Nonparametric Canonical Correlation Analysis: Overview and Comparison with other CCA methods

Ilaria Bonavita

May 16, 2017

Motivation

- Reveal common variability in multiple views of the same phenomenon.

Motivation

- Reveal common variability in multiple views of the same phenomenon.
- Can provide insight into the data.

Motivation

- Reveal common variability in multiple views of the same phenomenon.
- Can provide insight into the data.
- Suppress view-specific noise factors.

Motivation

- Reveal common variability in multiple views of the same phenomenon.
- Can provide insight into the data.
- Suppress view-specific noise factors.
- Induce features that capture some of the information of the other view.
- Has been applied to problems in computer vision, natural language processing, speech recognition, genomic etc.

Motivation

- Reveal common variability in multiple views of the same phenomenon.
- Can provide insight into the data.
- Suppress view-specific noise factors.
- Induce features that capture some of the information of the other view.
- Has been applied to problems in computer vision, natural language processing, speech recognition, genomic etc.
- Several extensions (nonlinear, nonparametric, generalized...) of CCA proposed.

CCA

[Hotelling, 1936]

Problem formulation:

$X \in \mathbb{R}^{D_x}, \quad Y \in \mathbb{R}^{D_y}$ two random vectors (views)

$$\begin{aligned} \max_{\mathbf{W}_1, \mathbf{W}_2} \quad & \mathbb{E}\left[(\mathbf{W}_1^T X)^T (\mathbf{W}_2^T Y)\right] \\ \text{s.t.} \quad & \mathbb{E}\left[(\mathbf{W}_1^T X)(\mathbf{W}_1^T X)^T\right] = \mathbb{E}\left[(\mathbf{W}_2^T Y)(\mathbf{W}_2^T Y)^T\right] = \mathbf{I}. \end{aligned} \tag{1}$$

$\mathbf{W}_1 \in \mathbb{R}^{D_x \times L}, \quad \mathbf{W}_2 \in \mathbb{R}^{D_y \times L}.$

The solution can be expressed in terms of SVD of the matrix $\mathbf{T} = \mathbf{\Sigma}_{xx}^{-1/2}\mathbf{\Sigma}_{xy}\mathbf{\Sigma}_{yy}^{-1/2}$:

$$(\mathbf{W}_1, \mathbf{W}_2) = (\mathbf{\Sigma}_{xx}^{-1/2}\mathbf{U}, \mathbf{\Sigma}_{yy}^{-1/2}\mathbf{V}) \tag{2}$$

$\mathbf{\Sigma}_{xy} \approx \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T$, $\mathbf{\Sigma}_{xx}$, $\mathbf{\Sigma}_{yy}$,

$\mathbf{U} \in \mathbb{R}^{D_x \times L}$, $\mathbf{V} \in \mathbb{R}^{D_y \times L}$ top L left and singular values of $\mathbf{T}$.

Solution (**??**) can be expressed in terms of the *optimal predictor*(in *MSE* sense) of X from Y. The optimal projections are then:

$$\mathbf{W}_1^T X = \mathbf{U}^T \mathbf{\Sigma}_{xx}^{-1/2} X, \quad \mathbf{W}_2^T Y = \mathbf{D}^{-1/2} \mathbf{U}^T \mathbf{\Sigma}_{xx}^{-1/2} \hat{X} \tag{3}$$

$\hat{X} = \mathbf{\Sigma}_{xx}^{-1/2} \mathbf{\Sigma}_{\hat{x}\hat{x}} \mathbf{\Sigma}_{xx}^{-1/2}$, $\mathbf{\Sigma}_{\hat{x}\hat{x}} = \mathbf{\Sigma}_{xy} \mathbf{\Sigma}_{yy}^{-1} \mathbf{\Sigma}_{yx}$, $\mathbf{U}$ corresponds to the eigenvectors of $\mathbf{K} = \mathbf{T}\mathbf{T}^T$ and $\mathbf{D}$ diagonal matrix with the top L eigenvalues of $\mathbf{K}$ on its diagonal.

Finding maximally correlated *nonlinear* projections

Find $\mathbf{f} : \mathbb{R}^{D_x} \to \mathbb{R}^L$, $\mathbf{g} : \mathbb{R}^{D_y} \to \mathbb{R}^L$ solving

$$\begin{aligned} \max_{\mathbf{f}\in\mathcal{A},\mathbf{g}\in\mathcal{B}} \quad & \mathbb{E}\left[(\mathbf{f}(X)^T\mathbf{g}(Y))\right] \\ \text{s.t.} \quad & \mathbb{E}\left[(\mathbf{f}(X)\mathbf{f}(X)^T\right] = \mathbb{E}\left[(\mathbf{g}(Y)\mathbf{g}(Y)^T\right] = \mathbf{I}. \end{aligned} \tag{4}$$

$\mathcal{A}$ and $\mathcal{B}$ two families of measurable functions.

Kernel CCA

[Lai and Fyfe, 2000, Akaho, 2001, Melzer et al., 2001, Bach and Jordan, 2002, Hardoon et al., 2004]

$\mathcal{A}, \mathcal{B}$ two RKHSs associated with user-specified kernels $k_x(\cdot,\cdot)$ $k_y(\cdot,\cdot)$.

For the representation theorem, projections can be written as: $f_l(x) = \sum_{i=1}^{N} \alpha_{i,l} k_x(x, x_i)$ and $g_l(y) = \sum_{i=1}^{N} \beta_{i,l} k_y(y, y_i)$

$\mathbf{K}_x = [k_x(x_i, x_j)], \mathbf{K}_y = [k_y(y_i, y_j)]$ $N \times N$ kernel matrices.

Kernel CCA formulation

KCCA can be written as finding the coefficients $\{\alpha_{i,l}\}, \{\beta_{i,l}\}$ solving the optimization problem:

$$\max_{\alpha \in \mathbb{R}^N, \beta in \mathbb{R}^N} \frac{\alpha^T \mathbf{K}_x \mathbf{K}_y \beta}{\sqrt{(\alpha^T \mathbf{K}_x^2 \alpha + r_x \alpha^T \mathbf{K}_x \alpha)(\beta^T \mathbf{K}_y^2 \beta + r_y \beta^T \mathbf{K}_y \beta)}}. \quad (5)$$

The optimal coefficients are the the top L eigenvectors of $(\mathbf{K}_x + r_x \mathbf{I})^{-1} \mathbf{K}_y (\mathbf{K}_y + r_y \mathbf{I})^{-1} \mathbf{K}_\mathbf{x}$ with r_x, r_y positive regularization parameters.

Kernel CCA: keypoints

- Computing exact solution for large datasets is infeasible.
- Computational complexity of a naive implementation is $O(N^3)$.
- Several approximate techniques proposed [Bach and Jordan, 2002, Hardoon et al., 2004, Arora and Livescu, 2012, Lopez-Paz et al., 2014].
- e.g. Low-rank KCCA approximation with rank M, with complexity $O(M^3 + M^2 N)$.

Deep CCA

[Andrew et al., 2013]

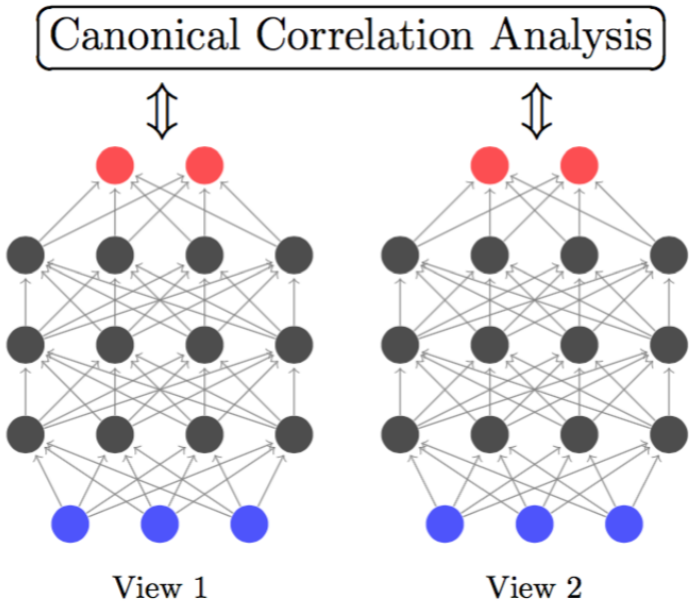

Figure: A schematic of deep CCA, consisting of two deep networks learned so that the output layers (topmost layer of each network) are maximally correlated.

Deep CCA Training

1. Pretrain the layers of each side individually (done with *denoising autoencoder* [Vincent et al., 2008]).
2. Jointly fine-tune all parameters to maximize the total correlation of the output layers H1, H2. Requires computing correlation gradient.
3. Compare the total correlation of the top k components of the two representations by performing a final (linear) CCA on the output layers of the two views on the training data.
4. The final CCA produces two projection matrices A_1, A_2, which are applied to the DCCA test output before computing test set correlation.

Deep CCA Model

DCCA computes representations of the two views by passing them through multiple stacked layers.
Let c_1 hidden units for network 1 with d layers, o units of the final layer, $x_1 \in \mathbb{R}^{n_1}$ instance of the first view, then:

$$h_1 = s(\mathbf{W}_1^1 x_1 + b_1^1) \in \mathbb{R}^{c_1}$$
$$h_2 = s(\mathbf{W}_2^1 h_1 + b_2^1) \in \mathbb{R}^{c_1}$$
$$\dots$$
$$f_1(x_1) = s(\mathbf{W}_d^1 h_{d-1} + b_d^1) \in \mathbb{R}^{o}$$

$s : \mathbb{R} \to \mathbb{R}$ non linear function, $\mathbf{W}_i^1 \in \mathbb{R}^{c_1 \times n_1}$ matrix of weights, $b_i^1 \in \mathbb{R}^{c_1}$ vector of biasis.

Deep CCA Model

Similarly, for view 2, $f_2(x_2) = s(\mathbf{W}_l^2 h_{d-1} + b_d^2) \in \mathbb{R}^o$ with $\mathbf{W}_i^2 \in \mathbb{R}^{c_2 \times n_2}$, $b_i^2 \in \mathbb{R}^{c_2}$.
The goal is to jointly learn parameters for both views that maximize the correlation between f_1 and f_2.
Given θ_v vector of all parameters $\mathbf{W}_l^v$ and b_l^v, $v = 1, 2$, the objective become:

$$\max_{\theta_1, \theta_2} \quad corr(f_1(X_1; \theta_1), f_2(X_2; \theta_2)). \tag{6}$$

Deep CCA Objective Gradient

To fine-tune all parameters via backpropagation, we follow the gradient of the correlation objective as estimated on the training data.
$\mathbf{H}_1 \in \mathbb{R}^{o \times m}$, $\mathbf{H}_2 \in \mathbb{R}^{o \times m}$ matrices of the top level representation produced by the deep models, m size of the training set.
$\bar{\mathbf{H}}_1$, $\bar{\mathbf{H}}_2$ centered data matrices.
$\hat{\mathbf{\Sigma}}_{12} = \frac{1}{m-1}\bar{\mathbf{H}}_1\bar{\mathbf{H}}_2^T$, $\hat{\mathbf{\Sigma}}_{11} = \frac{1}{m-1}\hat{\mathbf{H}}_1\hat{\mathbf{H}}_2^T + r_2\mathbf{I}$, $\mathbf{T} = \hat{\mathbf{\Sigma}}_{11}^{-1/2}\hat{\mathbf{\Sigma}}_{12}\hat{\mathbf{\Sigma}}_{22}^{-1/2}$.
Then:

$$corr(\mathbf{H}_1, \mathbf{H}_2) = \|\mathbf{T}\|_{tr} = tr(\mathbf{T}^T\mathbf{T})^{1/2} \tag{7}$$

Deep CCA Objective Gradient

Backpropragation gradient is computed on both views. Given the *SVD* decomposition of $\mathbf{T} = \mathbf{UDV}^T$:

$$\frac{\partial corr(\mathbf{H}_1, \mathbf{H}_2)}{\partial \mathbf{H}_1} = \frac{1}{m-1}(\nabla_{12}(\mathbf{H}_2 - \bar{\mathbf{H}}_2) - \nabla_{11}(\mathbf{H}_1 - \bar{\mathbf{H}}_1)). \quad (8)$$

Where $\nabla_{12} = \mathbf{\Sigma}_{11}^{-1/2}\mathbf{UV}^T\mathbf{\Sigma}_{22}^{-1/2}$ and $\nabla_{11} = \mathbf{\Sigma}_{11}^{-1/2}\mathbf{UDU}^T\mathbf{\Sigma}_{11}^{-1/2}$.
(Similarly for $\partial corr(\mathbf{H}_1, \mathbf{H}_2)/\partial \mathbf{H}_2$).

Nonparametric CCA

[Michaeli et al. 2016]

$\mathcal{A}$, $\mathcal{B}$ set of all (nonparametric) measurable functions of X and Y. We can rewrite (**??**) as an optimization problem over the Hilbert spaces:

$$\mathcal{H}_x = \{q : \mathbb{R}^{D_x} \to \mathbb{R} \mid \mathbb{E}[q^2(X)] < \infty\},$$
$$\mathcal{H}_y = \{u : \mathbb{R}^{D_y} \to \mathbb{R} \mid \mathbb{E}[u^2(Y)] < \infty\},$$

endowed with the inner products $\langle q, r\rangle_{\mathcal{H}_x} = \mathbb{E}[q(X)r(X)]$ and $\langle u, v\rangle_{\mathcal{H}_y} = \mathbb{E}[u(X)v(X)]$.

Nonparametric CCA

Then,

$$\mathbb{E}[f_i(X)g_i(Y)] = \int f_i(\mathbf{x})\Big(\int g_i(\mathbf{y})s(\mathbf{x},\mathbf{y})p(\mathbf{y})d\mathbf{y}\Big)p(\mathbf{x})d\mathbf{x} = \langle f_i, \mathcal{S}g_i\rangle_{\mathcal{H}_x}, \tag{9}$$

where

$$s(\mathbf{x},\mathbf{y}) = \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x})p(\mathbf{y})}, \tag{10}$$

and $\mathcal{S} : \mathcal{H}_y \to \mathcal{H}_x$ operator defined by $(\mathcal{S}u)(\mathbf{x}) = \int u(\mathbf{y})s(\mathbf{x},\mathbf{y})p(\mathbf{y})d\mathbf{y}$.

Nonparametric CCA

The *nonlinear* CCA problem (**??**) can be expressed as:

$$\max_{\substack{\langle f_i, f_j\rangle_{\mathcal{H}_x}=\delta_{ij},\\ \langle g_i, g_j\rangle_{\mathcal{H}_y}=\delta_{ij}}} \sum_{i=1}^{L} \langle \mathcal{S} g_i, f_i\rangle_{\mathcal{H}_x}. \tag{11}$$

When $\mathcal{S}$ is compact solution can be expressed in terms of its SVD, then the optimal projections are the left and right singular functions $\psi_i \in \mathcal{H}_x$, $\phi_i \in \mathcal{H}_y$:

$$f_i(\mathbf{x}) = \psi_i(\mathbf{x}), \quad g_i(\mathbf{y}) = \phi_i(\mathbf{y}), \tag{12}$$

$\sigma_1 + \cdots + \sigma_L$ is the maximal objective value, with $\sigma_1 \geq \sigma_2 \ldots$ singular values of $\mathcal{S}$.

Nonparametric CCA: interesting interpretations

1. $\log s(\mathbf{x}, \mathbf{y})$ is the *pointwise mutual information* PMI between X and Y.
2. $\mathcal{S}$ corresponds to the *optimal prediction* (in *MSE* sense) of one view based on the other, as $(\mathcal{S}g_i)(\mathbf{x}) = \mathbb{E}[g_i(Y)|X = \mathbf{x}]$ and $(\mathcal{S}^* f_i)(\mathbf{x} = \mathbf{y}) = \mathbb{E}[f_i(X)|Y = \mathbf{y}]$.
3. NCCA solution can be expressed in terms of the *eigen-decomposition* of a certain operator. e.g. View 1 projections are eigenfunctions of $\mathcal{K} = \mathcal{S}\mathcal{S}^*$, operator $(\mathcal{K}q)(\mathbf{x}) = \int q(\mathbf{x})k(\mathbf{x}, \mathbf{x}')p(\mathbf{x})d\mathbf{x}$ with kernel: $k(\mathbf{x}, \mathbf{x}') = \int s(\mathbf{x}, \mathbf{y})s(\mathbf{x}', \mathbf{y})p(\mathbf{y})d\mathbf{y}$.

Partially Linear CCA

$\mathcal{A}$ set of all *linear* function X, $\mathcal{B}$ set of all (nonparametric) measurable functions of Y.
$\mathbf{f}(\mathbf{x}) = \mathbf{W}^T\mathbf{x}$, $\mathbf{W} \in \mathbb{R}^{D_x \times L}$. $\tilde{\mathbf{W}} = \mathbf{\Sigma}_{xx}^{1/2}\mathbf{W}$.
$\mathcal{S}_{PL} : \mathcal{H}_y \to \mathbb{R}^{D_x}$ operator defined by
$\mathcal{S}_{PL}u = \mathbf{\Sigma}_{xx}^{-1/2} \int \mathbb{E}[X|Y = \mathbf{y}]u(\mathbf{y})p(\mathbf{y})d\mathbf{y}$.
Problem (**??**) takes the form:

$$\max_{\substack{\tilde{\mathbf{w}}_i^T\tilde{\mathbf{w}}_j=\delta_{ij}, \\ \langle g_i, g_j\rangle_{\mathcal{H}_y}=\delta_{ij}}} \sum_{i=1}^{L} \tilde{\mathbf{w}}_i^T \mathcal{S}_{PL} g_i. \tag{13}$$

Partially Linear CCA

The optimal $\tilde{\mathbf{w}}_i$'s are the top L singular vectors of $\mathcal{S}_{PL}$ and the optimal g_i's the top L right singular functions of $\mathcal{S}_{PL}$.
If we define the matrix $\mathbf{K}_{PL} = \mathcal{S}_{PL}\mathcal{S}^*_{PL} = \mathbf{\Sigma}_{xx}^{-1/2}\mathbf{\Sigma}_{\hat{x}\hat{x}}\mathbf{\Sigma}_{xx}^{-1/2}$, and we denote by $\mathbf{U}$ the top L eigenvectors of $\mathbf{K}_{PL}$, we get $\mathbf{W} = \mathbf{\Sigma}_{xx}^{-1/2}\mathbf{U}$.
Substituting $\mathbf{W}$ in the optimal $\mathbf{f}(X)$ and using *lemma 3.1* [Michaeli et. al 2016] for finding the optimal $\mathbf{g}(\mathbf{y})$, PLCCA solutions are:

$$\mathbf{f}(X) = \mathbf{W}^T\mathbf{x} = \mathbf{U}^T\mathbf{\Sigma}_{xx}^{-1/2}X, \quad \mathbf{g}(\mathbf{y}) = \mathbf{D}^{-1/2}\mathbf{U}^T\mathbf{\Sigma}_{xx}^{-1/2}\hat{X}, \quad (14)$$

$\mathbf{D}$ diagonal $L \times L$ matrix of the top L eigenvalues of $\mathbf{K}_{PL}$.

Join Probability Density Estimation

$p(\mathbf{x}, \mathbf{y})$ estimated from the set of training data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ with *kernel density estimates* (KDEs):

$$\hat{p}(\mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^{N} w(||\mathbf{x} - \mathbf{x}_i||^2/\sigma_x^2 + ||\mathbf{y} - \mathbf{y}_i||^2/\sigma_y^2),$$

$w(\cdot)$ Gaussian kernel and σ_x and σ_y kernel widths.

- KDE accuracy is affected only by *intrinsic* dimensionality but real-world data often have low-dimentional manifold structure.

NCCA implementation

$\langle \mathcal{S}g_i, f_i \rangle = \mathbb{E}[(\mathcal{S}g_i)(X)f_i(X)] \approx \frac{1}{N}\sum_{l=1}^{N}(\mathcal{S}g_i)(\mathbf{x}_l)f_i(\mathbf{x}_l)$.
$(\mathcal{S}g_i)(\mathbf{x}_l) = \mathbb{E}[s(\mathbf{x}_l, Y)g_i(Y)] \approx \frac{1}{N}\sum_{m=1}^{N} s(\mathbf{x}_l, \mathbf{y}_m)g(\mathbf{y}_m)$.
$\mathbf{S} = [s(\mathbf{x}_l, \mathbf{y}_m)]$, $\mathbf{f}_i = \frac{1}{\sqrt{N}}(f_i(\mathbf{x}_1), \ldots, f_i(\mathbf{x}_N))^T$ (simil. $\mathbf{g}_i$), then, NCCA problem (**??**) become:

$$\max_{\substack{\mathbf{f}_i^T\mathbf{f}_j=\delta_{ij},\\ \mathbf{g}_i^T\mathbf{g}_j=\delta_{ij}}} \quad \frac{1}{N}\sum_{i=1}^{L}\mathbf{f}_i^T\mathbf{S}\mathbf{g}_i. \tag{15}$$

NCCA implementation

To make the algorithm computationally efficient...

- *PCA* performed on the inputs if dimensionality is too high.
- Gaussian affinities matrices $\mathbf{W}_{ij}^{x} = 0$ if $\mathbf{x}_i$ is not within k-nearest neighbour of $\mathbf{x}_j$.
- $\mathbf{S}$ is sparse, SVD can be computed efficiently.
- The optimal projections of test sample $\mathbf{x}$ are approximated through the Nyström method [Williams and Seeger, 2001], which avoid recomputing SVD:
  $f_i(\mathbf{x}) = \frac{1}{\sigma_i^2} \sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n) f_i(\mathbf{x}_n) = \frac{1}{\sigma_i} \sum_{n=1}^{N} s(\mathbf{x}, \mathbf{y}_n) g_i(\mathbf{y}_n)$.

Relationship with KCCA

KCCA

- two kernels, one for each view.
- Factorization gives the coefficients in RKHS.
- Requires regularization.
- Need to compute inverse of kernel matrices.
- $O(M^3 + M^2N)$ time complexity (low-rank M approx.).

NCCA

- one kernel, depending on both views.
- Factorization gives the projections.
- No regularization.
- Truncated kernel affinities "trick".
- $O(kN^2)$ operations.

Future works

- Replace exact *kNN* serach with approximate search to reduce computation time.
- Explore other density estimates option different from *KDE*.