Canonical Correlation Analysis based on Hilbert-Schmidt Independence Criterion and Centered Kernel Target Alignment: Supplementary Materials

Billy Chang BILLY.CHANG@MAIL.UTORONTO.CA
Dalla Lana School of Public Health, University of Toronto, M5T 3M7, Canada

Uwe Kruger UWEKRUGER@SQU.EDU.OM
Dept. Mechanical & Industrial Engineering, Sultan Qaboos University, Al Khoud, Sultanate of Oman

Rafal Kustra R.KUSTRA@UTORONTO.CA
Dalla Lana School of Public Health, University of Toronto, M5T 3M7, Canada

Junping Zhang JPZHANG@FUDAN.EDU.CN
Shanghai Key Lab of Intelligent Information Processing & School of Computer Science, Fudan University, China

1. Gradient Derivation

This section describes the derivation of (9-12) in the main article. First note that:

$$\frac{\partial K^u_{ij}}{\partial u^T} = -2\sigma_x u^T K^u_{ij}(x_i - x_j)(x_i - x_j)^T \quad (1)$$

$$\frac{\partial K^v_{ij}}{\partial v^T} = -2\sigma_v v^T K^v_{ij}(y_i - y_j)(y_i - y_j)^T \quad (2)$$

By writing:

$$tr(K^u \tilde{K}^v) = \sum_i \sum_j K^u_{ij} \tilde{K}^v_{ij}, \quad (3)$$

Equation 9 in the main article is obtained by replacing K^u_{ij} with (1) in (3).

The cyclic property of the trace operator implies:

$$tr(K^u \tilde{K}^v) = tr(\tilde{K}^u K^v) = \sum_i \sum_j \tilde{K}^u_{ij} K^v_{ij} \quad (4)$$

Equation 10 in the main article is obtained by replacing K^v_{ij} with (2) in (4).

To derive equation 11 in the main article, note that:

$$\frac{\partial \log(\rho_a(u,v))}{\partial u^T} = \frac{\frac{\partial tr(K^u \tilde{K}^v)}{\partial u^T}}{tr(K^u \tilde{K}^v)} - \frac{1}{2}\frac{\frac{\partial tr(K^u \tilde{K}^u)}{\partial u^T}}{tr(K^u \tilde{K}^u)} \quad (5)$$

The first term on the R.H.S. above is just Equation 9 in the main article divided by $tr(K^u \tilde{K}^v)$. The second term on the R.H.S. of (5) involves the derivative of $tr(K^u \tilde{K}^u)$ w.r.t. u^T. As $tr(\tilde{K}^u \tilde{K}^u) = tr(K^u \tilde{K}^u)$, we will compute the derivative of $tr(\tilde{K}^u \tilde{K}^u) = \sum_i \sum_j \tilde{K}^{u2}_{ij}$ instead, which gives:

$$\frac{1}{2}\frac{\partial tr(\tilde{K}^u \tilde{K}^u)}{\partial u^T} = \sum_i \sum_j \tilde{K}^u_{ij} \frac{\partial \tilde{K}^u_{ij}}{\partial u^T} \quad (6)$$

By considering the matrix centering formula (the equation below equation 2 in the main article):

$$\frac{\partial \tilde{K}^u_{ij}}{\partial u^T} = \frac{\partial K^u_{ij}}{\partial u^T} - \frac{1}{N}\sum_s \frac{\partial K^u_{sj}}{\partial u^T} - \frac{1}{N}\sum_t \frac{\partial K^u_{it}}{\partial u^T} + \frac{1}{N^2}\sum_{s,t}^{N} \frac{\partial K^u_{st}}{\partial u^T} \quad (7)$$

Putting (7) into (6), (6) becomes:

$$\sum_{i,j} \tilde{K}^u_{ij} \frac{\partial K^u_{ij}}{\partial u^T} - \frac{1}{N}\sum_{i,j} \tilde{K}^u_{ij} \sum_s \frac{\partial K^u_{sj}}{\partial u^T} \quad (8)$$

$$- \frac{1}{N}\sum_{i,j} \tilde{K}^u_{ij} \sum_t \frac{\partial K^u_{it}}{\partial u^T} \quad (9)$$

$$+ \frac{1}{N^2}\sum_{i,j} \tilde{K}^u_{ij} \sum_s \sum_t \frac{\partial K^u_{st}}{\partial u^T} \quad (10)$$

For the 2nd term in (8):

$$\frac{1}{N}\sum_i \sum_j \tilde{K}^u_{ij} \sum_{s=1}^{N} \frac{\partial K^u_{sj}}{\partial u^T} = \frac{1}{N}\sum_s \sum_j \frac{\partial K^u_{sj}}{\partial u^T} \sum_i \tilde{K}^u_{ij} = 0$$

since $\sum_i \tilde{K}^u_{ij} = 0$ as $\tilde{K}^u$ is column centered. Using similar arguments, along with the fact that $\tilde{K}^u$ is row centered, (9) and (10) can also be shown to be 0. Combining (1) with the first term in (8), we arrive at an explicit expression for (6):

$$-2\sigma_x u^T \sum_i \sum_j \tilde{K}^u_{ij} K^u_{ij} (x_i - x_j)(x_i - x_j)^T \qquad (11)$$

All the partial gradients involved in (5) have now been evaluated explicitly, and equation 11 in the main article, along with its weights W^u_{ij}, can be obtain by simple factorizing and reordering.

Finally, equation 12 in the main article and its associated W^v_{ij} can be obtained by considering a derivation symmetrical to the derivation above.

2. Computational Cost

The time required to run hsicCCA and ktaCCA, besides the sample size and data dimension, will also depend on the number of iterations, the convergence threshold, the bandwidth parameter σ_x and σ_y (which affects the roughness and the amount of local minima of the cost surface), the starting-parameters, and the structure of the signals within the data. At the most basic level, the time required to evaluate the cost function and the gradient for hsicCCA and ktaCCA, as a function of sample size and data dimension (P for the x-variable set and Q for the y-variable set), are presented here. Each variable set is generated using the standard Gaussian distribution, and the computations are performed using a laptop with an Intel Core i7-3517U processor and 8GB RAM.

As seen in Figure 1, the time required for evaluating the cost function is roughly linear in sample size, and rather robust against the data dimension. However, the complexity for gradient evaluation is quadratic in both the sample size and data dimension.

Although the gradient evaluation is computationally more intensive than cost function evaluation, the computational bottleneck for hsicCCA and ktaCCA lies in the step-sizes search step, where the Nelder-Mead algorithm may requires more than 30 evaluations of the cost function. For example, to compute one pair of canonical vectors using ktaCCA on a data set with sample size 1200 and dimension $P = Q = 10$, each iteration will require one gradient computation (about 1 second for each evaluation), but may require around 30 cost function evaluations (about 0.4 second for each evaluation) for step-sizes search using Nelder-Mead. For 100 iterations, this will take approximately $100 \times 1 + 100 \times 30 \times 0.4 = 1300$ seconds.

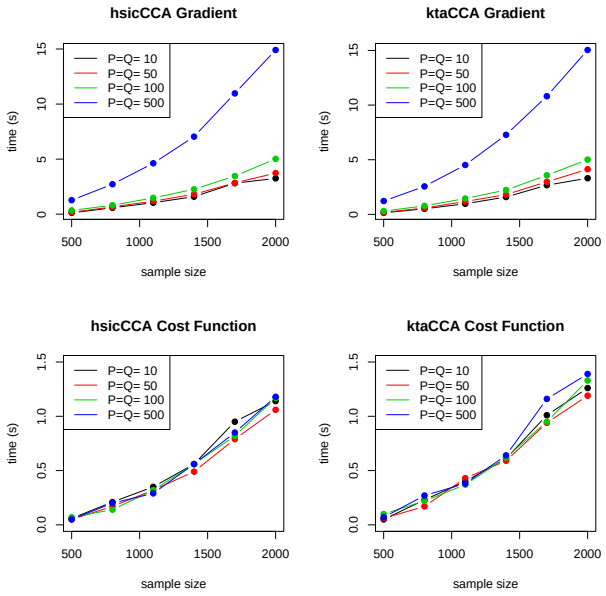

Figure 1. Time (in seconds) required to compute hsicCCA and ktaCCA's cost functions and their gradients.

3. CCA with Quadratic Features

For the Boston Housing data set, we further consider CCA and DCCA with quadratic features attached (abbreviated as qCCA and qDCCA respectively), i.e. besides the original inputs $x_i \in \mathbb{R}^P$ and $y_i \in \mathbb{R}^Q$, we attach x_i^2 and y_i^2 respectively to x_i and y_i, where x_i^2 is the vector whose elements equal the corresponding elements of x_i, but raised to the 2nd power. y_i^2 is defined similarly. 5-fold cross-validation results are presented in Figure 2.

The results suggest that quadratic relationship does exist between the two sets of variables, as qCCA has provided slightly higher Spearman correlation scores over other methods through their first canonical projections. However, strong non-linear and non-quadratic relationship exists, and this relationship is only extracted by hsicCCA and ktaCCA through their 2nd canonical projections.

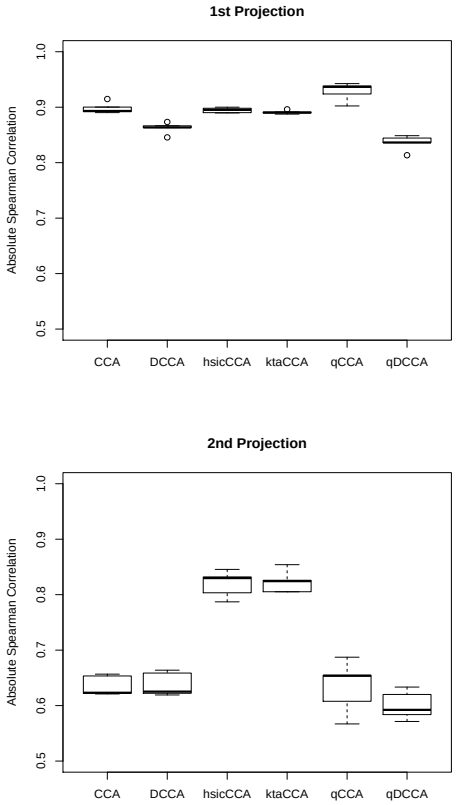

Figure 2. Boston Housing Data Results.