ECE598: Representation Learning	Fall 2017
Lecture 3: Nonlinear Canonical Component Analysis	
Lecturer: Pramod Viswanath	*Scribe: Bryan Clifford, Sept. 19, 2017*

3.1 Problem Setup

The setup is the same as that of CCA.

- We're given 2 data sets each consisting of N samples

$$\{\boldsymbol{x}_n\}_{n=1}^N \text{ and } \{\boldsymbol{y}_n\}_{n=1}^N.$$

- The data are two totally different measurement modalities of the same hidden (latent) variable. For example, $\boldsymbol{x}_n$ could be a video (vectorized) of a person saying a word and $\boldsymbol{y}_n$ could be the audio from the video, and the latent variable is the word the person is saying. Hence $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ live in different dimensional spaces:

$$\boldsymbol{x}_n \in \mathbb{R}^{d_x}, \qquad \boldsymbol{y}_n \in \mathbb{R}^{d_y} \qquad \forall n.$$

- We will organize the data into row matrices

$$\begin{aligned} \boldsymbol{X} &= (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)^T \in \mathbb{R}^{N \times d_x} \\ \boldsymbol{Y} &= (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)^T \in \mathbb{R}^{N \times d_y} \end{aligned} \tag{3.1}$$

3.1.1 Goal:

Our goal is to find two transforms for $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ that we can use to reduce the dimensionality of each data set to the dimension that we think the latent variable is. However, because both data sets are generated from the same latent variable, we want the transforms to be such that the transformed $\boldsymbol{x}$s and $\boldsymbol{y}$s are maximally covariant. Furthermore we want the transformed $\boldsymbol{x}$s to be uncorrelated from themselves (and the same for the transformed $\boldsymbol{y}$s).

This is almost the same goal as for CCA, however in CCA we restricted ourselves to only linear transforms of our variables. In this case we want to remove this restriction. This approach is called **Nonlinear Canonical Component/Correlation Analysis (CCA)**.

More formally we want to find two sets of functions $\{f_m\}_{m=1}^{d_z}$ such that $f_m : \mathbb{R}^{d_x} \to \mathbb{R}$ and $\{g_m\}_{m=1}^{d_z}$ such that $g : \mathbb{R}^{d_y} \to \mathbb{R}$ which solve the problem

$$\begin{aligned} \max_{\substack{\{f_m\}_{m=1}^{d_z} \\ \{g_m\}_{m=1}^{d_z}}} & \frac{\mathrm{E}\left[f_m(\boldsymbol{x}) g_m(\boldsymbol{y})\right]}{\sqrt{\mathrm{Var}\left[f_m(\boldsymbol{x})\right] \mathrm{Var}\left[g_m(\boldsymbol{y})\right]}} \\ \text{s.t.} \quad & \mathrm{E}\left[f_m(\boldsymbol{x}) f_k(\boldsymbol{x})\right] = \mathrm{E}\left[g_m(\boldsymbol{y}) g_k(\boldsymbol{y})\right] = 0 \quad \text{if } m \neq k. \end{aligned} \tag{3.2}$$

Note, this is no longer the Pearson correlation. Instead this is called the **maximum correlation** or also the **Hirschfeld-Gebelein-Renyi maximal correlation** and it is always greater than or equal to the Pearson-correlation since

$$\max_{f,g} \frac{\mathrm{E}\left[f(\boldsymbol{x})g(\boldsymbol{y})\right]}{\sqrt{\mathrm{Var}\left[f(\boldsymbol{x})\right]\mathrm{Var}\left[g(\boldsymbol{y})\right]}} \geq \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \frac{\mathrm{E}\left[(\boldsymbol{x}^T\boldsymbol{\alpha})(\boldsymbol{y}^T\boldsymbol{\beta})\right]}{\sqrt{\mathrm{Var}\left[\boldsymbol{x}^T\boldsymbol{\alpha}\right]\mathrm{Var}\left[\boldsymbol{x}^T\boldsymbol{\alpha}\right]}} \tag{3.3}$$

owing to the fact that $(\boldsymbol{x}^T\boldsymbol{\alpha}, \boldsymbol{y}^T\boldsymbol{\beta})$ is a special case of $(f(\boldsymbol{x}), g(\boldsymbol{y}))$.

The maximum correlation ρ has many important properties. For scalars (i.e. $d_x = d_y = 1$) these properties are

1. $\rho : \mathrm{P}_{XY} \to [0, 1]$, where P_{XY} is the joint probability distribution of the random variables x and y.
2. $\rho = 0$ if and only if x and y are independent.
3. $\rho = 1$ if and only if there exist either or both smooth functions h_1 and h_2 such that $x = h_1(y)$ and $y = h_2(x)$.
4. $\rho(x, y) = \rho(y, x)$
5. If x and y are jointly Gaussian then $\rho(x, y) = \rho_{\mathrm{Pearson}}(x, y)$.

It is interesting to note that these properties are also used to define the maximum correlation. Most of the work goes in to showing the "only if"s and property 5.

3.2 Solution

This section shows a method for solving (3.2). It is very similar to CCA in that it involves a change of variables and the SVD; however as will be seen the solution in this section is not very practical for high-dimensional data (and hence in general, since part of the goal is to be able to reduce dimensionality of high-dimensional data).

We begin by further restricting ourselves to only centered functions with unit variance. This can be done without loss of generality since any uncentered function $f_{\mathrm{uncentered}}$ can be centered by defining a new function $f_{\mathrm{centered}} = f_{\mathrm{uncentered}} - \mathrm{E}\left[f_{\mathrm{uncentered}}\right]$. Likewise, the variance can be set to one by dividing by the standard deviation. Our new optimization problem is thus

$$\begin{aligned} \max_{\substack{\{f_m\}_{m=1}^{d_z} \\ \{g_m\}_{m=1}^{d_z}}} \quad & \mathrm{E}\left[f_m(\boldsymbol{x})g_m(\boldsymbol{y})\right] \\ \text{s.t.} \quad & \mathrm{E}\left[f_m(\boldsymbol{x})f_k(\boldsymbol{x})\right] = \mathrm{E}\left[g_m(\boldsymbol{y})g_k(\boldsymbol{y})\right] = \delta_{m,k} \\ & \mathrm{E}\left[f_m(\boldsymbol{x})\right] = \mathrm{E}\left[g_m(\boldsymbol{y})\right] = 0. \end{aligned} \tag{3.4}$$

It is instructive to consider the case when $\boldsymbol{x}$ and $\boldsymbol{y}$ can only take on a finite number of values. This case was first proven in using a "geometric and operator theoretic interpretation" in [Wit75]. In

this case, denoting the probability mass function of the joint distribution as $p_{\boldsymbol{x},\boldsymbol{y}}$, we have

$$\begin{aligned}\mathrm{E}\left[f_m(\boldsymbol{x})g_m(\boldsymbol{y})\right] &= \sum_{\boldsymbol{x},\boldsymbol{y}} p_{\boldsymbol{x},\boldsymbol{y}} f_m(\boldsymbol{x}) g_m(\boldsymbol{y}) \\ &= \sum_{\boldsymbol{x}} f_m(\boldsymbol{x}) \sum_{\boldsymbol{y}} p_{\boldsymbol{x},\boldsymbol{y}} g_m(\boldsymbol{y}) \\ &= \boldsymbol{f}_m^T \mathbf{P} \boldsymbol{g}_m,\end{aligned} \tag{3.5}$$

where $\boldsymbol{f}_m^T$, $\boldsymbol{g}_m$ are the vectors of values of $f_m(\boldsymbol{x})$ and $g_m(\boldsymbol{y})$, respectively and $\mathbf{P}$ is the matrix of probability mass values. We also have

$$\begin{aligned}\mathrm{E}\left[f_m(\boldsymbol{x})\right] &= \mathrm{E}\left[f_m(\boldsymbol{x})1\right] \\ &= \boldsymbol{f}_m^T \mathbf{P} \mathbf{1},\end{aligned} \tag{3.6}$$

and

$$\begin{aligned}\mathrm{E}\left[g_m(\boldsymbol{y})\right] &= \mathrm{E}\left[1 g_m(\boldsymbol{x})\right] \\ &= \mathbf{1}^T \mathbf{P} \boldsymbol{g}_m,\end{aligned} \tag{3.7}$$

where $\mathbf{1}$ is a vector with 1 for every element, so that Eq. (3.4) becomes

$$\begin{aligned}&\max_{\substack{\{\boldsymbol{f}_m\}_{m=1}^{d_z} \\ \{\boldsymbol{g}_m\}_{m=1}^{d_z}}} \boldsymbol{f}_m^T \mathbf{P} \boldsymbol{g}_m \\ \text{s.t.} \quad &\boldsymbol{f}_m^T \mathbf{P} \mathbf{1} = \mathbf{1}^T \mathbf{P} \boldsymbol{g}_m = 0 \\ &\boldsymbol{f}_m^T \mathbf{P}_x \boldsymbol{f}_k = \boldsymbol{g}_m^T \mathbf{P}_y \boldsymbol{g}_k = \delta_{m,k}.\end{aligned} \tag{3.8}$$

where $\mathbf{P}_x = \mathrm{diag}\left\{\boldsymbol{p}_x\right\}$, $\mathbf{P}_y = \mathrm{diag}\left\{\boldsymbol{p}_y\right\}$, and $\boldsymbol{p}_x = \mathbf{P}\mathbf{1}$ and $\boldsymbol{p}_y = \mathbf{P}^T\mathbf{1}$ are the marginal mass function vectors.

To simplify the constrains a bit we define

$$\begin{aligned}\tilde{\boldsymbol{f}} &= \mathbf{P}_x^{1/2} \boldsymbol{f} \\ \tilde{\boldsymbol{g}} &= \mathbf{P}_y^{1/2} \boldsymbol{g}.\end{aligned} \tag{3.9}$$

which gives us

$$\begin{aligned}&\max_{\substack{\{\tilde{\boldsymbol{f}}_m\}_{m=1}^{d_z} \\ \{\tilde{\boldsymbol{g}}_m\}_{m=1}^{d_z}}} \tilde{\boldsymbol{f}}_m^T \mathbf{P}_x^{-1/2} \mathbf{P} \mathbf{P}_y^{-1/2} \tilde{\boldsymbol{g}}_m \\ \text{s.t.} \quad &\tilde{\boldsymbol{f}}_m^T \boldsymbol{p}_x^{1/2} = \tilde{\boldsymbol{g}}_m^T \boldsymbol{p}_y^{1/2} = 0 \\ &\tilde{\boldsymbol{f}}_m^T \tilde{\boldsymbol{f}}_k = \tilde{\boldsymbol{g}}_m^T \tilde{\boldsymbol{g}}_m = \delta_{m,k}.\end{aligned} \tag{3.10}$$

where we have made use of the fact that $\mathbf{P}_x^{-1/2}\mathbf{P}\mathbf{1} = \mathbf{P}_x^{-1/2}\boldsymbol{p}_x = \boldsymbol{p}_x^{1/2}$ (the vector of square-roots of marginal mass function values). Similarly, $\mathbf{1}^T\mathbf{P}\mathbf{P}_y^{-1/2} = (\mathbf{P}_y^{-1/2}\boldsymbol{p}_y)^T = (\boldsymbol{p}_y^{1/2})^T$.

Like with CCA we are very close to the SVD problem. The only difference seems to be the extra constraint

$$\tilde{\boldsymbol{f}}_m^T \boldsymbol{p}_x^{1/2} = \tilde{\boldsymbol{g}}_m^T \boldsymbol{p}_y^{1/2} = 0 \tag{3.11}$$

which says that $\boldsymbol{f}_m$ and $\boldsymbol{g}_m$ must be orthogonal to the vectors $\boldsymbol{p}_x^{1/2}$ and $\boldsymbol{p}_y^{1/2}$ respectively. However, this is actually not a problem.

Note that if we let $\tilde{\boldsymbol{f}}_m = \boldsymbol{p}_x^{1/2}$ and $\tilde{\boldsymbol{g}}_m = \boldsymbol{p}_y^{1/2}$ then the objective function equals 1 which is the **maximum** it can be! Hence from this observation we see that the first left and right singular vectors of the matrix

$$\mathbf{Q} := \mathbf{P}_x^{-1/2}\mathbf{P}\mathbf{P}_y^{-1/2} \tag{3.12}$$

are in fact $\tilde{\boldsymbol{f}}_1 = \boldsymbol{p}_x^{1/2}$ and $\tilde{\boldsymbol{g}}_1 = \boldsymbol{p}_y^{1/2}$ respectively and the corresponding singular value is $\sigma_1 = 1$.

Thus we see that we are really trying to compute the SVD of $\mathbf{Q}$, but we want to ignore the first singular vectors and focus on the remaining set. Thus **the maximum correlation is given by the second singular value of $\mathbf{Q}$ and the corresponding left and right singular vectors are the optimal values of $\boldsymbol{f}$ and $\boldsymbol{g}$.**

It is worth noting that this can be generalized using tensor products to independent pairs of random variables $((\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2))$ coming from independent distributions $p_{\boldsymbol{x}_1,\boldsymbol{y}_1}$ and $p_{\boldsymbol{x}_2,\boldsymbol{y}_2}$. Again, this was first shown in [Wit75].

3.2.1 Generalization to continuous variables

The above solution easily generalizes to the case of continuous variables, albeit with a slight caveat. Following the same argument, we see that with continuous variables Eq. (3.5) becomes

$$\mathrm{E}\left[f_m(\boldsymbol{x})g_m(\boldsymbol{y})\right] = \int\int p_{xy}(\boldsymbol{x},\boldsymbol{y})f_m(\boldsymbol{x})g_m(\boldsymbol{y})d\boldsymbol{x}d\boldsymbol{y}. \tag{3.13}$$

Now, if we restrict f_m and g_m to be members of a finite dimensional Hilbert space (see [NS82] or ECE 513 notes) then we can write them as a linear combinations of a set of basis functions

$$\begin{aligned} f_m(\boldsymbol{x}) &= \sum_{i=1}^{L_x} f_{m,i}\phi_i(\boldsymbol{x}) \\ g_m(\boldsymbol{y}) &= \sum_{i=1}^{L_y} g_{m,i}\psi_i(\boldsymbol{x}) \end{aligned} \tag{3.14}$$

An example of this could be the polynomial basis but with a domain restriction on $\boldsymbol{x}$ and $\boldsymbol{y}$ (i.e. they both have finite support) in which case the basis functions could be given by (for $d_x = d_y = 1$)

$$\phi_i(x) = x^{i-1}, \quad \psi_i(y) = y^{i-1}. \tag{3.15}$$

With the Hilbert space restriction Eq. (3.13) becomes

$$\begin{aligned} \mathrm{E}\left[f_m(\boldsymbol{x})g_m(\boldsymbol{y})\right] &= \sum_{i=1}^{L_x}\sum_{j=1}^{L_y} f_{m,i}g_{m,j}\int\int p_{xy}(\boldsymbol{x},\boldsymbol{y})\phi_i(\boldsymbol{x})\psi_j(\boldsymbol{y})d\boldsymbol{x}d\boldsymbol{y} \\ &= \boldsymbol{f}_m^T\mathbf{P}\boldsymbol{g}_m, \end{aligned} \tag{3.16}$$

where now we have

$$\mathrm{P}_{i,j} = \int\int p_{xy}(\boldsymbol{x},\boldsymbol{y})\phi_i(\boldsymbol{x})\psi_j(\boldsymbol{y})d\boldsymbol{x}d\boldsymbol{y}. \tag{3.17}$$

Note that the only difference is in the meaning our matrix and vector elements which are now the basis coefficients and basis cross-correlation values.

Continuing along the same steps and without much loss of generality (just restricting the set of Hilbert spaces we can have) assuming that $\phi_1(\boldsymbol{x}) = \psi_1(\boldsymbol{y}) = 1$ we have

$$\begin{aligned} \mathrm{E}\left[f_m(\boldsymbol{x})\right] &= \mathrm{E}\left[f_m(\boldsymbol{x})1\right] \\ &= \boldsymbol{f}_m^T \mathbf{P} \boldsymbol{e}_1 \\ &= \boldsymbol{f}_m^T \bar{\boldsymbol{\phi}}, \end{aligned} \tag{3.18}$$

and

$$\begin{aligned} \mathrm{E}\left[g_m(\boldsymbol{y})\right] &= \mathrm{E}\left[1 g_m(\boldsymbol{x})\right] \\ &= \boldsymbol{e}_1^T \mathbf{P} \boldsymbol{g}_m \\ &= \bar{\boldsymbol{\psi}} \boldsymbol{g}_m \end{aligned} \tag{3.19}$$

where $\boldsymbol{e}_1$ is the standard basis vector with all zero elements except for the first one which is 1, $\bar{\boldsymbol{\phi}} = \mathbf{P}\boldsymbol{e}_1$, $\bar{\boldsymbol{\psi}} = \mathbf{P}^T\boldsymbol{e}_1$ are the vectors of expected values of the basis functions. Now replacing $\mathbf{P}$ with $\Sigma_{\phi,\psi}$ $\mathbf{1}$ with $\boldsymbol{e}_1$ and $\boldsymbol{p}_x$ and $\boldsymbol{p}_y$ with $\bar{\boldsymbol{\phi}}_x$ and $\bar{\boldsymbol{\psi}}_y$ respectively, in the equations for the discrete case, the results are the same.

Instead of requiring that $\phi_1(\boldsymbol{x}) = \psi_1(\boldsymbol{y}) = 1$ we could also require that our basis functions be defined such that $\bar{\boldsymbol{\phi}} = \mathbf{0}$ and similarly for $\bar{\boldsymbol{\psi}}$. With this setup, the zero-mean constraint vanishes, and the maximum correlation becomes the first singular value (with respective f and g given by the singular vectors). In this case the N-CCA problem is solved by replacing $\mathbf{P}_x$ and $\mathbf{P}_y$ with the cross-correlation functions of the original basis functions, i.e.

$$\{\mathrm{P}_x\}_{i,j} = \int p_x(\boldsymbol{x})\phi_i(\boldsymbol{x})\phi_j(\boldsymbol{x})d\boldsymbol{x} \tag{3.20}$$

and respectively for $\mathbf{P}_y$.

3.2.2 Computational challenges

While the above SVD solutions to non-linear CCA can be rather practical when either the number of discrete values of $\boldsymbol{x}$ and $\boldsymbol{y}$ or the dimension of the spaces of f_m and g_m (L_x and L_y respectively) are small. In these cases the SVD can be computed efficiently and, more importantly, it is feasible to estimate the probability mass function or basis cross-correlation values ($\mathbf{P}$) from the data (via the sample estimates). Unfortunately these method's don't scale well with the size of the problem. The number of terms in $\mathbf{P}$ scales quadratically with the size of the the spaces. Hence when $\boldsymbol{x}$ and $\boldsymbol{y}$ take on many values or L_x and L_y are large, vast amounts of training data would be needed to obtain reasonable estimates of $\mathbf{P}$. There are two methods to do this in practice Kernel-CCA (K-CCA) [BJ02] and the more prominent method, the Alternating Conditional Expectation (ACE) algorithm [BF85] which are detailed in the following sections.

3.3 K-CCA

K-CCA was proposed in [BJ02] as a biproduct of the development of a form of independent component analysis that used the maximum correlation as its contrast function. This method is very similar to the continuous case discussed above. The difference is the type of Hilbert space used. In K-CCA the Hilbert space is required to have a very special structure (called a **Reproducing**

Kernel Hilbert Space) (RKHS). Specifically a RKHS is a Hilbert space of functions defined by a Mercer kernel function $K(\boldsymbol{x}_1, \boldsymbol{x}_2)$ which is defined such that the "Gram matrix" $\mathbf{K}$ with elements

$$\mathrm{K}_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{3.21}$$

is semi-positive definite for any set of $\boldsymbol{x}$s, i.e. for $\mathbf{K}$ defined from $K(\boldsymbol{x}_1, \boldsymbol{x}_2)$ and $\{\boldsymbol{x}_n\}_{n=1}^N$

$$\boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} \geq 0 \quad \forall \boldsymbol{\alpha} \neq \mathbf{0} \in \mathbb{R}^N \tag{3.22}$$

and such that any function $f(\boldsymbol{x})$ in the RKHS can be written as

$$f(\boldsymbol{x}) = \sum_{n=1}^{N} f_n K(\boldsymbol{x}, \boldsymbol{x}_n). \tag{3.23}$$

and with an inner product between elements (functions) in the space defined as

$$\langle f_1(\boldsymbol{x}), f_2(\boldsymbol{x}) \rangle := \sum_{i,j} f_{1,i} f_{2,j} K(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{3.24}$$

It should be noted that this definition of the RKHS is based on the Moor-Aronszajn theorem. In our previous notation we would say that the set of basis functions $\{\phi_l(\boldsymbol{x})\}$ is given by $\{K(\boldsymbol{x}, \boldsymbol{x}_n)\}_{n=1}^N$, and L_x is given by N.

Given the above definition every function f in an RKHS has the **reproducing property** which is

$$f(\boldsymbol{x}') = \langle K(\boldsymbol{x}, \boldsymbol{x}'), f(\boldsymbol{x}) \rangle. \tag{3.25}$$

This property leads to an important interpretation of K which is as a "feature map", i.e. a map from the $\boldsymbol{x}$ space to a function. Denoting the function defined by a particular $\boldsymbol{x}'$ as $\Phi_{\boldsymbol{x}'}(\boldsymbol{x}) = K(\boldsymbol{x}, \boldsymbol{x}')$ we have

$$\langle \Phi_{\boldsymbol{x}'}(\boldsymbol{x}), \Phi_{\boldsymbol{x}''}(\boldsymbol{x}) \rangle = K(\boldsymbol{x}', \boldsymbol{x}'') \tag{3.26}$$

In K-CCA the RKHS for f and g are the ones defined by the sets of samples $\{\boldsymbol{x}_n\}_{n=1}^N$ and $\{\boldsymbol{y}_n\}_{n=1}^N$. Let the feature maps for the spaces be denoted as $\Phi_{\boldsymbol{x}'}(\boldsymbol{x})$ and $\Psi_{\boldsymbol{y}'}(\boldsymbol{y})$, and the corresponding kernels by K_x and K_y, respectively. Given these definitions we can see that Eq. (3.27) becomes

$$\begin{aligned} \mathrm{P}_{i,j} &= \int \int p_{xy}(\boldsymbol{x}, \boldsymbol{y}) \Phi_{\boldsymbol{x}_i}(\boldsymbol{x}) \Psi_{\boldsymbol{y}_j}(\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \\ &= \int \int p_{xy}(\boldsymbol{x}, \boldsymbol{y}) K_x(\boldsymbol{x}, \boldsymbol{x}_i) K_y(\boldsymbol{y}, \boldsymbol{y}_i) d\boldsymbol{x} d\boldsymbol{y}. \end{aligned} \tag{3.27}$$

In K-CCA we then replace the integral by the sample expectation so that

$$\mathrm{P}_{i,j} = \frac{1}{N} \sum_{n=1}^{N} K_x(\boldsymbol{x}_n, \boldsymbol{x}_i) K_y(\boldsymbol{y}_n, \boldsymbol{y}_i) \tag{3.28}$$

and hence

$$\mathbf{P} = \frac{1}{N} \mathbf{K}_x \mathbf{K}_y. \tag{3.29}$$

By further restricting the kernels such that

$$\sum_{n=1}^{N} \Phi_{\boldsymbol{x}_n}(\boldsymbol{x}) = 0 \tag{3.30}$$

and respectively for $\{\Psi_{\boldsymbol{y}_n}\}_{n=1}^{N}$ then when we replace the population expectations with sample expectations we have the same problem as discussed in the previous section.

Equation (3.29) is particularly important since it shows that $\mathbf{P}$ will be written as the product of two semi-positive definite matrices (and similarly for the $\mathbf{P}_x$ and $\mathbf{P}_y$). This fact enables the K-CCA algorithm to be computed efficiently using incomplete Cholesky decompositions, which leads to linear scaling with the number of data points [BJ02].

3.4 ACE

The ACE algorithm is a very widely used algorithm that enables the maximum correlation and respective mappings f and g to be found through a simple alternating procedure which under certain conditions is proven to converge globally [BF85].

The method can be derived as follows. First note that

$$\mathrm{E}\left[(f(\boldsymbol{x}) - g(\boldsymbol{y}))^2\right] = \mathrm{E}\left[f^2(\boldsymbol{x})\right] + \mathrm{E}\left[g^2(\boldsymbol{y})\right] - 2\,\mathrm{E}\left[f(\boldsymbol{x})g(\boldsymbol{y})\right]. \tag{3.31}$$

Thus if we include the mean-zero and unit variance constraints on f and g we have

$$\mathrm{E}\left[(f(\boldsymbol{x}) - g(\boldsymbol{y}))^2\right] = 2\left(1 - \mathrm{E}\left[f(\boldsymbol{x})g(\boldsymbol{y})\right]\right). \tag{3.32}$$

we will be able to solve the N-CCA problem (for the first pair of functions) with the solution to

$$\begin{aligned} &\min_{f,g} \mathrm{E}\left[(f(\boldsymbol{x}) - g(\boldsymbol{y}))^2\right] \\ \text{s.t.} \quad &\mathrm{E}\left[f(\boldsymbol{x})\right] = \mathrm{E}\left[g(\boldsymbol{y})\right] = 0 \\ &\mathrm{E}\left[f^2(\boldsymbol{x})\right] = \mathrm{E}\left[g^2(\boldsymbol{y})\right] = 1. \end{aligned} \tag{3.33}$$

The ACE algorithm does this alternatively by iterating between the follow problems

1.
$$\begin{aligned} f_k(\boldsymbol{x}) &= \arg\min_{f} \mathrm{E}\left[(f(\boldsymbol{x}) - g_{k-1}(\boldsymbol{y}))^2\right] \quad \text{s.t.} \quad \mathrm{E}\left[f^2(\boldsymbol{x})\right] = 1 \\ &= \frac{\mathrm{E}\left[g_{k-1}(\boldsymbol{y}) \mid \boldsymbol{x}\right]}{\sqrt{\mathrm{Var}\left[\mathrm{E}\left[g_{k-1}(\boldsymbol{y}) \mid \boldsymbol{x}\right]\right]}} \end{aligned} \tag{3.34}$$

2.
$$\begin{aligned} g_k(\boldsymbol{y}) &= \arg\min_{g} \mathrm{E}\left[(f_k(\boldsymbol{x}) - g(\boldsymbol{y}))^2\right] \quad \text{s.t.} \quad \mathrm{E}\left[g^2(\boldsymbol{y})\right] = 1 \\ &= \frac{\mathrm{E}\left[f_k(\boldsymbol{x}) \mid \boldsymbol{y}\right]}{\sqrt{\mathrm{Var}\left[\mathrm{E}\left[f_k(\boldsymbol{x}) \mid \boldsymbol{y}\right]\right]}} \end{aligned} \tag{3.35}$$

where k is the iteration number. The solutions to the above optimization problems are proven as follows. In practice, dividing by the variance in the first problem can be neglected. The algorithm can be shown to converge to an optimal solution [BF85].

Following [BF85], it can be shown that the space of functions of random variables with the inner product between $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ given by $\mathrm{E}\left[f_1(\boldsymbol{x})f_2(\boldsymbol{x})\right]$ and the induced norm is a Hilbert space and that the conditional expectation operator is the projection operator from the space of functions of random variables $\boldsymbol{x}$ to the space of functions of random variables $\boldsymbol{y}$ and hence

$$f^*(\boldsymbol{x}) = \arg\min_f \mathrm{E}\left[(f(\boldsymbol{x}) - g(\boldsymbol{y}))^2\right] = \mathrm{E}\left[g(\boldsymbol{y}) \mid \boldsymbol{x}\right]. \tag{3.36}$$

This can also be shown via the orthogonality principle.

Then since we have $\mathrm{E}\left[g(y)\right] = 0$ by definition and by the tower property $\mathrm{E}\left[\mathrm{E}\left[g(\boldsymbol{y}) \mid \boldsymbol{x}\right]\right] = \mathrm{E}\left[g(y)\right]$. Hence we automatically satisfy the mean-zero constraint on $f^*(\boldsymbol{x})$.

Now the set of functions of $\boldsymbol{y}$ with unit variance is a subset of the functions of $\boldsymbol{y}$. By the tower property if we project a projection onto a subset of the first projections space its the same as projecting onto the subset. It can be shown that the projection of a vector $\boldsymbol{f}$ in a Hilbert space onto the unit ball (subset of unit norm vectors) is given by dividing the vector by $\max\{1, \|\boldsymbol{f}\|\}$. Hence the solution to the optimization problem with the norm constraint is given by the conditional expectation divided by it's variance.

In practice, the ACE algorithm is applied to data samples and the population expectations are substituted for their sample equivalents. However, computing the conditional expectations from the data can be difficult. Consider the case of continuous variables, in which every value may be unique. Hence in the case of continuous variables the conditional expectation is often replaced with a **data smooth** which could be a regression of the data or a normalized histogram of bin frequencies [BF85]. An essential property of the smoother is its having a zero-mean (which can be done by subtracting the mean of the uncentered version).

One of the main problems with the ACE algorithm is that the performance often depends strongly on the choice of data smoother.

3.5 Example

It is instructive to consider a simple 1D example in which x and y are generated from a latent variable $\boldsymbol{z} \sim \mathcal{N}(0,1)$ with additive noise for $\boldsymbol{\epsilon}_x \sim \mathcal{N}(0,\sigma)$ and $\boldsymbol{\epsilon}_y \sim \mathcal{N}(0,\sigma)$ as

$$\begin{aligned} x &= w_x e^{-z} + \boldsymbol{\mu}_x + \boldsymbol{\epsilon}_x \\ y &= w_y e^{-z.^2} + \boldsymbol{\mu}_y + \boldsymbol{\epsilon}_y \end{aligned} \tag{3.37}$$

Let us use as a smoother polynomial regression, i.e.

$$\begin{aligned} \mathrm{E}\left[g(y) \mid x\right] &\approx \sum_{m=0}^{M} a_{f,m} x^m \\ \mathrm{E}\left[f(x) \mid y\right] &\approx \sum_{m=0}^{M} a_{g,m} y^m \end{aligned} \tag{3.38}$$

where the least-squares regression solutions are given by

$$\hat{\boldsymbol{a}}_f = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\boldsymbol{g} \tag{3.39}$$

where $\mathbf{X} = (\mathbf{1}, \boldsymbol{x}, \ldots, \boldsymbol{x}^m)$, and $\hat{\boldsymbol{a}}_f$ is the vector of estimated coefficients of $f(x)$, $\boldsymbol{x}$ is the vector of x samples, and $\boldsymbol{g}$ is the vector of $g(y)$ values. The solution is similar for $\boldsymbol{a}_g$ and b_g.

We initialize the coefficients to $f(x) = (x - \mathrm{E}\left[(]\, x\right))/\operatorname{Var}\left[(]\, x\right)$ and similarly for $g(y)$ which is the optimal linear estimator. Then we estimate the coefficients of each estimator and normalize and center them repeatedly as per the ACE algorithm. The following MATLAB code simulates this for $M = 8$ and Fig. 3.1 shows the convergence of the estimated maximum correlation as well as the transformed variables. Note that since both x and y are generated from the same latent variable and an inverse transform exists to map x to z, the maximum correlation should be 1. We won't actually see this because our choice of data smoother precludes this inverse transform from the set of transforms we are projecting onto.

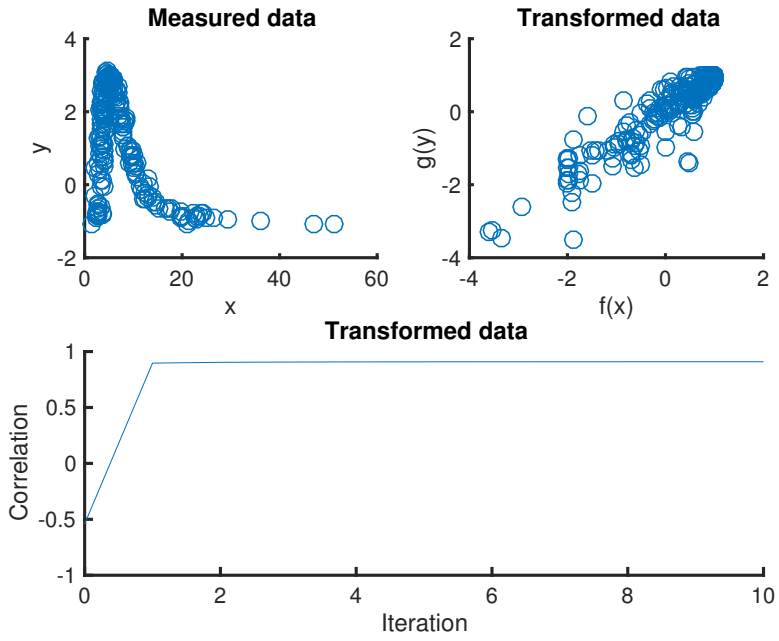

Figure 3.1: Simulation results for order 8 polynomial regression: The top left shows the original measured data and the top right shows the transformed data with the estimated optimal $f(x)$ and $g(y)$. The maximum correlation is shown on the bottom and converges very quickly.

```
%% ----- Perform ACE on random 1d example ----------------------------------------
N = 200; % number of samples
sigma_x = .5;
sigma_y = .1;

% generate random parameters and latent variable samples
z = randn(N,1);

% generate data
x = 3*exp(-z) + 2 + sigma_x*randn(N,1);
```

```
y = 4*exp(-z.^2) - 1 + sigma_y*randn(N,1);

% matrices for regression
M = 8; % polynomial order
X = ones(N,1);
Y = ones(N,1);
for m = 1:M
    X = [X,x.^m];
    Y = [Y,y.^m];
end

% perform the ACE algorithm
Nitr = 10; % number of iterations
f = (x - mean(x,1))/std(x);
g = (y - mean(y,1))/std(y);
a_f_vals = [];
b_f_vals = [];
a_g_vals = [];
b_g_vals = [];
cor_vals = mean(f.*g);
for k = 1:Nitr
    % estimate E[g(y)|x]
    coefs = (X'*X)\(X'*g);
    f = X*coefs;
    f = f - mean(f);
    f = f./std(f);

    % estimate E[f(x)|y]
    coefs = (Y'*Y)\(Y'*f);
    g = Y*coefs;
    g = g - mean(g);
    g = g./std(g);

    % compute correlation
    cor_vals(end+1) = mean(f.*g);
end

% display results
figure;
% ---- measured ----
subplot(2,2,1);
scatter(x(:,1),y(:,1),128);
title('Measured data');
xlabel('x');
ylabel('y');
% ---- transformed ----
subplot(2,2,2);
```

```
scatter(f,g,128);
title('Transformed data');
xlabel('f(x)');
ylabel('g(y)');
% ---- convergence ----
subplot(2,1,2);
hold all;
plot(0:Nitr,cor_vals);
title('Transformed data');
ylabel('Correlation');
xlabel('Iteration');
```

Bibliography

[BF85] L. Breiman and J. Friedman. Estimating optimal transformations for multiple regression and correlation. *J. Am. Stat. Assoc.*, 80:580 – 598, 1985.

[BJ02] F. Bach and M. Jordan. Kernel independent component analysis. *J. Machine Learn. Res.*, 3:1 – 48, 2002.

[NS82] A. W. Naylor and G. R. Sell. *Linear Operator Theory in Engineering and Science*. Springer-Verlag, 2 edition, 1982.

[Wit75] H. Witsenhausen. On sequences of pairs of dependent random variables. *SIAM J. Appl. Math.*, 28:100 – 113, 1975.