedure (CCCP) finds the successive linear lower bounds on $L_2^{i,j}(.)$ and solves the resulting convex problems in w_1 and w_2 separately.

Algorithm A–Multi-View Neighborhood Preserving Projection

Input: Data sources $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_m\}$, an inter-view neighborhood relationship $\{\mathcal{S}_{x_i}\}_{i=1}^m$, number of alternations N
Output: w_1^* and w_2^*
Initialize w_1 and w_2
for $t = 1$ to N **do**
  Solve the convex optimization problem w.r.t. w_1 and obtain w_1^t
  Solve the convex optimization problem w.r.t. w_2 and obtain w_2^t
end for

Algorithm B–Hybrid-{PCA and Multi-NPP}

Input: Data sources $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_m\}$ and an inter-view neighborhood relationship $\{\mathcal{S}_{x_i}\}_{i=1}^m$
Output: w_1^{PCA} and w_2^*
Initialize w_2
Solve the optimization problem w.r.t. w_2 while fixing $w_1 = w_1^{\text{PCA}}$ and obtain w_2^*

Experiments

Dataset Statistics:

- 1000 images with 11 categories from the Israeli-Images dataset (`http://www.cs.umass.edu/~ronb/image_clustering.html`);
- We use global color descriptors as one view and local SIFT descriptors as another;
- Performance metric: k-Nearest Neighbor classification metric.

Algorithm A v. Baselines (PCA and CCA) for a Cross-Retrieval Task (accuracy ± std):

Method	#dim	5-NN	10-NN	30-NN
PCA	10	9.3±1.66	9.3±2.03	10.0±2.31
	50	9.4±1.17	10.7±1.38	10.5±2.04
CCA	10	15.4±4.27	15.8±4.53	15.9±4.59
	50	16.2±4.83	16.8±5.27	18.2±6.30
Ours	10	**18.6±2.07**	**18.9±2.28**	**18.7±2.21**
	50	**20.4±3.43**	**20.4±2.88**	**21.8±3.21**

Color Query - SIFT Database

Method	#dim	5-NN	10-NN	30-NN
PCA	10	8.2±2.54	9.2±3.35	9.4±3.36
	50	8.6±2.65	9.8±2.47	9.8±3.33
CCA	10	12.5±2.98	13.8±2.36	13.8±2.82
	50	13.2±1.77	13.2±2.32	13.4±2.62
Ours	10	**19.0±3.63**	**20.8±3.52**	**22.0±3.98**
	50	**22.6±2.07**	**22.9±1.93**	**22.4±4.30**

SIFT Query - Color Database

Cross-Retrieval Results with Algorithm B (accuracy ± std):

Crossing Type	#dim	5-NN	10-NN	30-NN	50-NN	70-NN	100-NN
Color Query	10	24.2±2.59	24.9±2.72	26.3±2.82	26.4±2.56	25.8±1.90	25.8±1.73
- SIFT Database	50	30.0±3.20	29.2±3.12	30.2±3.42	29.6±3.74	29.6±4.04	29.0±3.51
SIFT Query	10	18.8±3.59	19.1±3.14	19.4±3.71	19.8±3.91	19.7±4.19	19.9±3.92
- Color Database	50	27.8±4.27	26.8±4.28	27.0±3.09	27.4±3.78	27.8±3.90	27.9±3.82

For more experimental results, please refer to the paper.

Extensions

Kernelization:

- By the Representer Theorem, the projection matrices admits $w_1 = \sum_{i=1}^m \alpha_i k(x_i, \cdot)$, and $w_2 = \sum_{j=1}^m \beta_j l(y_j, \cdot)$, for a positive-definite kernel k on $\mathcal{X}$ and a kernel l on $\mathcal{Y}$.

Beyond 2-View:

- For the case with more than two data sources we build an analogous objective function by summing up the terms of all pairwise objectives.